DESIGN AND IMPLEMENTATION OF A WEB-BASED UNIVERSITY ADMISSION AND PLACEMENT NEURAL NETWORK MODEL

BY

AIGBE, PATIENCE ERINMA

A THESIS SUBMITTED TO THE POSTGRADUATE SCHOOL IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

DEPARTMENT OF MATHEMATICS

FACULTY SCIENCES

AHMADU BELLO UNIVERSITY

ZARIA

NIGERIA.

AUGUST, 2008

DECLARATION

I declare that the work in this thesis entitled 'Design And Implementation Of A Web-Based University Admission And Placement Neural Network Model' has been carried out by me in the Department of Mathematics under the supervision of Dr. O. S. Adewale.

The information derived from the literature has been duly acknowledged in the text and a list of references provided. No part of this thesis was previously presented for another degree or diploma at any other University to the best of my knowledge.

AIGBE, Patience Erinma	_____	_____
Name of Student	Signature	Date

CERTIFICATION

This thesis entitled “DESIGN AND IMPLEMENTATION OF A WEB-BASED UNIVERSITY ADMISSION AND PLACEMENT NEURAL NETWORK MODEL” by AIGBE, Patience Erinma meets the regulations governing the award of the degree of Master of Science of Ahmadu Bello University, Zaria, and is approved for the contribution to knowledge and literacy presentation.

_____ _____

Dr. O. S. Adewale Date

Major Supervisor

_____ _____

Dr. D. N. Choji Date

Minor Supervisor

_____ _____

Prof. S. O. Falaki Date

External Examiner

_____ _____

Prof. G. U. Garba Date

Head of Department

_____ _____

Prof. S. A. Nkom Date

Dean Postgraduate School

DEDICATION

This research work is dedicated to my Lord and Saviour Jesus Christ; Who is my Life, my Way, and my End. In His mercy, He sought me, found me in my nothingness, and made somebody out of me.

ACKNOWLEDGEMENT

My heart overflows with joy and gratitude to my Father and my God who gave me back my life in the operating theatre, and strengthened me to continue and finish this research. I thank Him immensely for granting me journey mercies, provision and wisdom during this programme. From the depth of my heart, I sincerely appreciate my lecturer and supervisor, Dr. O. S. Adewale whose critical supervision and inputs made it possible for me to surmount the hurdles I encountered during the course of this research. Sir, you were patient with me even in times when I displayed gross lack of understanding and out of your wealth of knowledge, you unreservedly released 'light' to me which saw me out of the 'dark tunnel'. I lack words to express my gratitude to you. May God who alone knows how best to reward every one, bless and reward you and your family abundantly.

I very much appreciate Prof. O. C. Akinyokun, Prof. S. Junaidu, and Dr. Choji, my lecturers, from whom I learnt a great deal during the period of course work. To the entire staff of the department of Mathematics, ABU Zaria, I say 'thank you' and especially to Prof. D. Singh, Prof. G. U. Garba (HOD, Mathematics Dept.), Dr. B. K. Jha, and Mr. A.A. Obiniyi, who so much encouraged me to finish up even when the end was no where in sight. I also appreciate Mr. Ajibade and Dr. A. Abdulraheem whose sound counsel on procedures helped me a great deal. My course mates are not left out. Being in class after quite some years would have been very laborious where it not for the cordial relationship we enjoyed.

I sincerely thank the management of Ahmadu Bello University, Zaria for giving me this opportunity to further my studies, and to the management and staff of the Iya Abubakar Computer Centre who did their best to encourage me, even in the midst of so much changes and challenges.

Very importantly and with my heart in my mouth, I say a ‘Big Thank You’ to my darling husband without whose understanding and sacrificial love, I may have abandoned this programme half way through. My children, my jewels, I am very grateful to you for your understanding and prayers for mummy. Thank you for permitting me to leave you all those long hours and even days just to see to the conclusion of this programme. I love you, my family and may God’s blessings remain with you, always.

This dissertation has been completed with support from the ABU MacArthur Foundation Projects Grant. I am grateful for the support.

ABSTRACT

Every year the number of applicants seeking admission into Nigerian Universities increases by leaps and bounds although the Universities lack the commensurate facilities to meet the challenges of admitting the high number of applicants. For this reason, the admission officers have to manually evaluate every candidate's data against the set admission requirements to screen the applicants in order to select the number of candidates that their universities can accommodate. The procedures involved are very cumbersome, time consuming and prone to a lot of human errors and irregularities. Many candidates miss out on the admission every year, and the most painful aspect of this manual process is that many who are not qualified for a particular course end up being given placement into such courses while the more qualified ones are left out. Consequently, for lack of aptitude for the course, the students struggle through and many even resort to cheating their way through examinations and then graduate out of the Universities ill-equipped for the job market and the society. On the other hand, some of the less fortunate but qualified ones who are not given University admission year after year, become so frustrated over time and end up in hideous lifestyles. Whichever way, the society suffers and national growth is hindered.

In this work, a web-based model was designed to considerably take care of the above problems. The system was developed to provide a time-efficient, detailed and unbiased automated procedure for selecting the most qualified candidates for admission into universities, and ensure that qualified candidates, who fail to meet the requirements for a particular course, are automatically placed into other courses for which they meet the admission requirements and where vacancies exist, using neural network model. The model also provides an avenue for students self-screening admission system.

The system design, implementation and results are presented in chapter four. The implementation was based on AMP (Apache, MySQL, and PHP) open source solutions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE
INTRODUCTION

1.1 Background to the Study and Statement of the Problem

Higher education in Nigeria can be traced to 1932 when Yaba Higher College was established for the purpose of producing assistants who would relieve the then colonial administrators of menial tasks. Thus in 1940, the University College, Ibadan was established but the programmes offered there and then were narrow because the agenda of the colonial administration did not include the training of high-level manpower for many of the professions. The Ashby Commission in 1960, recommended the establishment of regional universities in the then three regions of Nigeria. Three universities were established: the University of Nigeria, Nsukka (1960) in the Eastern region; the University of Ife, now Obafemi Awolowo University (1961) in the Western region and Ahmadu Bello University, Zaria (1962) in the Northern region, while the existing University College, Ibadan was granted full-fledged University status in 1962. Also, the University of Lagos, Akoka came into existence in 1962 and as a city University, it provided courses in law, social sciences, medicine, humanities, engineering and part-time programmes for working students. Lastly, the University of Benin was established in 1970, making the sixth of the Universities that have come to be known as Nigeria's first generation Universities (Adesina, 1988).

Today the higher education system in Nigeria is composed of universities, polytechnics, institutions of technology, colleges of education that form part of, or are affiliated to, universities, and professional, specialized institutions. They can be further categorized as private, state or federal owned institutions. Federal

universities categorized as first, second, or third generation universities, are owned and funded by the federal government, while state universities are owned and financed by the states (there are 36 states in all), and private universities are owned and funded by individuals or religious organisations. As seen from above, the first generation universities are the six universities established in the 1960s and early 1970's; second generation universities are seven universities established in the mid 1970's; while third generation universities refer to the eleven institutions, including the universities of technology, established in the 1980's and 1990's (Hartnett, 2000).

According to the National Universities Commission (NUC)'s report on the results of the November 2005 System-Wide Accreditation Exercise, there are twenty-five (25) federal universities including three (3) universities of agriculture, twenty (20) state universities, twenty (23) private universities, five (5) degree-awarding colleges of education, sixty-nine (69) National Certificate in Education (NCE) - awarding colleges of education, one (1) military university, four (4) inter-university centres. This gives a total of one hundred and forty-two (142) higher education institutions excluding the polytechnics and the ten (10) newly approved private universities in 2006.

Higher education in Nigeria can be further divided into the public or private, and the university or non-university sectors. Public universities, owned by the federal and state governments, dominate the higher education system. The non-university sector is composed of polytechnics, institutions of technology, colleges of education, and professional institutions. There is no sharp distinction between the university and the non-university sectors; most of the institutions in the latter sector are affiliated with universities.

1.1.2 Access to Higher Education in Nigeria

There are three levels of university education in Nigeria. The university level first stage offers a Bachelor's degree after a minimum of three years and a maximum of six years study (e.g. in medicine). The university level second stage offers a Master's degree following one year of post-Bachelor's study or one of post-graduate diploma study and a year of post-Bachelor's study in the relevant discipline. The university level third stage offers doctorate degree of two to three years duration after the Master's degree. To gain admission into the first level of university education, a potential student has to pass the competitive University Matriculation Examination (UME).

In Nigeria and in fact most nations of the world, the University is the highest citadel of learning for the production of high-level human resources for the labour market. In recognition of this and the role of higher education in perpetuating national unity, the Federal Government of Nigeria took appropriate steps to ensure equity with regard to access to university education. The Joint Admissions and Matriculation Board (JAMB) was created by Act No. 2 of 1978 of the Federal Military Government (JAMB, 2004). The main aim for the establishment of the Joint Admissions and Matriculation Board (JAMB) was to provide an opportunity for eligible Nigerians to have access to university education, and to diversify the intakes, and achieve a high rate of national spread in the placement of applicants into Nigerian universities (JAMB, 2004). In addition, the JAMB was to place suitably qualified candidates into the existing tertiary institutions after taking into account the vacancies available in each tertiary institution. Placement was to be done on the basis of merit, catchment

area, and with a special focus on females and the Educationally Less Advantaged States (Omoike and Aluede, 2007).

The Federal Government controls the universities and other higher education institutions through the following organs: the Federal Ministry of Education; the National Universities Commission, which among other things allocates funds to federal universities and also prescribes the spending formula; and the Committee of Vice Chancellors of Nigerian Universities, which acts as a coordinating body and offers advice to government and universities' governing councils on matters of general and specific concern to higher education.

Individual university administration is the joint responsibility of the university's Governing Council and the Senate. Although the Governing Council remains the highest policy-making body in the university, an appointed Vice-Chancellor acts as the Chief Executive Officer, coordinating both academic and administrative functions. Within universities and colleges, the institutes and centers are more or less autonomous. The university system polity consists of three distinct categories of staff viz: administrative, academic and technical, each having a union that protects the interest of members. For instance, the Academic Staff Union of Universities (ASUU) safeguards the interests of the academicians in the Nigerian university system. Most universities operate the semester system, where a session comprises of two semesters. The university academic year begins in October and runs through to July.

The Federal Government in order to make education relevant to the needs and aspirations of the people and so bring about the desired development reviewed her educational system by introducing the 6-3-3-4 system of education

(Maduewesi and Imhanlahimi, 2006). By this system, before reaching university entrance level, students undertake 12 years of school education, the first six years of which are spent at primary school while the remaining six years are split between junior and senior secondary school education. Then the last four years of the education system are spent on an average four–year course in the university.

Generally, students are 18 years old at the start of their university education, though some students are able to gain admission at the younger age of 16. Students may be admitted into the first year of a four-year degree course based on results achieved in the Universities Matriculation Examination (UME) conducted by the Joint Admissions and Matriculation Board (JAMB). Entrance to the first year of a three-year programme is based on results obtained in the West African School Certificate (WASC) O level, in addition to either the General Certificate of Education (GCE) A level, or equivalent examinations such as the Interim Joint Matriculation Board Examination (IJMBE), and the National Diploma (ND) certificates.

Admission is through the Joint Admissions and Matriculation Board (JAMB) for both UME and direct entry applicants. For UME, interested applicants purchase the JAMB form and subsequently write the University Matriculation Examination (UME). Based on the score profile of applicants and the recommended enrolment figure for the admission year, a minimum cut-off score is usually proposed for UME admissions. For instance, for the 2005/2006 session, a total enrolment figure was proposed as follows: ARTS (2139) and

SCIENCE (3210) for Ahmadu Bello University, and the proposed merit cut-off score was 234 for that session.

1.1.3 Statement of the Problem

As stated in the preceding section, the Joint Admissions and Matriculations board has the responsibility of placing suitably qualified candidates into Nigerian tertiary institutions after taking into account the vacancies available in each institution. However, the records show that well over 500,000 candidates seek placements into universities annually in Nigeria and only about 13% (on approximation) of them secure admission, which is a far cry from the target. Admission decisions are made by educational institutions by considering a variety of factors. Some of the evaluation criteria normally used are: JAMB UME subject combination; university's admission requirements; overall scores in JAMB UME results; UME merit cut-off score; five credits obtainable in 'O' level certificate (in not more than two sittings); catchment area considerations; and educationally less developed states (elds) considerations. In 2006, the Post UME criterion was introduced whereby, a potential candidate having sat and passed the JAMB UME, is examined by his/her University of choice. The Post UME criterion further narrow down or increases the chances of the candidates being selected and placed in the course of their choice or any available course.

Faced with all the multiple criteria as stipulated above, the admission officer manually evaluates every candidate's data against the various admission

requirements before taking admission and placement decisions. This process is quite cumbersome and highly not cost-effective in terms of cost and processing time. Also, as with every manual process, it is fraught with inaccurate decisions resulting from avoidable human processing errors and at times deliberate manipulations to achieve some unwholesome personal aims like admitting unqualified candidates who has personal affiliations with the admission officer.

The inherent problems with the manual system of admission gave rise to the need for development of a computerised model that can be used to carry out the admission procedures with a view to:

(a) reducing to the barest minimum the admission processing cost and time in terms of man-hours

(b) removing all elements of human errors be they intentional or unintentional

(c) streamlining admission processing work

(d) making admission decisions more objective and impartial

(e) admitting only qualified candidates and

(f) providing a self-screening and evaluation mechanism for candidates.

1.2 Research Motivations

Every year more than 500,000 applicants seek placement to universities. Of these applicants only about 13% are selected for admission. That shows that a large number of these candidates miss out on the opportunity of being admitted into the Universities and, for some candidates, this trend continues for many years leading to frustrations.

This problem has remained difficult to surmount because, for any admission and placement process, several admission criteria are considered and the candidates' data would have to be manually evaluated against the various admission requirements by each university's admission officer before selecting the few qualified candidates. And this is not an easy task.

Every Nigerian that seeks admission into the University must have one thing or the other to do with one of the following agencies:

i. Joint Admission and Matriculation Board (JAMB);
ii. West African Examination Council (WAEC);
iii. National Examination Council (NECO);
iv. National Teacher's Institute (NTI);
v. Interim Joint Matriculation Board (IJMB).

Adebiyi (2006) developed a web-based model for JAMB candidates' admission and placement into Nigerian Universities. In addition to the general admission requirement of five 'O' level credits in not more than two sittings, Adebiyi, in his model, took into consideration other admission requirements such as UME subject combination and total score requirements, and University's course subject combination requirements as the criteria for admission and placement. His model also presented an opportunity for students self – screening admission system.

The model however, did not take into consideration candidates that get University placement based on variables such as catchment areas and those from educationally less developed states (elds). Since these variables are part of the standard criteria in the University admission and placement process in Nigeria, Adebiyi's model would deny admission and placement to many candidates who these variables are meant to favour.

This study therefore makes an improvement on the model presented by Adebiyi (2006). This is achieved by the development of a web-based model that takes into consideration admission requirements other than those considered by his model. These admission requirements include:

a. UME subject combination and total score requirements,
b. University's course subject combination requirements,
c. 'O' level subject combination and credit pass requirements,
d. Post UME requirements,
e. Candidates from the University's catchment area, and
f. Candidates from the educationally less developed states.

In this era of modern computer technology and information science, sophisticated information systems can be built to make decisions or predictions based on information contained in available past data. Such systems are called learning systems and are currently used for the purpose of classification and prediction. A student data evaluation approach based on neural networks was described in his dissertation (Sheel et. al., 2002). This was used to determine the placement of university students into basic mathematics courses. The existing

approach used the results of a mathematics placement examination given to incoming freshmen to determine course placement, but the study suggested neural networks to be suitable alternatives to such examinations.

1.3 Research Objectives

The objective of this study is to develop and implement a web-based model for the Admission and Placement of Potential Students into Nigerian Universities. The model evaluates potential students' data against University admission requirements thereby streamlining and automating the processing work involved in student admission and placement procedures. In addition, the model will provide candidates with an opportunity to perform self-screening and personally evaluate their chances of gaining admission and placement into the Universities of their choice.

1.4 Research Methodology

A review of related literature is made along with a few related works, and a web-based admission and student placement process model was developed. The technological approach to the implementation was based on open source solutions. The system requirement in terms of software and hardware includes a web server, which is apache extended with support for PHP and MySQL relational database.

In recognition of the sensitivity of the data contained in the system, communications over the public network are protected with open-ssl library for data encryption and authentication and role-based authorisation was built into

the system to specify access rights to the database system. System testing was carried out succinctly with real test data which were obtained from the faculty of science of the Ahmadu Bello University, Zaria. Also, as a follow up to the test-running and subsequent debugging of the system, the system was implemented. At the end of it all, potential students should be able to go on the relevant University's website and verify whether they qualify for admission and placement into that University. This is self-screening.

1.5 Limitations to the Study

Some limitations to this study are identified in the area of time-constraint and physical restriction to real data. Due to the volume of work involved in this study, much time is needed to design the model, develop the program code, test-run, debug and fully implement the new system, than is available for the period of this study. Other limitations include the unavailability of constant electric power supply, the erratic nature of Internet connectivity in most parts of the country, and the training and retraining of dedicated staff who will monitor the system.

1.6 Contributions to Knowledge

By the development of this Web-based Admission and Placement model, potential candidates' data are evaluated against the university admission requirements and the most qualified candidates are thereby selected. The new system has tremendous contributions to knowledge, as seen below:

(i) government's policy to promote higher education, learning and research is realised since the automated system frees a lot of man-hours to staff

involved in the cumbersome, time-consuming admission process, and, the recovered time can be gainfully employed in capacity building.

(ii) many more people now have fair opportunity to increase in knowledge because, when candidates fail to meet some of the admission requirements for their first course, the automated system automatically considers such candidates for placement into any other course for which they meet the requirements, provided there are vacant positions.

(ii) a time-efficient, detailed and unbiased automated procedure for selecting the most qualified candidates for admission into universities is put in place and most (if not all) of the bottlenecks that is rife with the current time consuming and error-prone manual admission and placement processes becomes history.

(iii) the prevalence of cheating is reduced and hard work is encouraged through the student self-screening admission system .

(iv) University admission and placement is made more reliable and this goes to reduce the tendency to corrupt practices even in the larger society.

(v) since it provides better admission opportunity for qualified candidates, better qualified graduates will now be turned out into the job market as opposed to the output that comes from persons who struggle through the universities because they were never qualified to be there in the first instance.

(vi) with qualified candidates admitted, lecturers have more time on their hands for research which ultimately increases knowledge in all ramifications, since they do not have to over labour themselves to impart

knowledge to students who do not possess the aptitude for university education.

1.7. Organization of the Thesis

This report is divided into five (5) chapters. In chapter one, a general introduction and background of the study is given, stating the problem, research motivations, objective, methodology, scope of the research and its contributions to knowledge.

Chapter two presents a review of various literature in the field of artificial neural networks and some related works that support this research work.

In chapter three, the system analysis and modelling is presented showing a detail of the research methodology.

The system design and implementation is contained in chapter four with emphasis on the software platform for the implementation, the system hardware requirements and the results of the implementation.

Finally, chapter five presents the conclusion and recommendations from the study.

CHAPTER TWO

LITERATURE REVIEW

First, a review is made of current multivariate predictive techniques such as neural networks and discriminant analysis. A description of the research that has been accomplished in the area of predicting student selection for admission or placement is also provided.

2.1 Artificial Neural Networks

Artificial Neural Networks, sometimes called the sixth generation of computing, offer a completely different approach to problem solving. They are structured to provide the capability to solve problems without the benefits of an expert and the need of programming. They can seek patterns in data that are not evident.

Artificial systems that mimic general brain processes have recently emerged as a promising new computing paradigm (Caudill and Butler, 1990; Simpson, 1990; Wasserman, 1990). These systems are proving their usefulness in novel statistical applications and in parallel with this development, the literature of neuroscience is showing a rapid increase in citations related to neural models (Barinaga, 1990). Despite the diversity of their applications, these models, generally referred to as **neural networks**, **connectionist networks**, or **artificial neural systems**, share certain attributes. The typical neural network is built around a set of adaptive elements, related through a connection matrix. The exact structure of the connection matrix, the range of its values and the details of interaction between neural elements vary from model to model. The important point is that structures built of populations of these adaptive elements are capable of complex learning behavior. This behavior is *self-organizing—*

it is not programmed or governed by an internal central processing unit (Kohonen, 1987).

As statistical computing systems, these structures can operate on sets of data with minimal supervision and can recognize patterns or find associations without being given explicit rules or descriptions. Neural computation may be applicable when assumptions required for conventional statistical processing, for example multivariate normality, or equality of group covariance matrices do not hold or cannot be tested. Consequently, neural networks may be able to solve previously intractable problems, particularly in classification, pattern recognition or analysis of noisy data.

Concisely, Neural Networks or Artificial Neural Networks, as they are often referred to in the literature, are crude electronic models based on the neural structure of the brain. Traditionally, the term neural network had been used to refer to a network or circuit of biological neurons. The modern usage of the term often refers to artificial neural networks, which are composed of artificial neurons or nodes. Thus the term 'Neural Network' has two distinct usages:

(i) Biological neural networks are made up of real biological neurons that are connected or functionally-related in the peripheral nervous system or the central nervous system. In the field of neuroscience, they are often identified as groups of neurons that perform a specific physiological function in laboratory analysis.

(ii) Artificial neural networks are made up of interconnecting artificial neurons (programming constructs that mimic the properties of biological neurons). Artificial neural networks may either be used to gain an understanding of biological neural networks, or for solving artificial

intelligence problems without necessarily creating a model of a real biological system.

In more practical terms neural networks are non-linear statistical data modelling or decision making tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. In the literature, little or no distinction is made between the terms neural networks and artificial neural networks and, both are used interchangeably in this work to avoid the confusion of taking them to be two different concepts.

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. It is composed of a large number of highly interconnected processing elements called neurones, working in unison to solve specific problems. Neural elements of a human brain have a computing speed of a few milliseconds, whereas the computing speed of electronic circuits is of the order of microseconds yet, the human brain resolves certain problems such as vision and language problems much faster than the fastest computers. Neural Networks, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Just as learning in biological systems involves adjustments to the synaptic connections that exist between the neurones, so it is with ANNs as well.

ANN models mimic the brain. They provide a computing architecture that is radically different from the computers that are widely used today; they are massively parallel systems.

Historical Background

Neural networks trace the computational side of their ancestry to the mathematical theories of Turing, the logic networks of McCulloch, Pitts, and Shannon and the theoretical and applied genius of von Neumann. Reprints of their original papers are contained in Anderson (1986).

Fundamental assumptions behind early (approximately 1945) models were:

1. the brain could be viewed as a logical computation system using binary elements; and
2. reasoning processes could be reduced to symbolic logic.

Thus, a cognitive machine might be built from a sufficient number of basic logic units. Pursuit of these ideas, and the addition of the stored program concept of von Neumann led to the electronic architecture used in almost all computers today (Brown, 1987).

The biological heritage of neural networks begins with the "neuron doctrine" of Cajal, Golgi, and Waldeyer (Shepherd, 1990). The first artificial neuron was produced in 1943 by the neurophysiologist Warren McCulloch and the logician Walter Pits. But the technology available at that time did not allow them to do much. Although the logical elements of McCulloch and Pitts were "neural" in that they either "fired or did not fire" in response to input pulses, these elements were unlike biological neurons in that they were fixed—there was no modification or learning in response to the history of inputs to the elements. However, Donald Hebb theorized that changes in synaptic strengths at the axonal-dendritic connections are the basis of learning in the central nervous system (Hebb, 1949). In his book “Organization of Behaviour”, Hebb (1949) pointed out that neural pathways are strengthened each time they are used and he further provided the learning law that became the starting point for artificial neural network algorithms.

Recent biological examples of Hebbian learning have been found (Zador, et *al*., 1990), and artificial models of Hebbian learning have been constructed (Carpenter, 1988).

The first significant artificial neural systems with modifiable elements were the perceptrons (Rosenblatt, 1958). The Perceptron was built in hardware and is the oldest neural network still in use today. The perceptron computes a weighted sum of inputs, subtracts a threshold, and passes one of two possible values out as the output. Although these systems could learn patterns without supervision, Marvin Minsky and Seymour Papert (1969) in their book “Perceptrons” showed that perceptrons could not discriminate classes of patterns known as "linearly separable sets" and were theoretically incapable of solving many simple problems. This discouraged many researchers. But, a few dedicated scientists such as Stephen Grossberg, Bernard Widrow, Teuvo Kohonen, John Hopfield, and James Anderson continued their research efforts and gradually, a new theoretical foundation emerged. The mathematics of perceptrons was further elucidated by Nilsson (1990). The linear separability problem has been overcome by using layers of elements with appropriate rules for feeding information through the layers (Barinaga, 1990).

Bernard Widrow and Marcian Hoff (1960) developed the models ADALINE and MADALINE. These models were named for their use of Multiple ADAptive LINear Elements. MADALINE was the first neural network to be applied to a real world problem. It is an adaptive filter which eliminates echoes on phone lines and it is still in commercial use.

Artificial Neural Network models are known as connectionist models of parallel distributed processing (pdp) models. PDP models are well described in the definitive book by Rumelhart et al. (1985). Many other books describing neural network models

are now available (Grossberg, 1989; Pao, 1989; Wasserman, 1989; Hecht-Nielsen, 1990; Khanna, 1990). The human nervous system consists of cells called neurons. There are hundreds of billions of neurons, each connected to hundreds or thousands of other neurons. Each neuron is capable of receiving, processing, and transmitting electrochemical signals over the neural pathways that comprise the brain's communication system. The neurons consist of four basic parts: the cell body or soma, synapse, axon, and dendrite. Dendrites are the branch-like structures that provide the input to a cell body. They receive signals from other neurons at connection points called synapses. On the receiving side of the synapse these inputs are connected to the soma (cell body). The cell body essentially sums the membrane potentials provided by the dendrites. When the cumulative excitation in the cell body exceeds the threshold, the cell fires, sending a signal down the axon to another neuron. (Adewale, 2006).

Figure 2.1 shows the relationship of these four parts.

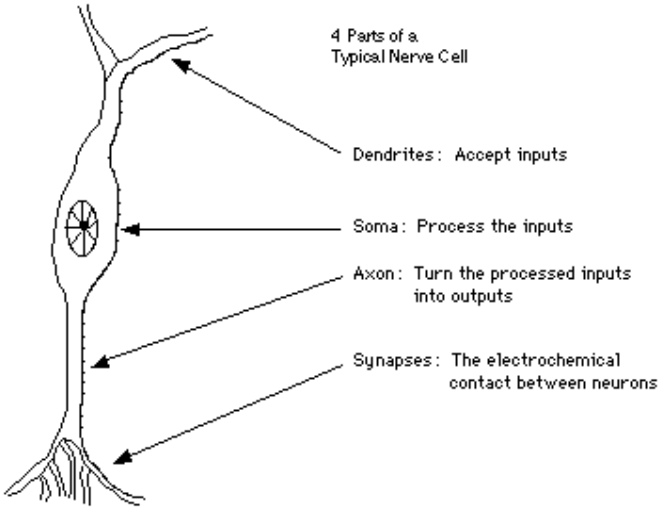

Fig. 2.1 A Simple Neuron.

Recent experimental data has provided further evidence that biological neurons are structurally and significantly more complex than the existing artificial neurons that are built into today's artificial neural networks (Anderson and McNeil, 1992). Neural

network researchers do not seek to recreate the brain. On the contrary, they are seeking an understanding of nature's capabilities for which people can engineer solutions to problems that have not been solved by traditional computing.

To do this, the basic unit of neural networks, the artificial neurons, simulates the four basic functions of natural neurons. Fig. 2.2 shows a fundamental representation of an artificial neuron.

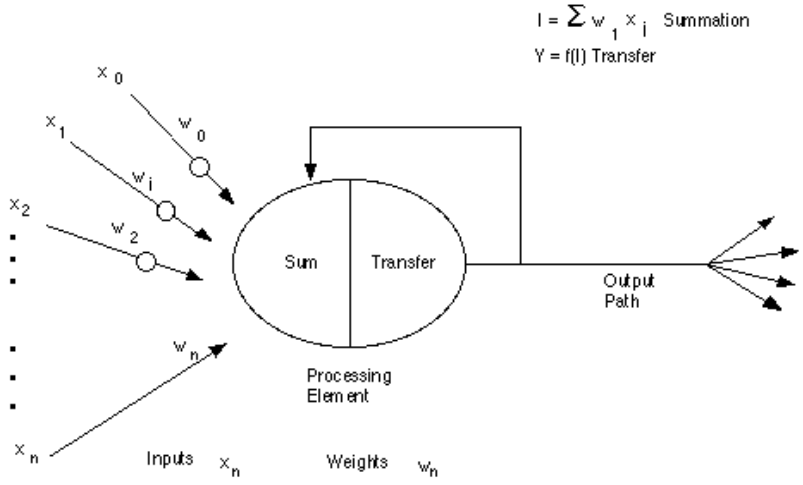

Fig. 2.2 A Basic Artificial Neuron.

In Figure 2.2, various inputs to the network are represented by the mathematical symbol, x(n) and each of these inputs are multiplied by a connection weight. These weights are represented by w(n). In the simplest case, these products are simply summed, fed through a transfer function to generate a result, and then output. This process lends itself to physical implementation on a large scale in a small package. This electronic implementation is still possible with other network structures which utilize different summing functions as well as different transfer functions.

When creating a functional model of the biological neuron, there are three basic components of importance. First, the synapses of the neuron are modelled as weights.

The strength of the connection between an input and a neuron is modelled by the value of the weight. Negative weight values reflect inhibitory connections, while positive values designate excitatory connections. The next two components model the actual activity within the neuron cell. An adder sums up all the inputs modified by their respective weights. This activity is referred to as linear combination. Finally, an activation function controls the amplitude of the output of the neuron. An acceptable range of output is usually between 0 and 1, or -1 and 1.

Mathematically, this process is described in the figure below:

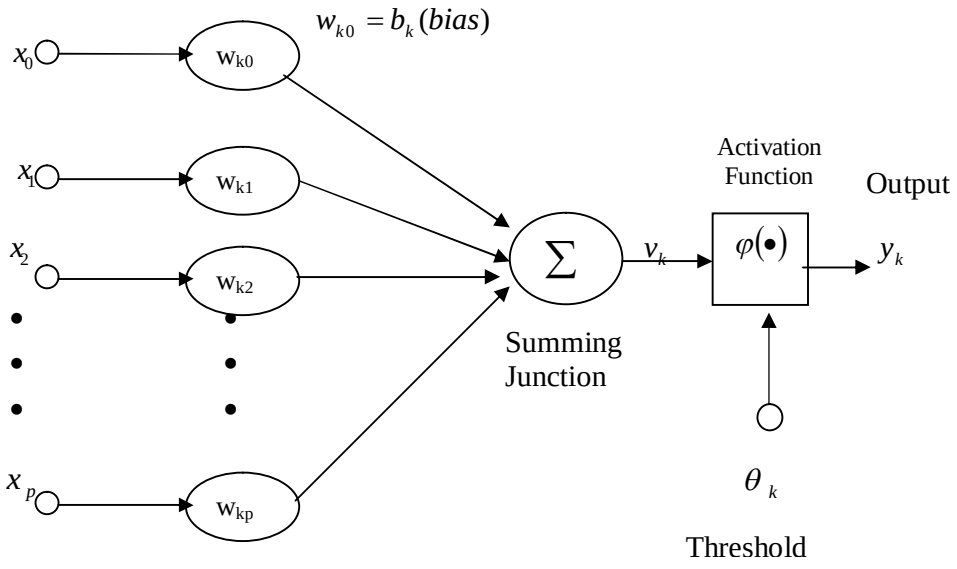

Fig. 2.3: Model of a Neuron

From this model the internal activity of the neuron can be shown to be:

$$v_k = \sum_{j=1}^{p} w_{kj} x_j \qquad \text{.....(2.1)}$$

The output of the neuron, y_k, is therefore the outcome of some activation function on the value of v_k.

The activation function acts as a squashing function, such that the output of a neuron in a neural network is between certain values (usually 0 and 1, or -1 and 1). In general, there are three types of activation functions, denoted by Φ(•). First, there is the Threshold Function which takes on a value of 0 if the summed input is less than a certain threshold value (v), and the value 1 if the summed input is greater than or equal to the threshold value.

$$\varphi(v) = \begin{cases} 1 \, if \; v \geq 0 \\ 0 \, if \; v < 0 \end{cases} \qquad \text{.....(2.2)}$$

Secondly, there is the Piecewise-Linear function. This function again can take on the values of 0 or 1, but can also take on values between that depending on the amplification factor in a certain region of linear operation.

$$\varphi(v) = \begin{cases} 1 & v \geq \frac{1}{2} \\ v & -\frac{1}{2} > v > \frac{1}{2} \\ 0 & v \leq -\frac{1}{2} \end{cases} \qquad \text{.....(2.3)}$$

Thirdly, there is the sigmoid function. This function can range between 0 and 1, but it is also sometimes useful to use the -1 to 1 range. An example of the sigmoid function is the hyperbolic tangent function.

$$\varphi(v) = \tanh\left[\frac{v}{2}\right] = \frac{1-\exp(-v)}{1+\exp(-v)} \qquad \text{.....(2.4)}$$

As already noted above, the artificial neural networks described are all variations on the parallel distributed processing (PDP) idea. The architecture of each neural network is based on very similar building blocks which perform the processing.

2.2 Electronic Implementation of Artificial Neurons

In currently available software packages these artificial neurons are called "processing elements" and have many more capabilities than the simple artificial neuron described above. Figure 2.3 is a more detailed schematic of this still simplistic artificial neuron.

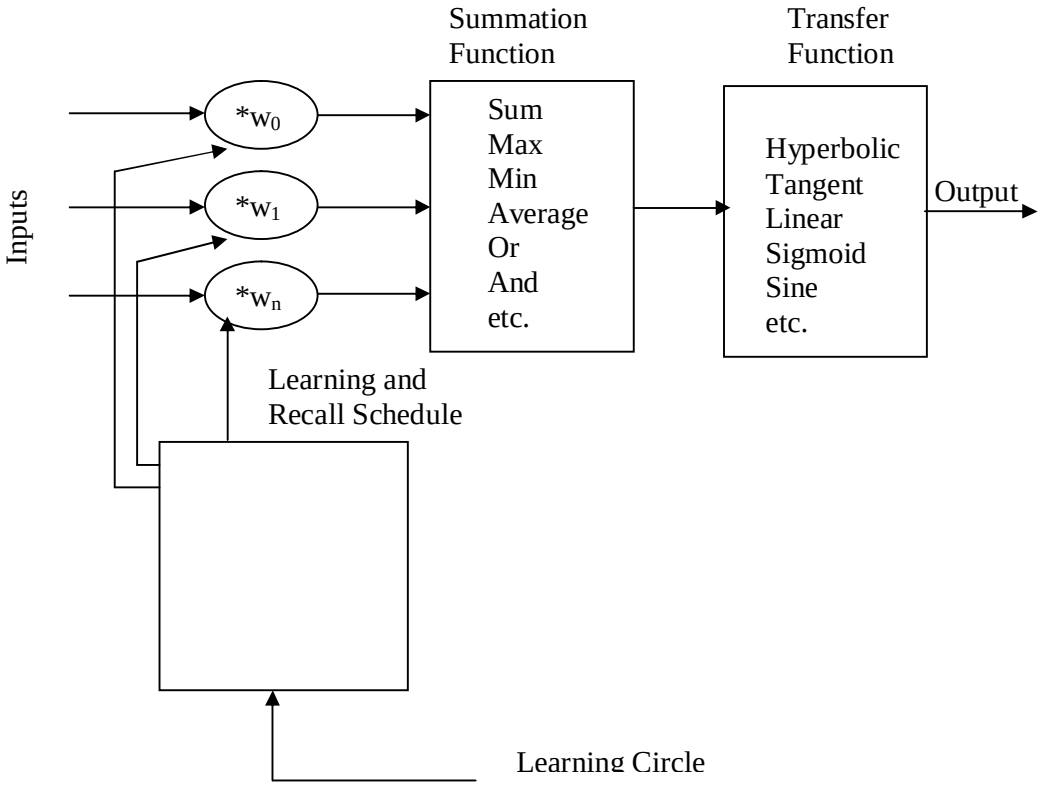

Fig. 2.4: A Model of a "Processing Element".

In Fig. 2.4, inputs enter into the processing element from the upper left. The first step is for each of these inputs to be multiplied by their respective weighting factor (w(n)). Then these modified inputs are fed into the summing function, which usually just sums these products. Yet, many different types of operations can be selected. These operations could produce a number of different values which are then propagated forward; values such as the average, the largest, the smallest, the ORed values, the ANDed values, etc. Furthermore, most commercial development products allow software engineers to create their own summing functions via routines coded in a higher level language (C is commonly supported). Sometimes the summing function is further complicated by the addition of an activation function which enables the summing function to operate in a time sensitive way.

Either way, the output of the summing function is then sent into a transfer function. This function then turns this number into a real output via some algorithm. It is this algorithm that takes the input and turns it into a zero or a one, a minus one or a one, or some other number. The transfer functions that are commonly supported are sigmoid, sine, hyperbolic tangent, etc. This transfer function also can scale the output or control its value via thresholds. The result of the transfer function is usually the direct output of the processing element. An example of how a transfer function works is shown in Fig. 2.5.

This sigmoid transfer function takes the value from the summation function, called sum in the Fig. 2.5, and turns it into a value between zero and one.

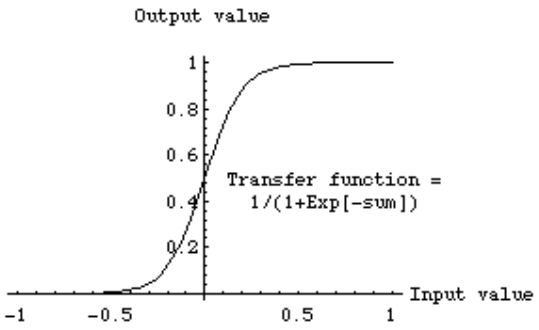

Fig. 2.5: Sigmoid Transfer Function. (*Source: Anderson & McNeil, 1992*)

Finally, the processing element is ready to output the result of its transfer function. This output is then input into other processing elements, or to an outside connection, as dictated by the structure of the network.

All artificial neural networks are constructed from this basic building block - the processing element or the artificial neuron. Variety and the fundamental differences in these building blocks partially cause the implementing of neural networks to be an "art."

2.3 Structure of an Artificial Neural Network

The other part of the "art" of using neural networks revolve around the myriad of ways these individual neurons can be clustered together. Integrated circuits, using current technology, are two-dimensional devices with a limited number of layers for interconnection, unlike biological neural networks that are constructed in a three-dimensional world from microscopic components, thus making them seem capable of nearly unrestricted connections. This physical reality restrains the types, and scope, of artificial neural networks that can be implemented in silicon.

Currently, neural networks are the simple clustering of the primitive artificial neurons. This clustering occurs by creating layers which are then connected to one another. How these layers connect is the other part of the "art" of engineering networks to resolve real world problems.

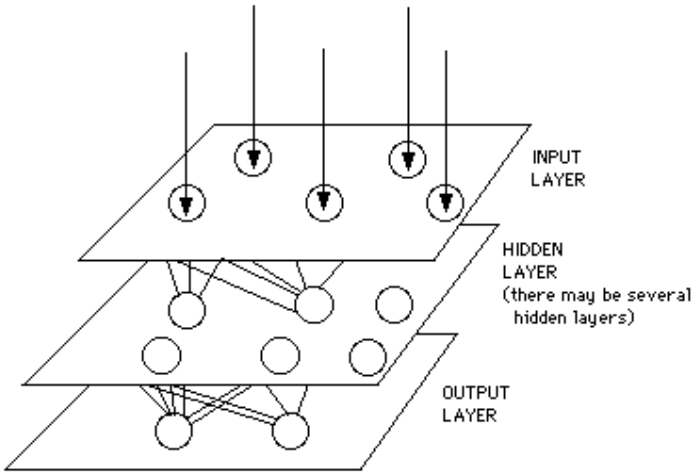

Fig. 2.6: A Simple Neural Network Diagram. (*Source: Anderson & McNeil, 1992*)

Basically, all artificial neural networks have a similar structure or topology as shown in Figure 2.6 In that structure some of the neurons interface to the real world to receive its inputs while others provide the real world with the network's outputs. This output might be the particular character that the network thinks that it has scanned or the particular image it thinks is being viewed. All the rest of the neurons are hidden from view.

One of the easiest ways to design a structure is to create layers of elements. The grouping of these neurons into layers, the connections between these layers, and the summation and transfer functions, make up a functioning neural network.

Although there are useful networks which contain only one layer, or even one element, most applications require networks that contain at least the three normal types of layers - input, hidden, and output. The layer of input neurons receives the data either from

input files or directly from electronic sensors in real-time applications. The output layer sends information directly to the outside world, to a secondary computer process, or to other devices such as a mechanical control system. Between these two layers can be many hidden layers. These internal layers contain many of the neurons in various interconnected structures. The inputs and outputs of each of these hidden neurons simply go to other neurons.

In most networks each neuron in a hidden layer receives the signals from all of the neurons in a layer above it, typically an input layer. After a neuron performs its function it passes its output to all of the neurons in the layer below it, providing a feedforward path to the output. These lines of communication from one neuron to another are important aspects of neural networks. They are the glue to the system. They are the connections which provide a variable strength to an input. There are two types of these connections. One causes the summing mechanism of the next neuron to add while the other causes it to subtract. In more human terms one excites while the other inhibits.

Some networks want a neuron to inhibit the other neurons in the same layer. This is called lateral inhibition. The most common use of this is in the output layer. For example in text recognition if the probability of a character being a "P" is .85 and the probability of the character being an "F" is .65, the network wants to choose the highest probability and inhibit all the others. It can do that with lateral inhibition. This concept is also called competition.

Another type of connection is feedback. This is where the output of one layer routes back to a previous layer. An example of this is shown in Figure 2.7.

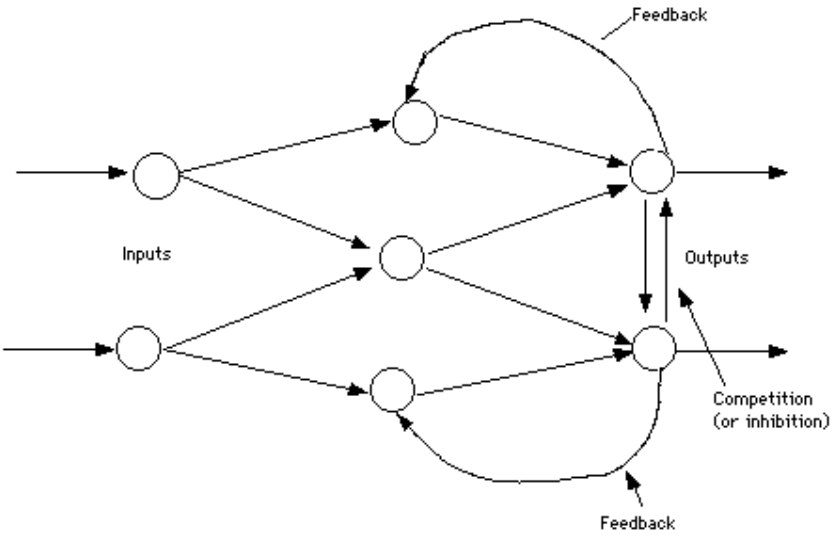

Figure 2.7: Simple Network with Feedback and Competition.

The way that the neurons are connected to each other has a significant impact on the operation of the network. In the larger, more professional software development packages the user is allowed to add, delete, and control these connections at will. By "tweaking" parameters these connections can be made to either excite or inhibit.

2.4 Training an Artificial Neural Network

Once a network has been structured for a particular application, that network is ready to be trained. To start this process the initial weights are chosen randomly. Then, the training, or learning, begins. An ANN must be trained in order for it to classify a target. Training is an iterative process in which the data is fed through the network and the weights are updated to better fit the data. The most prominent method of updating data is backpropagation. Backpropagation updates the weights by comparing the outputs of the current network to the desired output. Using a form of the gradient search methodology, the weights are systematically updated to reduce the error between the output and desired output. These weights can be updated two ways: by instantaneous

updating, that is, updating the weights after each exemplar is sent through the network or by batch training, that is, updating the weights after the entire exemplar set has been sent through the network. The backpropagation algorithm may incorporate the conjugate gradient method for updating the weights. The conjugate gradient method alters the weights of the node inputs using a compromise between the direction of the steepest gradient and the previous direction of change. Rigorous development of the steepest gradient and conjugate gradient methods are available in Looney.

There are two approaches to training - supervised and unsupervised. Supervised training involves a mechanism of providing the network with the desired output either by manually "grading" the network's performance or by providing the desired outputs with the inputs. Unsupervised training is where the network has to make sense of the inputs without outside help.

2.4.1 Supervised Training.

In supervised training, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights which control the network. This process occurs over and over as the weights are continually adjusted. The set of data which enables the training is called the "training set." During the training of a network the same set of data is processed many times as the connection weights are ever refined.

Current commercial network development packages provide tools to monitor how well an artificial neural network is converging on the ability to predict the right answer. These tools allow the training process to go on for days, stopping only when the system reaches some statistically desired point, or accuracy. However, some networks never

learn; either because the input data does not contain the specific information from which the desired output is derived, or there is not enough data to enable complete learning. Ideally, there should be enough data so that part of the data can be held back as a test. Many layered networks with multiple nodes are capable of memorizing data. To monitor the network to determine if the system is simply memorizing its data in some nonsignificant way, supervised training needs to hold back a set of data to be used to test the system after it has undergone its training. (Note: memorization is avoided by not having too many processing elements.)

If a network is unable to solve the problem, the designer then has to review the input and outputs, the number of layers, the number of elements per layer, the connections between the layers, the summation, transfer, and training functions, and even the initial weights themselves. Those changes required to create a successful network constitute a process wherein the "art" of neural networking occurs.

Another part of the designer's creativity governs the rules of training. There are many laws (algorithms) used to implement the adaptive feedback required to adjust the weights during training. The most common technique is backward-error propagation, more commonly known as **back-propagation**.

> To insure that the network is not overtrained, initially, an artificial neural network configures itself with the general statistical trends of the data. Later, it continues to "learn" about other aspects of the data which may be spurious from a general viewpoint.
>
> When finally the system has been correctly trained, and no further learning is needed, the weights can, if desired, be "frozen." In some systems this finalized

network is then turned into hardware so that it can be fast. Other systems don't lock themselves in but continue to learn while in production use.

Generalization is the ability of a network to accommodate variability and produce the correct output vector despite insignificant deviations between the input and test vectors. Generalization may be quantitatively defined for supervised training if the training set is considered to be randomly selected examples from a specific (but unknown) probability distribution. For instance, a network is first trained on such a set, and the resulting eror rate e_1 is measured. Another set of input/output vector pairs is selected from the same distribution, each input vector is applied to the network, the network's response compared to the desired response (the output vector) and another error rate e_2 is calculated. Then, the modulus of the difference between the error rates (i.e. $|e_1-e_2|$) becomes the generalizing ability of the network.

2.4.2 Unsupervised, or Adaptive Training.

The other type of training is called unsupervised training. Here, the network is provided with inputs but not with desired outputs and the system itself must then decide what features it will use to group the input data. This is often referred to as self-organization or adaptation.

At the present time, unsupervised learning is not well understood. It is necessary for life situations where exact training sets do not exist such as those of military action where new combat techniques and new weapons might be encountered. Because of this unexpected aspect to life and the human desire to be prepared, there continues to be research into, and hope for, this field.

One of the leading researchers into unsupervised learning is Tuevo Kohonen, an electrical engineer at the Helsinki University of Technology. He has developed a self-organizing network, sometimes called an auto-associator, which learns without the benefit of knowing the right answer. It is an unusual looking network in that it contains one single layer with many connections. The weights for those connections have to be initialized and the inputs have to be normalized. The neurons are set up to compete in a winner-take-all fashion.

Kohonen continues his research into networks that are structured differently than standard, feedforward, back-propagation approaches. Kohonen's work deals with the grouping of neurons into fields. Neurons within a field are "topologically ordered." Topology is a branch of mathematics that studies how to map from one space to another without changing the geometric configuration. The three-dimensional groupings often found in mammalian brains are an example of topological ordering.

According to Kohonen the lack of topology in neural network models make today's neural networks just simple abstractions of the real neural networks within the brain. As this research continues, more powerful self learning networks may become possible. But currently, this field remains one that is still in the laboratory.

2.5 Network Architectures

All artificial neural networks are based on the concept of neurons, connections and transfer functions, as such, there is a similarity between the different structures or architectures of neural networks. The majority of the variations stems from the various learning rules and how those rules modify a network's typical topology. The number of features, hidden nodes, and outputs in an ANN describes its architecture. In the following sections some of the most common artificial neural networks are outlined, organized in very rough categories of application according to specific applications that each network architecture best matches. Basically, most applications of neural networks fall into the following five categories and are summarized in the table below:

1. prediction
2. classification
3. data association
4. data conceptualization
5. data filtering

Table 2.5: Network Selector Table

Network Type	Networks	Use of Network
Prediction	• Back-propagation • Delta Bar Delta • Extended Delta Bar Delta • Directed random Search • Higher Order Neural Networks • Self-organizing map into Back-propagation	Use input values to predict some output (e.g. pick the best stocks in the market, predict weather, identify people with cancer risks etc).
Classification	• Learning Vector Quantization • Counter-propagation • Probalistic Neural Networks	Use input values to determine the classification (e.g. is the input of the letter A, is the blob of video data a plane and what kind of plane is it)
Data Association	• Hopfield • Boltzman Machine • Hamming Network • Bidirectional Associative Memory • Spation-temporal Pattern Recognition	Like Classification but it also recognizes data that contains errors (e.g. not only identify the characters that were scanned but identify when the scanner is not working properly)
Data Conceptualization	• Adaptive Resonance Network • Self Organizing Map	Analyze the inputs so that grouping relationships can be inferred (e.g. extract from a database the names of those most likely to buy a particular product)
Data Filtering	• Recirculation	Smooth an input signal (e.g. take the noise out of a telephone signal)

(Source: Anderson & McNeil, 1992)

Table 2.5 shows the differences between these network categories and shows which of the more common network topologies belong to which primary category. This chart is not meant to be all inclusive but intended as a guide as there are many other network derivatives. Some of these networks, which have been grouped by application, have been used to solve more than one type of problem. Feedforward back-propagation in particular has been used to solve

almost all types of problems and indeed is the most popular for the first four categories.

The most common ANN is the feedforward multi-layer perceptron. The multi-layer perceptron is comprised of an input layer, hidden layer, and output layer. The input layer is composed of features as defined by the data set. However, there may be information in the data set that is not important to the problem and may cause the ANN's performance to drop.

Looney later concludes that the number of neurons in the hidden layer, as well as the number of output neurons, should be minimally sufficient so as to separate the classes in a generalized manner without introducing extraneous error. Fig. 2.8 provides a generalized picture of the feedforward ANNs utilized in this research.

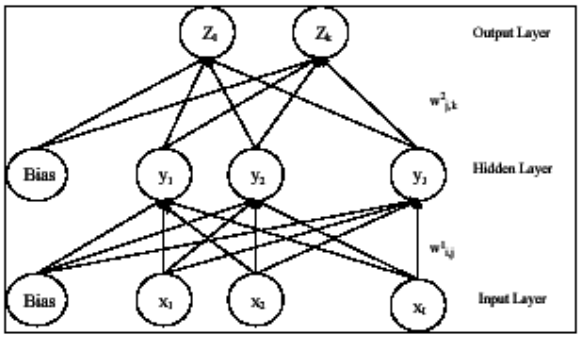

Fig. 2.8: General Feedforward Multi-layer Perceptron

The bias terms are added to the model to shift the function centres at each hidden node to represent the midpoint of the activation function (Looney, 1997). Numerous theories abound on how to initialise the weights prior to training. Looney cites the work of Li and Wu from 1993, which showed "that two different initial weight sets can lead to drastically different convergence

behaviours". However, Looney points out that the results from most of the research depend on a significant number of things that vary from problem to problem (such as, data set, ANN architecture, and the like). Due to his experience, he further concludes that the weights should not be started at too large a value so as to avoid the activation functions getting saturated during the gradient search process.

2.6 **Related Works**

Morris and his colleagues at the University of Minnesota attempted to predict the GPA of University of California students using neural networks (Morris and Gibsson, 2004). To achieve their objectives, they compared three different network topologies of which the average error rate for the network model was used within the range of 8 inputs as the highest and 1 output as the lowest. They obtained a result which showed that the attributes used were inadequate and they proceeded to increase the attributes.

The first network design had three inputs namely: gender, sleep and alcohol; two hidden layers of eight and five nodes; and one output node. The model was trained so that once fed with appropriate variables, data or attributes, it guesses the actual GPAs.

The second model phase of the neural network used in their work contained eight input nodes, which comprised of the hidden layer with 6 nodes, and 4 Boolean output nodes. They used this particular approach to determine if the model can effectively categorise a student's GPA by the 8 given attributes. Each of the output range corresponds to a range that falls within the standard grade

scale between 0 to 1.0; 1.01 to 2.0. 2.01 to 3.0 and 3.01 to 4.0 etc. To train the network to meet the set goal, each of the output nodes was initialised to the correct value using the student's GPA of 3.5 corresponding to {false, false, false, true}. The basic idea here was used to classify given GPA's into A, B, C or D etc. (Chris et.al., 2004)

In the third network, the model also had eight input nodes, two hidden layers of seven nodes each and one output node. The output as in the first network was the actual GPA prediction and the essence of this model was to integrate all the attributes into the prediction process.

The three networks each had its own merits and demerits; the first model only incorporated the three attributes after training the network, it was discovered that there was no correlation between the GPA and the gender, that is, gender is insignificant in the prediction process. The second network showed that there was a low validation rate, therefore, it did not predict the actual GPA, and this may be due to some problems associated with the network. The third network used all the attributes, but the queries showed that there were problems of overlap and noise. In the final analysis, the result showed that the attributes used in the neural networks model was not sufficient to predict GPA; that many other attributes can be added in order to improve the GPA predictions.

In the recent past, there have been comparisons between neural networks and classical statistical modelling as predictors of success or failure in various disciplines. Gorr and his team conducted a comparative study of neural network

and statistical models for predicting college student's grade point averages (Gorr et.el., 1994). Ashby and Kumar (1996) also compared neural networks and discriminant analysis in anticipating default among high yield bonds. Both studies revealed that the neural network serve as a better predictor than the discriminant analysis. The use of neural network as a predictor has increased over the past few years. Cripps (1996) equally used a neural network to predict grade point averages of middle Tennessee State University students. According to Carbone and Piras (1998), the neural network is instrumental in predicting high school dropouts. In Nelson and Henricksen's (1994) study, a neural network used input from the student responses on mathematics placement examination given to incoming students at Ball State University and outputs the mathematics course in which each student should be placed. Hardgrave et.al (1994) at a time used neural network for predicting MBA students' performance for admission in a graduate programme and the study show that the result obtained is promising. Hence the implementation of neural networks as a prediction tool for educational placement and assessment continues to increase.

Sheel and his colleagues also carried out a study on implementing a technique that can be used to classify college student placement into the appropriate mathematics course (Sheel et.al, 2002). In their work, they examined the alternative placement strategies using regression analysis. The accumulative high school grade point average, mathematics SAT, and the final grade in Algebra II were found to be the best predictors of success on a mathematics placement examination. Using these features, entry-level mathematics placement based on networks was later contrasted with discriminant analysis, and proposed

as an alternative to testing. Results obtained showed that neural networks outperformed classical discriminant analysis in predicting the recommended mathematics placement.

As technology continuously progresses, methodologies evolve to enhance abilities to perform arduous tasks more expediently. Utilizing modern computing technologies not only make completing tasks more efficiently, but also often achieve a higher degree of accuracy than human being does. For instance, in classifying students placement test.

The entry-level mathematics courses for placement are as follows: Math 130: Intensive College Algebra, Math 131: College Algebra, Math 133: Trigonometry and Analytic Geometry, and Math 160: Introductory Calculus.

All colleges within Coastal Caroline University, except the Wall School of Business are utilising the examination created by the mathematics department. Students entering the College of Business are assigned to a mathematics class according to their academic achievement in secondary education. The neural networks and discriminant analysis represent techniques that can be employed for membership classification.

The aim of the study was to compare the effectiveness of placement based on neural networks classification and discriminant analysis with the more traditional manual test-based placement. The results for both the neural approach and discriminant analysis were compared to the results obtained by the

mathematics placement test. If a trained neural network or prediction equation based on discriminant analysis yield statistically similar results to the mathematics placement test, then one or the other could be used for entry-level placement.

However, in the series of experiments, supervised feed-forward neural networks were chosen as the network model. The networks were trained using data for entry-level college students grouped according to their performance on the mathematics placement examination. The student's high school grade point average, SAT mathematics score, and high school algebra II score were used in the input layer. The output layer consisted of four node, representing courses into which the students may be placed in the mathematics placement exam. In this experiment, recurrent back-propagation was used to train the network.

In achieving their goal, 458 student records were randomly ordered and assigned to two equal sets. The first group was to train the neural networks and to create the discriminant analysis predictive equation. The second data set was used to test the trained neural networks and the predictive discriminant analysis equation. Both methodologies used the overall high school GPA, the SAT mathematics score, and the final grade in Algebra II as inputs, with the mathematics placement result as the output variable.

The topology chosen for the back-propagation network consist of 3-inputs, 10-hidden layers and 10-output networks, with ten nodes ensembles of weak classifiers generated under identical conditions (Renner and Lacher, 2000).

The recurrent back-propagation neural network significantly outperformed the predictive discriminant analysis equation as a tool for placing incoming freshmen into entry-level mathematics course.

Bijayananda and Srinivasan (2004) in their work titled “Predicting MBA performance for admission decisions is crucial for educational institutions” evaluated the ability of three different models - neural networks, logit and probit to predict MBA student performance in graduate programmes. In the work, neural network technique was used to classify applicants into successful and marginal student pools based on undergraduate GPA, GMAT scores, undergraduate major, age and other relevant data.

Three tables were used for the experiment carried out in the study and the results of the predictions of neural network and statistical models used showed that the neural network model outperformed the traditional statistical models and is a useful tool to predict MBA student performance. The overall results suggest that the classification accuracy (and implied predictive power) of the neural network model is 89.13 percent and that of the logit and probit models are 72.83 percent and 73.37 percent respectively.

The use of a non-linear model such as the neural network allows administrators to specifically incorporate uncertainties into the decision making process. The traditional check list and formula approach do not permit this flexibility. To assess the neural network model’s ability to classify students into successful and

marginal groups, the predictive ability of the neural network model was compared with two statistical models namely logistic regression (logit) and probit (Wright and Palmer (1994) and it was shown that the use of a neural network model can support and potentially improve decision making by MBA directors and deans.

Adebiyi (2006) developed a web-based model for JAMB candidates' admission and placement into Nigerian Universities. The inputs fed into the system at the input layer of the model, were the admission and placement criteria such as: five 'O' level credits required in not more than two sittings, UME subject combination and total score requirements, and University's course subject combination requirements. The sum of the products of the inputs yielded an output whereby, candidates who duly qualified for admission were adequately placed into the courses of their choice while those candidates who qualified but did not meet the ordinary level registration requirement for the course of their choice, are automatically placed in any other course provided there is vacancy in the department. The model also presented an opportunity for students self – screening admission system.

The model however, did not take into consideration other variables such as candidates' performance in the Post UME screening examination, catchment areas and educationally less developed states (elds). Since these variables are part of the standard criteria in the University admission and placement process in Nigeria, Adebiyi's model would deny admission and placement to many candidates from the catchment areas and educationally less developed states,

which are favoured by this new system. This study is necessitated by the need to produce a more standard model that includes these new criteria.

2.7 Discriminant Analysis

Discriminant analysis is a common procedure used in multivariate statistics for classifying data into two clearly defined groups (Pedhauzer, 1982; Johnson and Wichern, 1988). Discriminant analysis is a technique for classifying a set of observations into predefined classes. It attempts to provide a mathematical function that will differentiate between different groupings. The purpose is to determine the class of an observation based on a set of variables known as predictors or input variables. The model is built based on a set of observations for which the classes are known. This set of observations is sometimes referred to as the training set. Based on the training set, the technique constructs a set of linear functions of the predictors, known as discriminant functions, such that

$L = b_1x_1 + b_2x_2 + \ldots + b_nx_n + c$, ………………………………………….. (2.5)

where the b's are discriminant coefficients, the x's are the input variables or predictors and c is a constant.

These discriminant functions are used to predict the class of a new observation with unknown class. For a k class problem k discriminant functions are constructed. Given a new observation, all the k discriminant functions are evaluated and the observation is assigned to class i if the i^{th} discriminant function has the highest value. The estimation of the linear discriminant function has over the years, received much theoretical attention, both in the marketing literature (Dillon 1979; Dillon and Schiffman 1978; Crask and Perreault 1977; Morrison 1969: Frank, Massey and Morrison 1965), and in mathematical

statistics (Randles, Broffitt, Ramberg, Hogg 1978; McLachlan 1977; Kraznowski 1975; Fisher and VanNess 1973; Lachenbruch and Mickey 1968).

The concern for estimation has most often focused on the precision with which the discriminant function correctly classifies sets of observations, rather than with methods to better optimize the function itself. Specifically, methodological research has evaluated such areas as the influence of variable selection (Goldstein and Rabinow 1975; Urbakh 1971), bias in categorization (Krishnaswami and Nath 1968; Lachenbruch 1967; McLachlan 1974), and the validation of rules for classifying sets of observations (Dillon and Goldstein, 1978; Hills, 1966). Discriminant Analysis may be used for two objectives: either we want to *assess* the adequacy of classification, given the group memberships of the objects under study; or we wish to *assign* objects to one of a number of (known) groups of objects. Discriminant Analysis may thus have a descriptive or a predictive objective.

A graphical representation from Dillon and Goldstein is provided in Fig. 2.5 to demonstrate discriminant analysis. Two groups, A and B, are defined by the variables X_1 and X_2. By solving $\mathbf{Y} = \mathbf{b}_T\mathbf{X}$, where **Y** is a *1 x n* vector of discriminant scores, **b'** is a *1 x p* vector of discriminant weights, and **X** is a *p x n* matrix of the independent variables, a discriminant score is obtained (Dillon and Goldstein, 1984). The discriminant score is located on the discriminant function line and an individual is classified according to the distributions A' and B'.

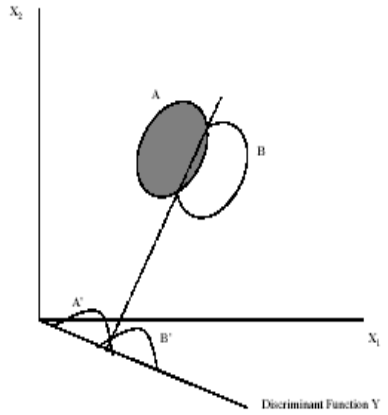

Fig. 2.9 – Discriminant Analysis Illustration

The first step in discriminant analysis is to test if the data is multivariate normal. The underlying theory for most procedures in discriminant analysis assumes the data is multivariate normal and that the variance-covariance matrix of the independent variables in each of the two groups is the same. Because of the difficulty of determining multivariate normality, it is standard to test the individual variables for univariate normality (Andrews, 1973).

CHAPTER THREE

SYSTEM ANALYSIS AND MODELLING

3.1 The Student Selection Solution

The selection of students from competing applicants is a complex decision making process, that often requires a comprehensive evaluation of the applicant's performance. There is a need for consideration of multiple selection criteria, and, the presence of subjective assessments and imprecision results in the use of fuzzy and imprecise data. Juan Carlos Leyva López (2004) formulated the student selection process as a multicriteria decision analysis problem, using the ELECTRE III methodology to construct a fuzzy outranking relation. He then used a genetic algorithm to exploit it and to obtain a ranking in decreasing order of preference.

Multicriteria analysis provides an effective framework for solving problems with multidimensional characteristics. The approach based on fuzzy outranking relations is adequate for dealing with situations in which imprecision and subjectiveness are present (Rogers et al., 2000). Multicriteria analysis is widely used for selecting or ranking alternatives in relation to multiple criteria (Roy, 1996; Vanderpooten, 1990). Given the complexity of the student selection and placement problem, a major aspect of this research is to determine a computing technique that can best be employed in solving the problem.

Statistical procedures, such as discriminant analysis and regression analysis are traditionally used for predicting potential academic success of applicants (Graham, 1991). The predictive validity study may help make admission or

selection procedures more efficient and effective (Powers and Lehman, 1983; Dobson et al., 1999; Lievens and Coetsier, 2002). However, the selection criteria used in higher education admission processes varies widely among programs and no consistent conclusions can be reached on the predictive values of these criteria (Wilson, 1999). This may partly be due to the fact that the predictive validity of the selection instruments is not in itself sufficient for an assessment of the validity of a selection, although it can be a critical factor (Wolming, 1999). The problem is that of both prediction and classification. The work of Bijayananda and Srinivasan (2004), established the fact that artificial neural models has outperformed the other statistical models in predicting student performance and classifying applicants into successful and marginal student pools based on given data sets. Other similar works hold this same premise (Gor et.al, 1994; Nelson and Henricksen, 1994; Ashby and Kumar, 1996; Cripps, 1996; Carbone and Piras, 1998). Hence, artificial neural network is adopted in this research as a multicriteria analysis tool which can adequately handle the multidimensional characteristics of the student selection problem

In this research, prediction is not the stated purpose for the student selection problem. The selection of applicants is made on the grounds of the candidates' merits (performance evaluation) assessed by an examination process and his academic background, based on a given set of criteria in accordance with the requirements of the undergraduate academic programme of any discipline in the University.

This chapter presents a model whereby selection is based on the candidates' merits assessed by the Joint Admissions and Matriculation Board (JAMB) coupled with the performance of the candidates (in not more than two sittings) in ordinary level examinations such as the West African School Certificate (WASC), the National Examination Council (NECO) and in other academic background based on a given set of criteria in accordance with the requirements of the undergraduate academic programmes in all disciplines.

3.2 The Model

Let the courses applied for by candidates be represented by $c_1, c_2, \ldots, c_k$ and the UME subjects requirements, the JAMB's UME and University's Post UME subjects sat for by the candidates x_i for the course c_k be represented by U_{kj}, J_{ij} and P_{kj} respectively. Also, let the ordinary level subjects-combination admission requirements for the course c_k be represented by O_{kj} and the ordinary level grades obtained by the candidate x_i be represented by A_{ij} and B_{ij} resepectively in not more than two sittings such that $i, j, k = 1, 2, \ldots, n$. Similarly, let the catchment areas and the educationally less developed states (elds) that are considered by a given University be represented by sets $\{y\}$ and $\{z\}$ respectively.

U_{kj}, J_{ij}, L_{kj}, A_{ij}, *and* B_{ij} could be represented as partition matrices with c_k as follows:

$$U_{kj}\begin{bmatrix} c_1 & u_{11} & u_{12} & \ldots & u_{1q} \\ c_2 & u_{21} & u_{22} & \ldots & u_{2q} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ c_k & u_{p1} & u_{p2} & \ldots & u_{pq} \end{bmatrix}; u_{kj} = \begin{cases} 1, if\ u_{kj}\ is compulsory\ for\ c_k \\ 2,3\ if\ u_{kj}\ is\ a\ group\ of\ subjects\ for\ c_k \\ 0, if\ u_{kj}\ is\ not\ required\ for\ c_k \end{cases}$$

$$J_{ij} = \begin{bmatrix} c_1 & j_{11} & j_{12} & \ldots & j_{1q} \\ c_2 & j_{21} & j_{22} & \ldots & j_{2q} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ c_k & j_{p1} & j_{p2} & \ldots & j_{pq} \end{bmatrix}; j_{ij} = \begin{cases} 1, if\ j_{ij}\ is\ registered\ and\ sat\ for\ by\ x_i \\ \\ 0, if\ j_{ij}\ not\ registered\ for \end{cases}$$

$$O_{kj} = \begin{bmatrix} c_1 & o_{11} & o_{12} & \ldots & o_{1q} \\ c_2 & o_{21} & o_{22} & \ldots & o_{2q} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ c_k & o_{p1} & o_{p2} & \ldots & o_{pq} \end{bmatrix}; l_{kj} = \begin{cases} 1, if\ o_{kj}\ is\ compulsory\ for\ c_k \\ 2,3\ if\ o_{kj}\ is\ group\ of\ subjects\ for\ c_k \\ 0, if\ o_{kj}\ not\ required\ for\ c_k \end{cases}$$

$$P_{kj} = \begin{bmatrix} x_1 & p_{11} & p_{12} & \ldots & p_{1t} \\ x_2 & p_{21} & p_{22} & \ldots & p_{2t} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_i & p_{q1} & p_{q2} & \ldots & p_{qt} \end{bmatrix}; p_{kj} = \begin{cases} 1, if\ \mathrm{p}_{kj}\ is\ compulsory\ for\ x_i \\ 2,3\ if\ \mathrm{p}_{kj}\ is\ group\ of\ subjects\ for\ x_i \\ 0, if\ \mathrm{p}_{kj}\ not\ required\ for\ x_i \end{cases}$$

$$A_{ij} = \begin{bmatrix} x_1 & a_{11} & a_{12} & \ldots & a_{1q} \\ x_2 & a_{21} & a_{22} & \ldots & a_{2q} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_i & a_{p1} & a_{p2} & \ldots & a_{pq} \end{bmatrix}; a_{ij} = \begin{cases} 1, if\ 2 \le a_{ij} \le 4\ (a_{ij}\ passed\ at\ credit\ level\ by\ x_i) \\ \\ 0, if\ a_{ij} < 2\ (a_{ij}\ failed\ or\ not\ attempted\ \mathrm{by}\ x_i) \end{cases}$$

$$B_{ij} = \begin{bmatrix} x_1 & b_{11} & b_{12} & \ldots & b_{1q} \\ x_2 & b_{21} & b_{22} & \ldots & b_{2q} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ x_i & b_{p1} & b_{p2} & \ldots & b_{pq} \end{bmatrix}; b_{ij} = \begin{cases} 1, if\ 2 \le b_{ij} \le 4\left(b_{ij}\ passed\ at\ credit\ level\ by\ x_i\right) \\ \\ 0, if\ b_{ij} < 2\left(b_{ij}\ failed\ or\ not\ attempted\ by\ x_i\right) \end{cases}$$

These matrices are partitioned sparse matrices because for instance, for course c_k, potential candidates registers and sits for only four UME subjects even though there are twenty-one subjects in all.

The UME subjects requirements for course c_k is a row vector $u_{pq} = \lfloor u_{p1} \quad u_{p2} \quad ... \quad u_{pz} \rfloor$ and $u_{pq} \subseteq U_{kj}, q = 1,2,..,t.$ Also, the potential candidate x_i's UME subjects for the course c_k is a row vector $j_{ip} = [j_{i1} \quad j_{i2} \quad ... \quad j_{im}]$ and $j_{ip} \subseteq J_{ij}, p = 1,2,..,n.$ The Post UME subjects requirements for candidate x_i is also a row vector $p_{qt} = \lfloor p_{q1} \quad p_{q2} \quad ... \quad p_{qt} \rfloor$ and $x_i, p_{qt} \subseteq P_{kj} = 1,2,...z$. The ordinary level subjects combination requirement for course c_k is also a row vector $o_{kq} = \lfloor o_{k1} \quad o_{k2} \quad ... \quad o_{kq} \rfloor$ and $o_{kz} \subseteq o_{kj}, q = 1,2,..,r.$ In the same vein, the ordinary level subjects grades obtained by potential candidates in either of first or second sitting are also vectors such that $a_{ij} \subseteq A_{ij}$, and $b_{it} \subseteq B_{ij}$ respectively. Fig. 3.1 shows the artificial neural networks for this model. The weights to the models are as shown in the sparse matrices above and the model is represented in figure 3.1 below.

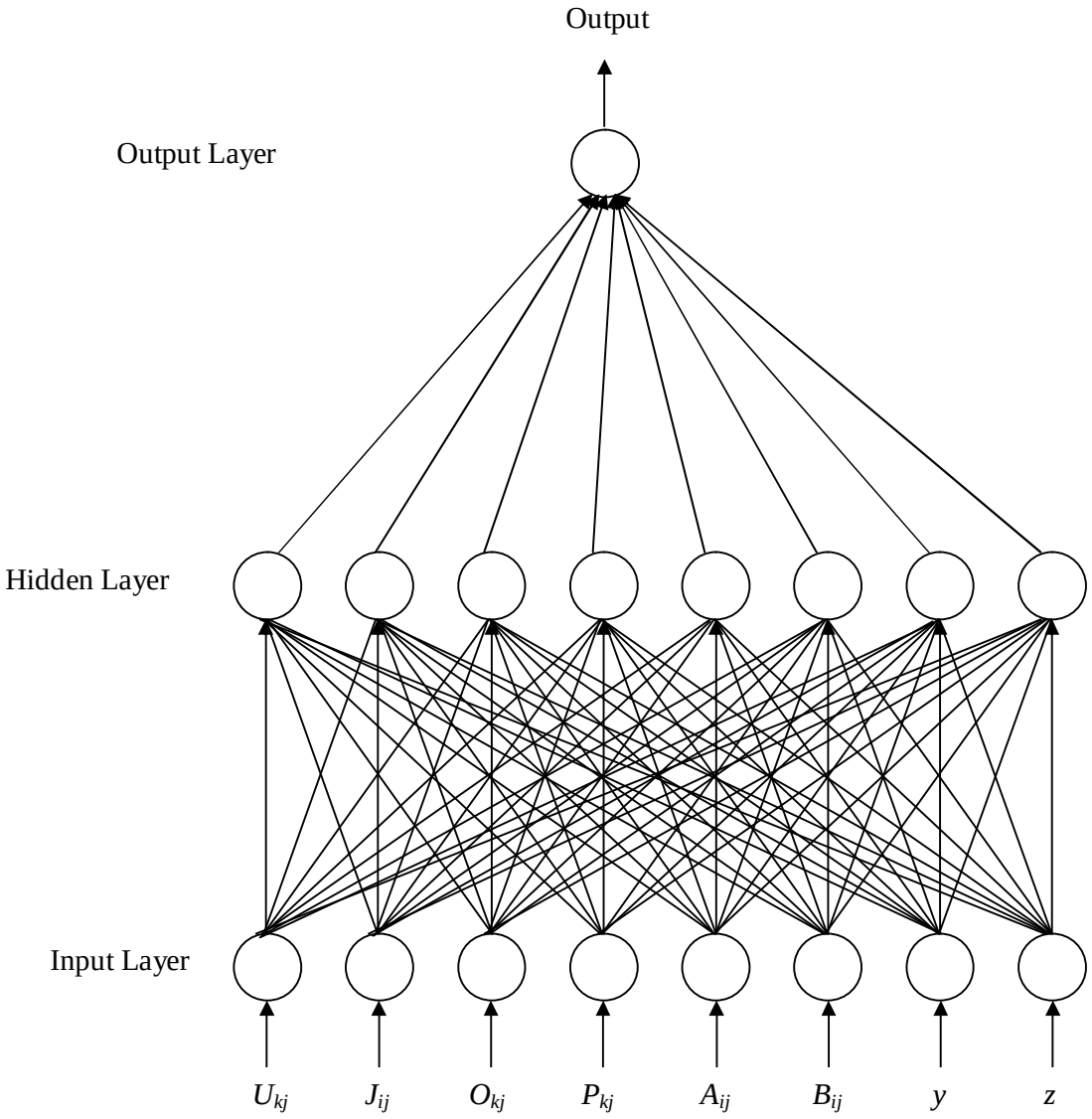

Figure 3.1: The Artificial Neural Networks Model

For each of the potential candidate x_i that applied for the course c_k, let s = 1, 2, 3 represent compulsory UME subjects and other subject groups from which candidates are expected to select from respectively and $T(s)$ represent the total number of subjects sat for at the JAMB UME examinations.

Therefore, let r equals total number of subjects sat for, for each $U_{kj} = s$ and s = 1 (compulsory subjects) such that $T(s) \leq 4$; if

$$U_{kj} \neq J_{ij} \quad 3.1$$

process next potential student's data for the course c_k, otherwise

$$X_r = M(J_{ij}),\ \text{r} = 1, 2, \ldots, 4. \quad 3.2$$

where M is a function that returns the mark obtained by the candidate in the JAMB UME subject J_{ij}. Process the next $U_{kj} = s$.

If $T(s) < 4$ (total number of subjects sat for), then there exists $s = 2, 3$ such that $U_{kj} = s$ where s represents other subject groups from which candidates ought to have selected $T(s)$ from. Therefore, let $m = s = 1, 2, \ldots, T(s)$ for each $s = 2, 3$; if $T(s) = 0$, select next s otherwise for each $U_{kj} = s$, obtain

$$X_r = M(J_{ij}),\ \textit{iff}\ U_{kj} = J_{ij}. \quad 3.3$$

If $T(s) - m > 0$, select the next $U_{kj} = s$. Let the potential student's cut-off score = $scut\_off_i$. If $m = 0$, select next potential candidate's data for the course c_k. If $\sum_{t=1}^{r} I(t) < 4$, select next s otherwise obtain the $scut\_off_i$ as follows

$$scut\_off_i = \begin{cases} 1,\ 160 \leq \sum_{r=1}^{4} X_r \leq 400 \\ 0,\ otherwise \end{cases} \quad 3.4$$

where $\sum_{r=1}^{4} X_r$ represents the total scores obtained by the candidates x_i for course c_k.

Let $pscore = P_{kj}$ be the scores obtained by candidates from the Post UME and let *pufac* represent the percentage cut-off mark for the Post UME scores in each of the faculties.

For candidates x_i whose $scut\_off_i = 1$ and $pscore \geq pufac$, let the percentage contributions of UME examination be *U%* and Post UME be *P%* respectively such that $U\% + P\% = 100\%$, then

$$overallscore = U\% \times \sum_{r=1}^{4} X_r + P\% \times \sum_{i=1}^{4} P_{kj} \qquad 3.5$$

Define $r = 0$, $z = 0$, $y = 1, 2, \ldots, n$, for each $O_{ky} = s = 1$ (compulsory ordinary level subjects) such that $N(s) \leq 5$. Let $F(e)$ be a function that returns 1 *iff* a_{iy} or $b_{iy} = O_{ky}$ otherwise it returns 0. If

$$O_{ky} \neq F(a_{iy}) \vee F(b_{iy}) \qquad 3.6$$

process next potential candidate's data for the course c_k, otherwise $z = z + 1$ and process the next O_{ky}.

If $N(s) < 5$ (total number of ordinary level credits obtained), then there exist $s = 2, 3$ such that $O_{ky} = s$, where s represents other subjects groups where potential student is expected to have O/L credits from. Therefore, let $m = 0$, for each $s = 2, 3$, if $N(s) = 0$, select next s otherwise for each $O_{ky} = s$,

$$z = z + 1,\ m = m + 1,\ iff\ O_{ky} = F(a_{iy}) \vee F(b_{iy}) \qquad 3.7$$

If $N(s) - m > 0$, select next $O_{ky} = s$. If $m = 0$, select next potential candidate's data for the course c_k. If $\sum_{t=1}^{z} I(t) < 5$, select next s. For $z = 5$,

$$adm_i = 1 \qquad 3.8$$

Select the next potential candidate i for the course c_k otherwise select the next course c_k.

Arrange the overall scores for x_i in descending order to determine the candidates that are admit-able on merit, catchment areas and educationally less developed states (elds). Let T_k = Target number of candidates to be admitted for course c_k

and let *%merit* = 50%; %*catchment* = 30%; and *% elds* = 20% such that *%merit* + *%catchment* + *%elds* = T_k. The *%catchment* and *%elds* admission must be evenly distributed among the concerned states.

Therefore, let *y* and *z* represent the University's Catchment areas and the Educationally Less Developed States, respectively. Using the Ahmadu Bello University, Zaria, admissions and placement guideline, the catchment area can be taken care of by the set

$$y = \begin{Bmatrix} Adamawa, Bauchi, Benue, Gombe, Jigawa, \\ Kaduna, Kano, Kat\sin a, Kebbi, Kogi, \\ Kwara, Nasarawa, Niger, Plateau, Sokoto, \\ Taraba, Yobe, and Zamfara \end{Bmatrix}$$ and the educationally less

developed states (elds) which are applicable to candidate x_i for course c_k is the set $z = \{y, Ebonyi, Rivers\ and\ CrossRiver\}$

For $x_i(state) \subseteq y$ and $x_i(state) \subseteq z$, the first of *%merit* are selected automatically. From *%merit* + 1 to *N*, if $x_i(state) \subseteq y$, and *%catchment* = 0, then $x_i = adm_i = 1$. From *%merit* + 1 to *N*, if $x_i(state) \subseteq z$ and *%elds* = 0, then $x_i = adm_i = 1$.

Finally, for placing students whose $adm_i \neq 1$, select each of the students whose $adm_i \neq 1$, for each of the courses c_k, if

$$160 \leq \sum_{r=1}^{4} X_r \leq 400 \qquad 3.9$$

apply equations 3.1 through 3.3 and equations 3.6, if equation 3.7 holds, place the candidate into the course and select next course c_k.

For candidates' self evaluation the entire equations 3.1 to 3.8 are applied to evaluate individual candidate's results against the admission requirements in order to determine whether he qualifies or not. However, it is also possible to perform the candidate's self evaluation process in two stages. The first stage allows the application of equations 3.1 to 3.5 to the candidate's UME subjects' scores. This stage is useful for candidates awaiting their ordinary level results. Later, the if the candidate meets the admission requirements and the O/L results are now available, then equations 3.6 to 3.8 will only be applied on the ordinary level results as the system keeps the outcome of equation 3.5 in a database table.

The internal working of the above model is represented in the figure 3.2.

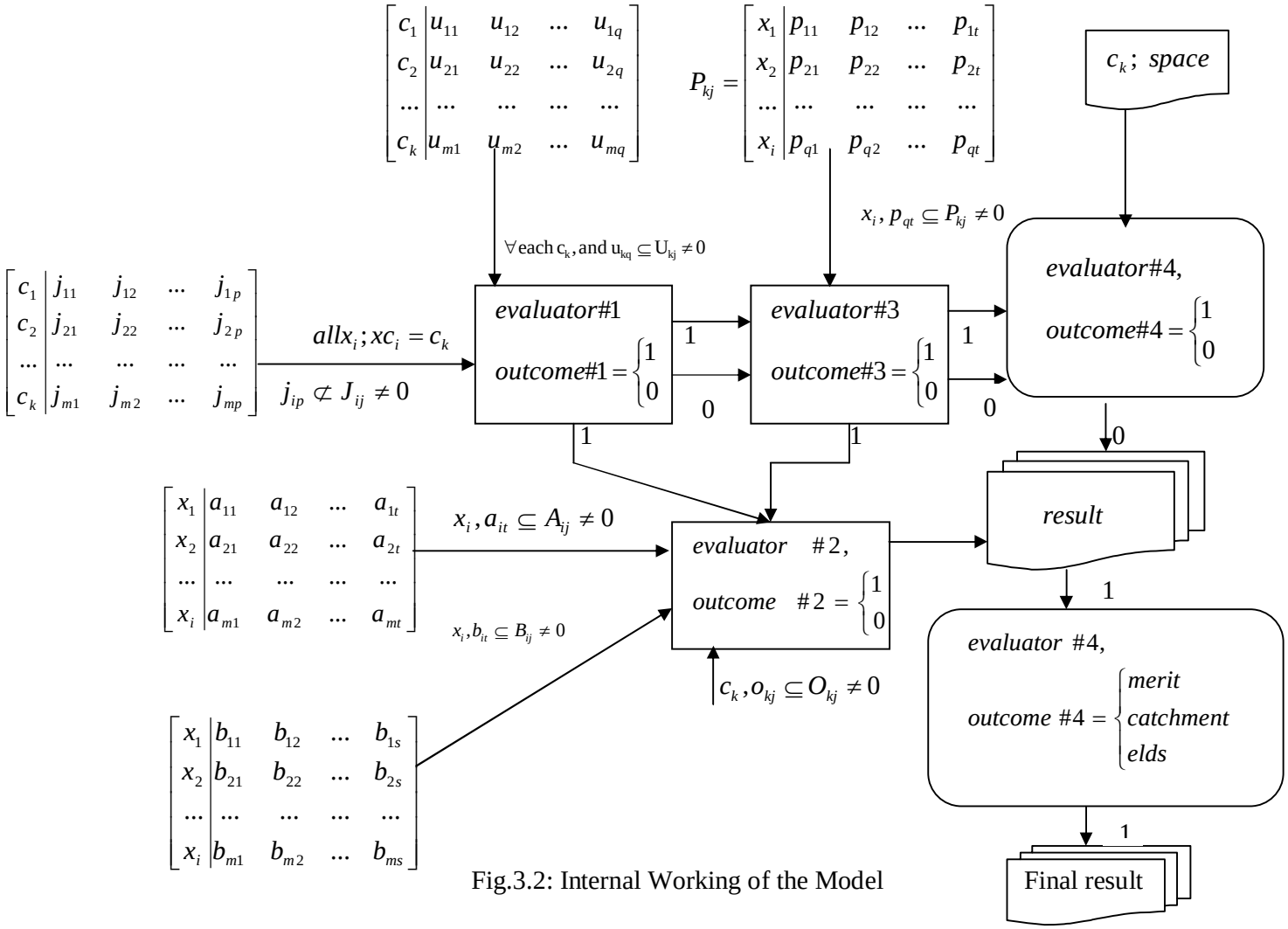

Fig.3.2: Internal Working of the Model

CHAPTER FOUR

SYSTEM DESIGN AND IMPLEMENTATION

4.1 Software Platform for Implementation

The technological approach for the development of the system is based on AMP (Apache, MySQL, and PHP) open source solution (Adewale, 2006). Other solutions such as Microsoft's Internet Information Server (IIS) are popular, but they lack security and attract potential higher cost of hardware and maintenance, which keep them out of reach of many small organizations. Rather than spending millions of dollar on licenses and administrative costs for inferior products, one can choose to run a free software solution. Speed is another significant factor. The system being designed needs to respond quickly and remain snappy throughout the user's experience. With proper coding techniques, PHP is many times faster than Microsoft's ASP or Sun Microsystems' Java platform.

The Web works off a client/server architecture. Applications developed with MySQL and PHP make use of a single client called the Web browser. For any application to run on the Web, there must be a *Web server* which is an application that enables communication with the browser, *a relational database server*, which stores all the information required by the Web server, *a programming language* needed to broker request between the Web server and the database server and to perform programmatic tasks on the information that comes to and from the Web server, and off course, *an operating system* that will interface between the servers and the programming language. A robust

operating system such as Windows NT/2000 or Unix is better suited for running Web applications and they account for more than 90% of all the Web servers on the Web.

Both MySQL and PHP run on Windows and Unix platform. However, in the real world, most PHP/MySQL applications run off on some version of Unix be it Linux, BSD, Irix, Solaris, HP-UX, or one of the other flavours. Unix has the advantage of inherent stability, it is free and run on standard PC hardware. So since PHP, MySQL and Unix are open source applications, developers naturally tend towards the Unix platform. On the other hand, Windows NT/2000 has the advantage of inertia and any organization that has been developing with Microsoft products for years will very easily use NT/2000 or even Windows 95/98 for her PHP/MySQL applications.

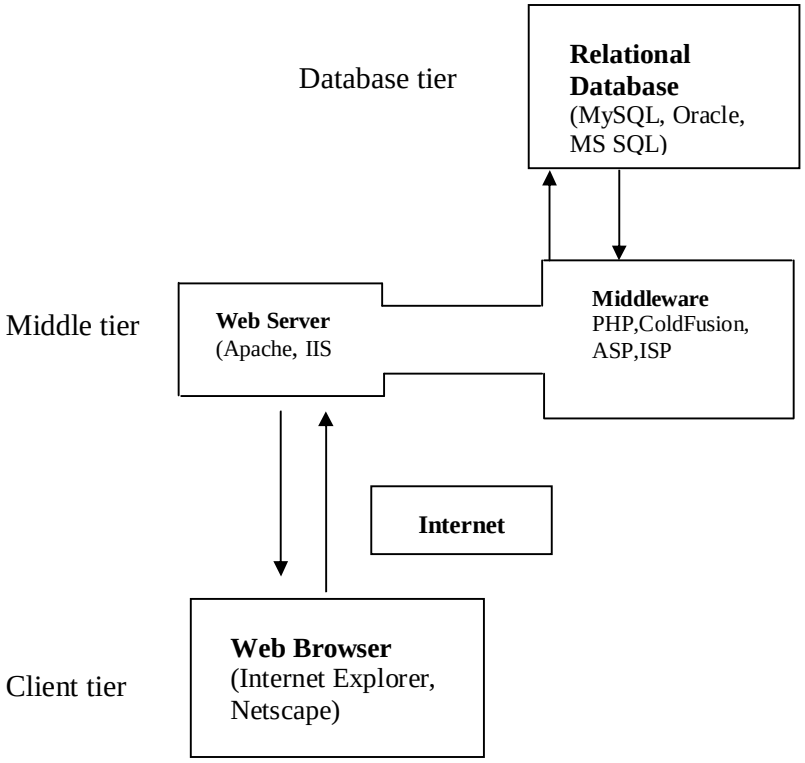

Fig. 4.1: Architecture of Web Applications

The term open source was coined in 1998 after Netscape decided to publish the source code for its popular navigator browser. The open source movement is a collaboration of some of the finest minds in computer programming. By allowing the open exchange of information, programmers from all over the world contribute to make a truly powerful and efficient piece of software available to everyone. Through the contributions of many people to the publicly available source code, bugs get fixed, improvements are made, and a good software program becomes a great one over time. Open source programs are very good because they are free; cross-platform and technology-neutral; not restricting other software; and embrace diversity (well coordinated diversity of minds and cultures simply produces better results).

AMP is a proven, efficient set of software that works well as a system. The open architecture of each of these elements allows for smooth and seamless integration with one another and results in a powerful combination. Early adopters of these technologies back in 1997 were seen as radical, but today the open source movement is on the rise, and both large and small enterprises are adopting the AMP method of development. Steering away from the high costs of implementing licensed server and client software is becoming increasingly beneficial because the stability of each application is surpassing that of its far more expensive brethren. In recent events, some governments have decided to make the jump to open source software, touting its reliability, efficiency, and substantial cost savings over proprietary solutions.

Besides reaching this dependability level capable of being embraced by entire governments, the major advantage seen by AMP adopters is *speed*. Each component of AMP exhibits benchmarks that far exceed those of their competitors.

APACHE The Apache web server developed by the Apache Software Foundation is an open source web server solution that is packed with features. It is extremely fast, amazingly stable and works best with the Linux operating system. Apache is rapidly growing in popularity and is currently the number one web server solution according to Netcraft surveys. It has held the number one position, with a large margin, for more than eight years now.

With the Apache web server, one can create *virtual hosts* that enable one to run multiple websites on a single server, and it has many more awesome features. PHP most often will run as an Apache extension, known as an Apache module. Although Apache works best on Unix systems, some versions of the Apache web server is available for the Windows environment; however, such system suffers from decreased performance because of Microsoft's memory management known as *leaks* and architectural differences.

Databases are managed by a database management system (DBMS) or database server. A database server supports a database language to create and delete databases and to manage and search data. The database language used by almost all database servers is Structured Query Language (SQL), a set of statements

that define and manipulate data. Relational Database Management Systems (RDBMSs) provide a great way to store and access complex information.

MySQL database server is a powerful, robust, open source relational database manager that enables storage and retrieval of data with a scripting language such as PHP. Its design is suited to managing databases that are typical of most web database applications. MySQL runs as a server (daemon) process or service, like Apache or IIS, and supports several different clients including a command-line interpreter and a PHP function library. Various types of data, such as Boolean operators, text, integers, images, binary digits, and blobs (binary large objects) may be stored quickly and efficiently with minimal effort. Using a database is important for creating dynamic sites. The term *dynamic site* is derived from being able to utilise a single page of code to display different information based on user's interaction. This would be virtually impossible without the use of a database and a scripting language such as PHP to manipulate the data.

Early in its history, MySQL occasionally faced opposition due to its lack of support for some core *structured query language* constructs such as subselects and foreign keys. Ultimately, however, MySQL found a broad, enthusiastic user base for its liberal licensing terms, perky performance, and ease of use. Its acceptance was aided in part by the wide variety of other technologies such as PHP, Java, Perl, Python, and the like that have encouraged its use through stable, well-documented modules and extensions. MySQL has not failed to

reward the loyalty of these users with the addition of both subselects and foreign keys as of the 4.1 series.

MySQL is packed full of features such as data replication, table locking, query limiting, user accounts, multiple databases, persistent connections, and as of MySQL 5 stored procedures, triggers, and views. Postgres and Borland's Interbase products are two other major open source relational databases that exist as alternatives to MySQL; but MySQL has grown exponentially in popularity.

PHP is a recursive acronym that stands for *PHP: Hypertext Preprocessor*. The product was originally named *Personal Home Page Tools*. This widely used general purpose scripting language is especially suited for web development and can be used as a standalone binary or embedded into HTML. It is also simply known as a server-side HTML-embedded scripting language and the code is executed on the server. PHP belongs to a class of programming languages known as ***middleware,*** which work closely with the Web server to interpret the requests made from the World Wide Web, process the requests, interact with other programs on the server to fulfil the requests, and then indicate to the Web server what to return to the client's browser. PHP can collect form data, generate dynamic page content, or send and receive cookies, but its strongest and most significant feature is its support for a wide range of databases. With PHP, writing a database-enabled web page is quite easy. Other languages that perform similar functions as PHP are Microsoft's Active Server Pages (ASP), Macromedia's ColdFusion, Perl, and Sun's Java Server Pages. PHP however is a

good choice because it has the advantages of being free and open for improvements; is flexible for integration with HTML; is suited to complex projects such as math, sorting, creating PDF documents, and sending email; it is fast at running scripts; and is platform- and operating- system portable. According to Netcraft, PHP code can now be found in approximately sixteen million web sites. PHP5's newly rebuilt object model brings PHP more in line with other object-oriented languages such as Java and C++, offering support for features such as overloading, interfaces, private member variables and methods, and other standard OOP constructions.

Due to the combined advantages of speed, ease of use, cross-platform, accessibility to functions, processing power, cost-effectiveness, flexibility, constant improvement and International acceptance, Apache, MySQL and PHP are the open source software running on windows operating system used in the development of the university admission and placement system.

4.2 The University Admission And Placement System Design

The university admission and placement system is built around a *three-tier architecture* model (Adewale, 2006). This is shown in Fig. 4.2. At the base of an application is the *database tier,* consisting of the *database management system* that manages the database holding the data that users create, delete, modify, and query and MySQL relational database is used to provide the required functionalities.

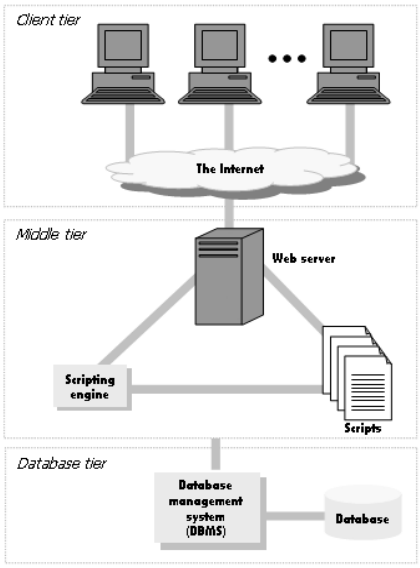

Fig. 4.2: Three-tier Database Architecture Model

A relation is customarily referred to as a file and generally perceived and represented by a set of structured tuples. Each tuple of a relation corresponds to a record in a file and attributes correspond to fields within a record. The general form of a relation is given by $R[A_1, A_2, \ldots, A_{k+1}, \ldots, A_{n-1}, A_n]$. The name of the relation is represented by R, the set $\{A_{i,\ i=1,2,\ \ldots,\ n}\}$ represents the attributes of the relation R. The database objects used in the university admission and placement system are:

course_ume_req[course_code, u_{k1}, u_{k2}, …, u_{k21}];

course_o_level_req[course_code, o_{k1}, o_{k2}, …, o_{k21}];

cand_ume_sat[cand_ume_no, course_code, state_name, j_{i1}, j_{i2}, …, j_{i21}];

cand_first_sit[cand_ume_no, olev_exam_no, a_{i1}, a_{i2}, …, a_{i21}];

cand_second_sit[cand_ume_no, olev_exam_no, b_{i1}, b_{i2}, …, b_{i21}];

cand_pume_req[cand_ume_no, cand_pume_no, p_{k1}, p_{k2}, …, p_{k21}];

cand_process_table[cand_ume_no, cand_pume-no, course_code, ume_score, scut_off, admin,admrem, placement, plmrem, adm_course];

quota_cut_off[course_code, quota, cut_off, batch].

Built on top of the database tier is the complex *middle tier*, which contains most of the application logic and communicates data between the other tiers. The web server is Apache and it is running under Windows operating system which is used to achieve a secured client-server communication. The scripting engine uses server-side PHP functions to communicate with the database. PHP scripts are used to co-ordinate all the procedures in the system. PHP handles data which are passed from the html forms in the way that structured query language forms are sent to the database and then results of the queries are processed and passed back in an html document format. On top is the *client tier*, usually web browser software that interacts with the application. The formality of describing most web database applications as three-tier architectures hides the reality that the applications must bring together different protocols and software. When we use the term *the web*, we mean three major distinct standards and the tools based on these standards: the Hypertext Markup Language (HTML), the Hypertext Transfer Protocol (HTTP), and the TCP/IP networking protocol suite. HTML works well for structuring and presenting information using a web browser application. It is the resource (may be the HTML document, image or a program output) usually carried as a HTTP response from the server back to the web browser. TCP/IP is an effective networking protocol that transfers data between applications over the Internet and has little impact on web database application developers. The problem in building web database applications is interfacing traditional database applications to the web using HTTP. This is where the complex application logic is needed. HTTP is a component that binds together the three-tier architecture. It allows resource to be communicated and shared over the Web.

The home page of the university admission and placement system is shown in Fig. 4.3. It consists of four main modules: *About*, *Services*, *Administration*, and *How to …* modules. Some other features such as *Admission*, *Submit Scores*, *Self Evaluation*, *Results*, *News* and *ABU* are available on the home page of the system.

Entry point to some of the university admission and placement system requires users' authentication and authorisation because of the sensitivity of the admission process. Since users access the system remotely, therefore a built-in security system forces users to login first.

There are several alternatives to users' authentication. These alternatives include among others HTTP basic authentication and PHP session schemes. In implementing HTTP basic authentication, one either has to properly configure it through web server such as using *.htaccess* file on an apache server or duplicate the functionality through the use of the PHP *Header* function. Although the HTTP mechanism is acceptable, even with a proper set up HTTP authentication has many shortcomings such as lack of security features (inactivity timeout); and developer-friendly benefits (the ability to display secure/non-secure data easily on the same page).

As a result of the above mentioned shortcomings, a PHP session authentication is used in the system to control access to most of the university admission and placement system's sensitive areas while at the same time providing a more

useful and clean implementation. Sessions are a mechanism that allows PHP to preserve state between executions. In simple terms, sessions allow system to store variables from one page and use them on another.

The basic mechanism of session works like this: first PHP generates a unique, thirty-two character string to identify the session. It then passes the value to the browser; simultaneously, it creates a file on the server and includes the session ID in the filename. There are two methods by which PHP can inform a browser of its session ID: by adding the ID to the query string of all relative links on the page, or by sending it as a cookie (a small piece of information that PHP script can store on a client-side machine). When the browser makes a request for another page, it tells PHP which session it was assigned via a URL query string or by returning the cookie. PHP then looks up the file it created when the session was started, and so has access to the data stored within the session.

Once the session has been established, it will continue until it is specifically destroyed by PHP in response to a clicking “Logout”, for example or the session has been inactive for longer than a given period of time (twenty-four minutes by default), at which point it becomes flagged for garbage collection and will be deleted by the next time PHP checks for outdated sessions.

In summary, the basic steps of using sessions are: starting a session, registering session variables, using session variables and deregistering variables and destroying the session.

Because we want to tie a user to session, the user's username and password are stored in a MySQL database table and authenticated against that. There are four main elements to the system user authentication module: user registration, login and logout, changing password and resetting passwords. A valid user or admission officer could perform each of these elements. To register a user, user supplies his user identification number, and the *preferred username*, *password* and *password confirmation* via an html form, and if the user identification number is found in the user table and both password and password confirmation are the same, it then registers the username and password in the user table. Registered users can now login by supplying his username and password details into an html form and submit it, the entries will be processed and the user will be logged on if the authentication is successful. Users are also allowed to change their passwords and in addition to this, the system also deals with the common situation in which a user has forgotten his password.

Some of the main features of the university admission and placement system now follow.

About module mainly consists of rules and procedures guiding the admission into the university system and other stuffs that candidates are expected to know.

Services module consists of four sub-modules that allow potential candidates to submit their UME scores, O/L scores or grades, evaluate themselves against the university admission requirements and finally a sub-module to view admission results. Fig. 4.4 shows an html form that a potential candidate needs to complete if he wish to submit his UME scores for admission process. Fig. 4.5 shows an

html form that a potential candidate needs to complete if he wish to submit his O/L scores or grades in not more than two sittings. Fig. 4.6 presents an html form to potential candidate who desires to evaluate his UME scores and/or O/L grades with the university admission requirements. This form is divided into two parts. The first part enables potential candidate to enter his UME scores while the second part enables him to enter his O/L scores or grades. Upon clicking the *Evaluate* button, JavaScript is invoked to validate entries so as to be sure that the entries are valid. For instance, scores must lie between 1 and 100 and no more. If the entries are okay, *evaluate.php* script is employed to evaluate the candidate against the university admission requirements based on the equations 3.1 to 3.9. It is important to stress that the first part of the form could also be submitted for processing for candidates awaiting their O/L results while the other part may be completed later for final admission. However, for any candidate that meets the university requirements based on self evaluation, his entries would be verified with the various examinations bodies before proper admission could be confirmed to the candidate. Self evaluation sub-module enables a candidate to know whether he will be admitted or not.

Admission results sub-module lets potential candidates view their results but on individual basis. Candidate must supply his UME exam number, and the system uses this identification number to query the admission table.

Administration module requires user authentication and authorisation due to sensitivity of the nature of the admission process. This module consists of options to set UME, O/L admission requirements, extract UME records of

potential candidates from the JAMB UME databases, and validate potential candidates against the university admission requirements based on the equations 3.1 to 3.9. Other options enable admission officer to set number of candidates to admit on course basis, cut-off marks for each of the courses and view admission results outcomes either on course basis or all the courses together.

UME admission requirements' *add* option presents an html form that enables the admission officer to add or edit UME admission requirements for each of the available courses. Fig. 4.7 shows this html form. Status on the html form enables admission officer to set a particular UME admission subject requirement to be compulsory or set a group of subjects through which candidate must select from. R#2 or R#3 is used to set the number of subjects to be selected from a particular group of subjects. The *edit* part of UME admission requirements option enables admission officer to update or delete previous created entries from the UME admission requirements table.

O/L admission requirements option is similar to the UME admission requirements option except that it deals with O/L subjects requirements.

Databases sub-module presents an html form in Fig. 4.8 and it allows admission officer to extract the data of all the potential candidates that made his institution either first choice or second choice from the JAMB UME databases. The NECO and WASC options enable one to validate all the potential candidates O/L data against the bodies' databases. In the future, these examinations bodies could make candidates' examinations data available on either CD or DVD to institutions.

Admission process sub-module employs equations 3.1 to 3.9 to evaluate potential candidates' data with the university admission requirements and then determine who are to be admitted for the academic session in question.

Check Post UME score option presents an html form in Fig. 4.9 to the admission officer to set quota, cut-off marks for each course and batches during the admission exercises. JavaScript is employed in validating the entries in the form upon clicking the submit button. Admission results option enables the admission officer to display results on course-by-course basis or display all admission results together.

How to … module provides information on admission procedures, how to apply for a course, admission requirements and the like to the users.

On the home page, other interesting features are available to users. These include Admission (courses available, UME and O/L requirements for admission, Submit scores, self evaluation, results, news and ABU. Most of these features are similar to the ones discussed above.

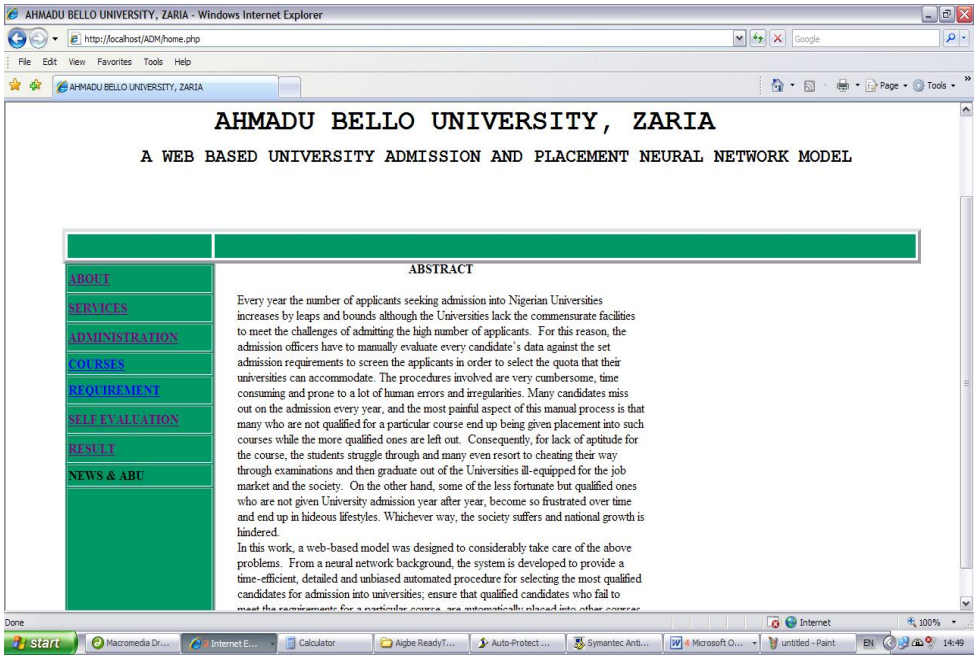

Fig 4.3 : University Admission and Placement Home Page

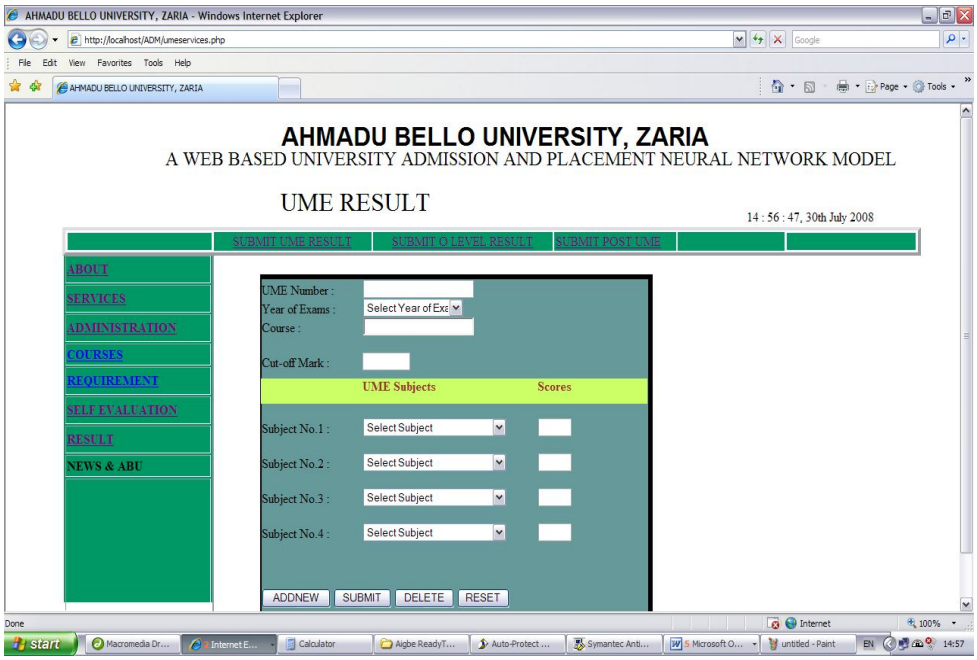

Fig 4.4: JAMB (UME) Results Form Page

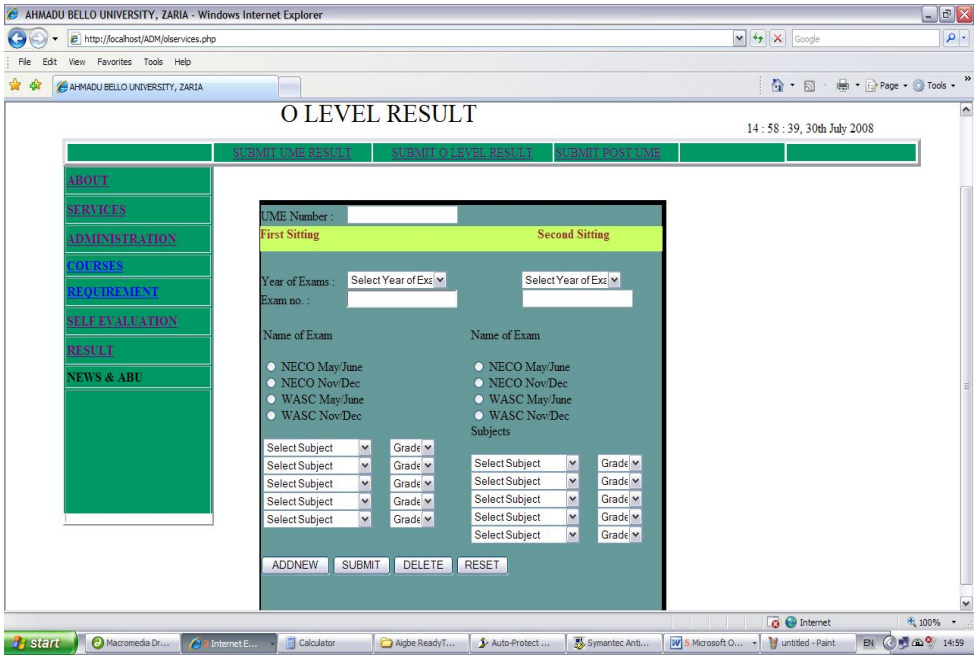

Fig. 4.5: O/Level Results Form Page

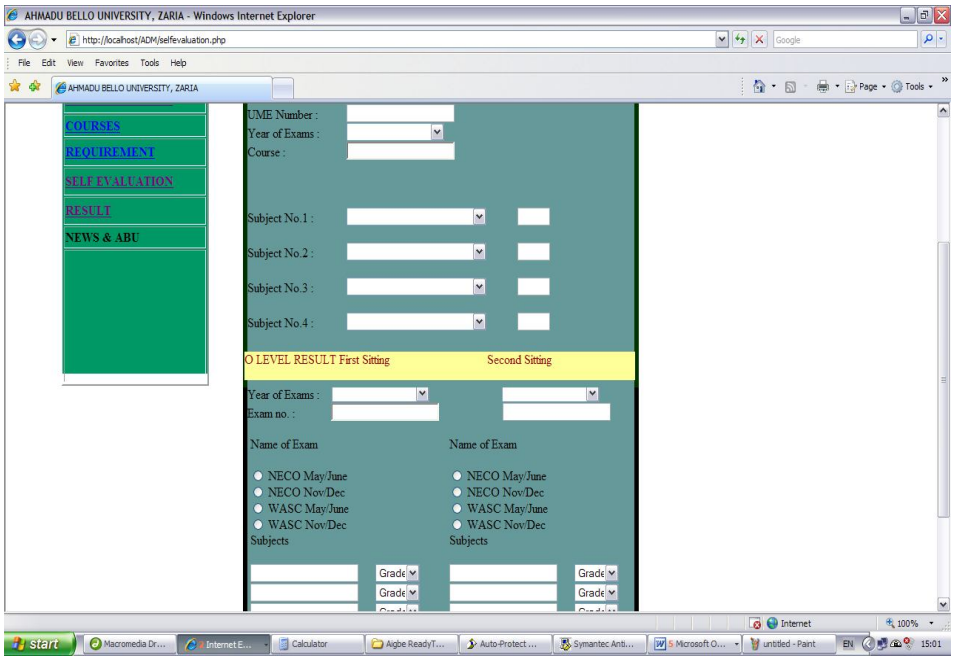

Fig. 4.6: Candidates' Self Evaluation Form Page

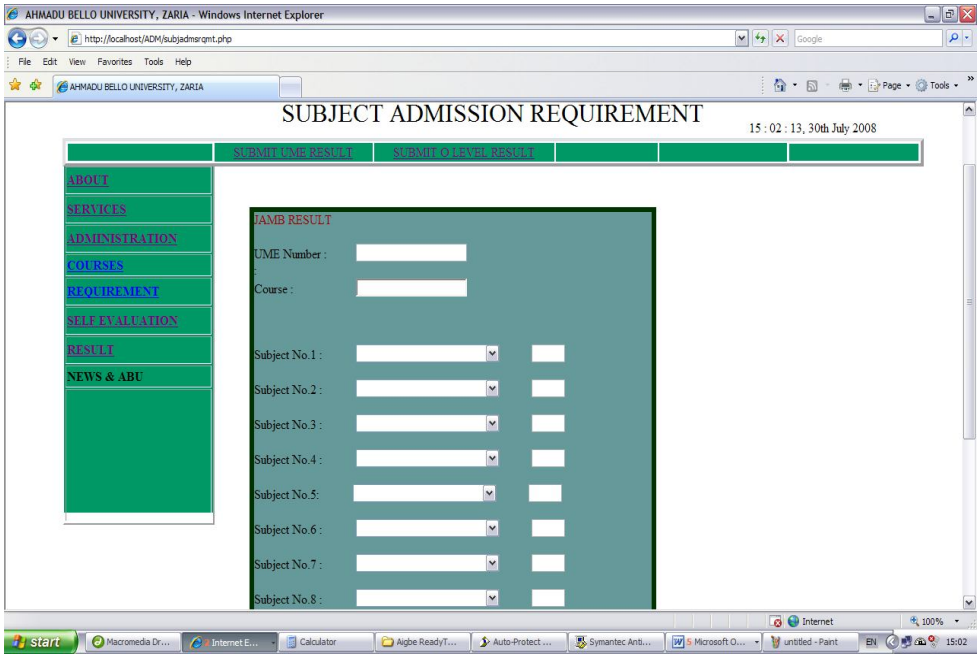

Fig. 4.7: UME Subjects Admission Requirements Form Page

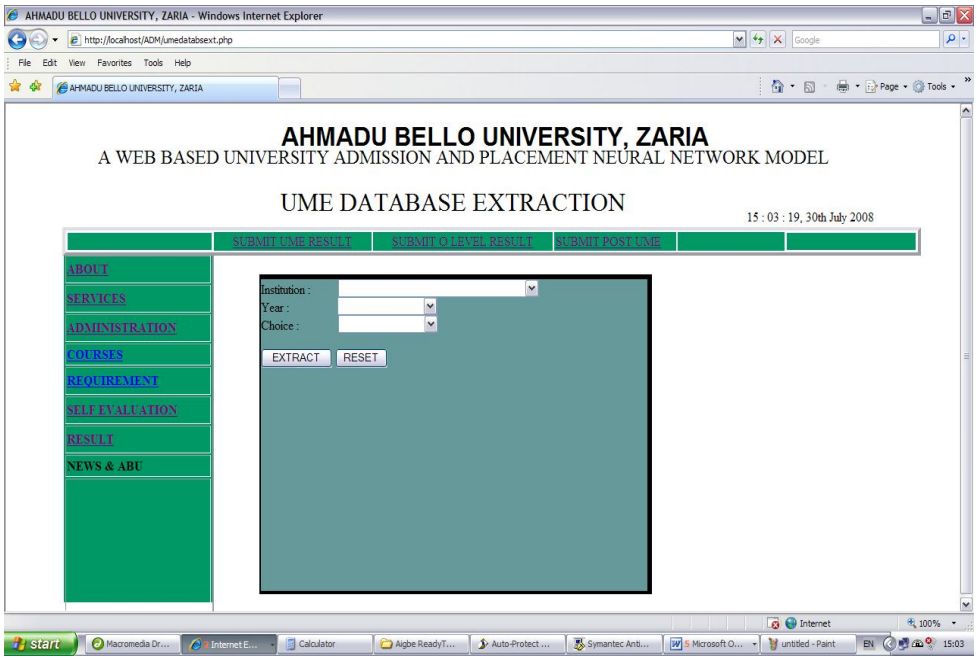

Fig. 4.8: UME Database Extraction Form Page

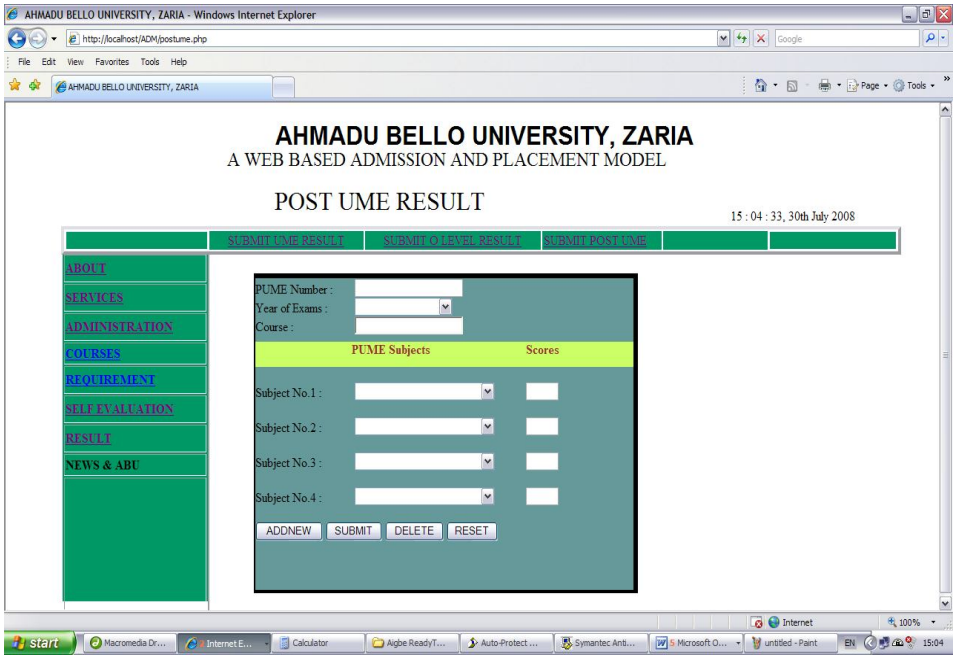

Fig. 4.9: University Post UME Score Information Form Page

4.3 System Implementation

The three-tier architecture is conceptual. Practically, different implementations of web database applications do fit this architecture. A common implementation for popular websites is to install the web server and the database server on separate machines to permit a more scalable and faster application. Very high-end applications require a cluster of computers, where the database and web servers are replicated and the load distributed across many machines. The most common implementation has the web server (including the scripting engine that processes the scripts and carries out the actions they specify) and the database management system installed on one machine. This implementation is secure and the simplest to manage.

The university admission and placement system will be a catalyst for change by creating the organisational and technical building blocks that allow the admission officers and admission seekers to conceive new ways of evaluating potential candidates' data against university admission requirements. The main objective of the system is to develop and implement an internet-based university admission and placement system.

The main goal being put forward while designing the system is its simple accessibility with minimum requirements on the users' side. Due to large flexibility of information delivery over internet, the system is implemented as a standard internet application. The client side requires no more than the standard internet browser installed on the local computer while the main application functionality is assured by the server side.

Tables 4.1 to 4.5 show some of the database tables required in order to evaluate potential candidates against the university admission requirements. Table 4.1 and table 4.2 show the course UME and O/L admission requirements respectively. Tables 4.3 to 4.5 show potential candidates' UME subjects attempted, O/L grades in either first or second sitting respectively. Admission process module applies equations 3.1 to 3.8 to evaluate Tables 4.3, 4.4 and 4.5 against Tables 4.1 and 4.2 to obtain Table 4.6.

4.4 System Requirements

The basic hardware requirement for the university admission and placement system is a simple local area network with a server and client machines. The

client side requires no more than the standard Internet browser installed on the local computer while the main application functionality is assured by the server side. However, a single computer system may also serve as a server and a client machine. It also requires a network link to be able to link JAMB or NECO or WASC network. The computer system may be a simple Pentium system with monitor and printer. The basic software requirement for the system is Apache, MySQL and PHP (AMP) presented in chapter 3 section 3.2, and Windows Operating System.

Table 4.1 – UME Admission Requirements Table

Course_code	u_{K1} Eng	U_{K2} Mth	U_{K3} phy	U_{K4} Bio	U_{K5} eco	U_{K6} agr	U_{K7} che	U_{K8} geo	U_{K9} sta	U_{K10} com	U_{K11} Crs	U_{K12} his	U_{K13} Acc	U_{K14} gov	U_{K15} Fat	U_{K16} frc	U_{K17} enl	U_{K18} irs	U_{K19} tdr	U_{K20} fma	U_{K21} hec
COSC	1	1	1	2	2	2	2	2													
MATH	1	1	2		3		2	3													
STAT	1	1	2		2		2	2	2											2	
BIOL	1	2	2	1			1														
BCHM	1	2	2	1			1														
PHYS	1	1	1	2			2														
CHEM	1	2	2	2			1														
GEOL	1	2	2	2			2	2													
VETM	1	1	1	1			1														
GEOG	1	2	2	2		2	2	1													
MICR	1	2	2	1			1														
TEXT	1	1	1	2	2	2	1	2	2		2							2			
NURS	1	1	1				1														
MBBS	1	1	1				1														
PHARM	1	1	1	1			1														
AGRIC	1	3	3	2	3	2	1	3													
BUAD	1	1			1			2	2	2			2	2							
PUAD	1				1					2	2	2	2	1			2	2			
ECON	1	1			1			2			2	2		2		2	2	2			
SOCI	1	4			3						3	2		2		3	3	3			
MENG	1	1	1	2		2	1														
EENG	1	1	1	2		2	1														
CENG	1	1	1	2	1	2	2														
WRES	1	1	1	1		1															
POLS	1	4			3						3	2		2		3	3	3			
MCOM	1	2			2					2	2	2		2		2	2	2		2	2
ENGL	1				2			2			2	2		2		2	1	2			
QSUR	1	1	1	2	2		1	2							2						
MTEN	1	1	1	2		2	1	2													

Table 4.2 – O/L Admission Requirements Table

	O_{K1}	O_{K2}	O_{K3}	O_{K4}	O_{K5}	O_{K6}	O_{K7}	O_{K8}	O_{K9}	O_{K10}	O_{K11}	O_{K12}	O_{K13}	O_{K14}	O_{K15}	O_{K16}	O_{K17}	O_{K18}	O_{K19}	O_{K20}	O_{K21}
COURSE _CODE	eng	mth	phy	Bio	eco	agr	che	geo	sta	com	crk	his	acc	gov	fat	frc	enl	irs	tdr	fma	hec
COSC	1	1	1	2		2	2	2													
MATH	1	1	2	3	3		2	3	3	3	3	3	3	3	3	3	3	3	3	3	3
STAT	1	1	2		2		2	2												2	
BIOL	1	1	2	1		2	1	2													
BCHM	1	1	1	1			1														
PHYS	1	1	1	2		2	1	2													
CHEM	1	1	1	1			1														
GEOL	1	1	2	1			1	2													
VETM	1	1	1	1			1														
GEOG	1	1	2	2		2	2	1													
MICR	1	1	1	1			1														
TEXT	1	1	1	2		2	1	2						2							
NURS	1	1	1	1			1														
MBBS	1	1	1	1			1														
PHARM	1	1	1	1			1														
AGRIC	1	3	3	2	3	2	1	3													
BUAD	1	1			1			2	2	2											
PUAD	1				1					2	2	2	2	1		2		2			
ECON	1	1			1			2		2	2	2		2			2	2			
SOCI	1	4	3	3	3	3	3	3	3	3	3	2	3	2	3	3	3	3	3	3	3
MENG	1	1	1	2		2	1														
EENG	1	1	1	2		2	1														
CENG	1	1	1	2		2	1														
WRES	1	1	1	2		2	1														
POLS	1	4	3	3	3	3	3	3	3	3	3	2	3	2	3	3	3	3	3	3	3
MCOM	1				2			2		2	2	2		2			2	2			
ENGL	1				2			2		2	2	2		2			1	2			
QSUR	1	1	1				1	2							2				2		
MTEN	1	1	1	2		2	1														

Table 4.3 – Candidate JAMB UME Subjects Table

		J_{I1}	J_{I2}	J_{I3}	J_{I4}	J_{I5}	J_{I6}	J_{I7}	J_{I8}	J_{I9}	J_{I10}	J_{I11}	J_{I12}	J_{I13}	J_{I14}	J_{I15}	J_{I16}	J_{I17}	J_{I18}	J_{I19}	J_{I20}	J_{I21}
STD_JNO	COURSE _CODE	eng	mth	phy	bio	eco	Agr	che	geo	sta	com	crs	his	acc	gov	fat	frc	enl	irs	tdr	phe	hec
	COSC	1	1	1			1															
52373803GE	MATH	1	1	1				1														
51111507HC	STAT	1	1			1		1														
52124887AJ	BCHM	1	1		1			1														
52873518FE	BCHM	1	1	1	1																	
52979113HB	BCHM	1	1		1			1														
52612994JH	BIOL	1		1	1			1														
RS1/04-05/0190	BIOL	1		1	1			1														
52979902DC	GEOG	1	1	1					1													
52364125JF	MICR	1	1		1			1														
RS1/04-05/0196	MICR	1		1	1			1														
52237331ED	GEOL	1	1	1				1														
RS3/04-05/0015	GEOL	1	1					1	1													
RS1/04-05/0485	GEOL	1	1		1			1														
52979600GF	PHYS	1	1	1				1														
52979333JA	PHYS	1		1	1			1														
53072574BI	PHYS	1	1			1									1							
53037901FD	TEXT	1	1	1				1														

Table 4.4 – Candidate O/L First Sit Table

		A_{I1}	A_{I2}	A_{I3}	A_{I4}	A_{I5}	A_{I6}	A_{I7}	A_{I8}	A_{I9}	A_{I10}	A_{I11}	A_{I12}	A_{I13}	A_{I14}	A_{I15}	A_{I16}	A_{I17}	A_{I18}	A_{I19}	A_{I20}	A_{I21}
STD_JNO	STUD_N0	eng	Mth	phy	bio	eco	agr	che	geo	sta	com	Crk	his	acc	gov	Fat	frc	Enl	irs	tdr	phe	Hec
52373803GE	4201006/027	3	2	3	3		3	3	4			3										
51111507HC	3147166FA	3			3	3	2	4	3			2										
52124887AJ	31135225IH	4	3	3	4	3	3	2	3													
52873518FE	42181119DA	3	3	4	3	4	3	3	4													
52979113HB	4191906/188	3	4	4	3	4	4	3	4			2										
52612994JH	4332115667	3	2	3	2	3		3	2						1							
RS1/04-05/0190	41546210GF	4	4	3	3	3		3	4										4			3
52979902DC	41626998AH	3	4	3	2	3	3	3	2			3										
52364125JF	4190911/286	3	3	2	2	3	3	2	1											2		
RS1/04-05/0196	4340901186	3	4	4	3	4	2	2	4										2			
RS3/04-05/0015	4020613/104	4		3	4	4	4		4													
RS1/04-05/0485	42O2410/110	3	4	3	3	4	3	2														
52979600GF	41626951FB	3	3	3	3	3		4	3											2		
52979333JA	41372424EI	3	3	3	3	4	3	3	4													
53072574BI	4191911033	4	3	3	3	4	3	3	2													
53037901FD	53176820JG	3	3	3	1		2	4				3										3

Key: A = 1
B = 2
C = 3
D – E = 4

Table 4.5 – Candidate O/L Second Sit Table

STD_JNO	STUD_N0	B_{I1}	B_{I2}	B_{I3}	B_{I4}	B_{I5}	B_{I6}	B_{I7}	B_{I8}	B_{I9}	B_{I10}	B_{I11}	B_{I12}	B_{I13}	B_{I14}	B_{I15}	B_{I16}	B_{I17}	B_{I18}	B_{I19}	B_{I20}	B_{I21}
		eng	mth	phy	bio	eco	agr	che	geo	sta	com	crk	his	acc	gov	fat	frc	enl	irs	tdr	fma	hec
52373803GE																						
51111507HC	2512043		3		3				4													
52124887AJ	5251627247	3	2	3	3	3	2	4	4													
52873518FE																						
52979113HB	61048112AG	3	3	3	4	4	3	4	3			3										
52612994JH																						
RS1/04-05/0190	RS1/04-05/0190	3	3					3														
52979902DC																						
52364125JF																						
RS1/04-05/0196																						
RS3/04-05/0015	4020613/104	3	4	4	3	4	4		3													
RS1/04-05/0485	RS1/04-05/0485	1	2	3	1																	
52979600GF																						
52979333JA																						
53072574BI	11737205GE	2	2	4	2	2	3	3	2													
53037901FD	4030708/057	4	2	2	2		3	2				2										

Key: A = 1
B = 2
C = 3
D – E = 4

Table 4.6 - Admission Results Table

STD_JNO	course_code	Jscore	Scut_off	Pume	Pufac	Admi	ADMREM	Plcmt	PLMREM	final course
52373803GE	MATH	245	160	167.5	40	1	admt			math
51111507HC	STAT	80	160		40	0	unadmt		nopume	
52124887AJ	BCHM	206	160	211	40	1	admt			bchm
52873518FE	BCHM	178	160	186	40	0	unadmt		wsjcmb	
52979113HB	BCHM	189	160	165	40	1	admt			bchm
52612994JH	BIOL	233	160	163	40	1	admt			biol
RS1/04-05/0190	BIOL	1.6	160		40	0	unadmt		unqlfd	
52979902DC	GEOG	246	160	196	40	1	admt			geog
52364125JF	MICR	219	160	193	40	1	admt			biol
RS1/04-05/0196	MICR	2.8	160		40	0	unadmt		unqlfd	
52237331ED	GEOL	229	160	125	40	1	admt			geol
RS3/04-05/0015	GEOL	2.4	160		40	0	unadmt		unqlfd	
RS1/04-05/0485	GEOL	3.8	160		40	0	unadmt		unqlfd	
52979600GF	PHYS	194	160	159	40	1	admt			geog
52979333JA	PHYS	189	160	192	40	1	admt			phys
53072574BI	PHYS	239	160	200	40		unadmt		wsjcmb	
53037901FD	TEXT	175	160	199.5	40	1	admt			text

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Technological advancements particularly in the area of information and communication technonology keep growing by leaps and bounds and on a daily basis. Keeping abreast and taking advantage of these technological advancements is a paramount concern to users, for solving encountered societal problems.

Apart from the manual system of university admission and placement being cumbersome and time-consuming, it is rife with a lot of irregularities which is understandably due to our human weaknesses and ineptitude. Successful implementation of this research study will not only ensure that the most qualified candidates are offered admission into Nigerian universities, it will also take care of the irregularities and provide opportunities for candidates to be automatically placed into courses (other than their chosen ones), for which they are most suited. And of course, it will give credence and transparency to the admission process by providing avenue for candidate's self-screening.

For optimum realisation of the objectives of this research, however, the following general challenges facing information and communication technology in Nigeria must be overcome.

- Internet security issues: The Internet must be protected from both internal and external attacks in order to prevent intrusion over the networks.
- Power supply problems: The current unreliable nature of electricity supply in Nigeria calls for an alternative provision of power supply to ensure uninterrupted service. Solar energy can be used as an alternative.
- Computer literacy: The level of computer literacy particularly at the secondary school level has to be stepped up if adequate advantage of the student's self-screening admission system is to be maximised.

5.2 **Recommendations**

To take advantage of the benefits provided by this study nationwide, it is recommended that the Nigeria University Commission (NUC) should make it mandatory for universities across the country to start implementing this system for their admission and placement/ screening exercise. The admission officers should be adequately trained by way of workshops and seminars to be able to use the new system when implemented. Necessary funds should be made available to the admission offices for a smooth take-off of the system in each of the universities. Lastly, a proper maintenance culture should be adopted and funds committed to maintenance. This will ensure that the lifespan of the new system is elongated.

Further research could be carried out to accommodate other admission requirements not considered in this study. One such requirement is that of direct entry whereby, a considerable number of candidates in addition to the 'O'level certificate, have higher school certificate such as the National Certificate of Education (NCE). By their qualification, such candidates do not need to write the UME. They are usually admitted into the second year of the prospective courses of their choice.

REFERENCES

Adebiyi, A.B. (2006). A Web Based Model for Joint Admissions and Matriculation Board (JAMB) Students Admission Placement. Unpublished M.Tech. Thesis. Federal University of Technology, Akure, Nigeria.

Adewale, O. S. (2006). *University Digital Library: An Initiative to Improve Research, Teaching and Service.* Adeyemo Publishing House, Akure, Nigeria, pp.134-142.

Anderson, D. and McNeil, G. (1992). Artificial Neural Networks Technology, Data & Analysis Center for Software. Rome, NY 13441-4909. dacs@dtic.mil

Anderson, James A. (1986). *Cognitive Capabilities of a Parallel System*, Springer-Verlag.

Ashby D. and Kumar N. (1996). A Comparison of Neural Networks and Classical Discriminant Analysis in Anticipating Default Among High-Yield Bonds Association for Information Systems, *Americas Conference [On-line]*, 1-4. http://hsb.baylor.edu/ramsower/ais.ac.96/papers/ashby.htm

Barinaga, M. (1990). Neuroscience Models the Brain, *Science 247*, pp 524-527

Bijayananda, N. and Srinivasan, R. (2004). Using Neural Networks to Predict MBA Student Success, *College Student Journal*, March pp 1-7

Brown, Robert J. (1987). An Artificial Neural Network Experiment, *Dr. Dobbs Journal.*

Carbone, V. and Piras, G. (1998). Palomar project: Predicting School Renouncing Dropouts, Using the Artificial Neural Networks as a Support for Educational Policy Decisions. *Substance Use and Misuse* **3**, 717-750

Carpenter, Gail A. and Grossberg, Stephen (1988). The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network, *Computer*.

Caudill, M. and Butler, C. (1990). *Naturally Intelligent Systems*. Cambridge, U.S.A: The MIT Press.

Chris, O' Brain, C. and Kevin, L. (2004). Predicting GPA with a Neural Network. A Paper Presented at the University of Minnesota, Morris Computer Science on Neural Networking Spring.

Cowan, J. D. and Sharp, D. H. (1988). Neural Nets and Artificial Intelligence. *Daedalus*, 117(1), pp. 85-121.

Crask, M.R. and Perreault W.D. Jr. (1977). Validation of Discriminant Analysis in Marketing Research. *Journal of Market Research* 14 (February), pp. 60-68.

Cripps, A. (1996). Predicting Grade Point Average Using Clustering and ANNs. Proceedings of World Congress on Neural Networks, 486-490.

Dayhoff, J. E. (1990). *Neural Network Architectures: An Introduction*. Van Nostrand Reinhold, New York, N.Y.

Dillon, W. R. and Goldstein M. (1978). On the Peformance of Some Multinomial Classification Rules. *Journal of the American Statistical Association* 73 (June), pp. 305-313.

Dillon, W. R. and Schiffman, L. (1978). Appropriateness of Linear Discriminant and Multinomial Classification Analysis in Marketing Research. *Journal of Marketing Research* 15 (Febrauary), pp. 103-112.

Dillon, W.R. (1979). The Performance of the Linear Discrimant Function in Non-Optimal Situations and the Estimation of Classification Error Rates: A Review of Recent Findings. 7' *Journal of Marketing Research*, 16 (August), pp. 370-391.

Dillon, W.R. and Goldstein M. (1984). *Multivariate Analysis Methods and Applications.* John Wiley and Sons, New York.

Dobson, P.; Krapljan-Barr, P. and Vielba, C. (1999). An evaluation of the validity and fairness of the Graduate Management Admissions Test (GMAT) used for MBA selection in a UK business school. *International Journal of Selection and Assessment*, **7**, 196-202.

Duda, R.O.; Hart, P.E.; Stork, D.G. (2001). *Pattern Classification (2nd edition)*, Wiley.

Eberhart, R.C. and R.W. Dobbins (1990). *Neural Network PC Tools: A Practical Guide.* Academic Press, San Diego, CA.

Fischler and Firschein (April 1987). *Intelligence: The Eye, the Brain, and the Computer*. Reading, MA, Addison-Wesley, p. 23.

Fisher, L. and Van Ness J.W. (1973). Admissible Discriminant Analysis. *Journal of the American Statistical Association* 68, pp. 603-607.

Frank, R.E., Brassy, W.F. and Morisson, D.G. (1965). Bias in Multiple Discriminant Analysis. *Journal of Marketing Research* 2 (August), pp. 250-258.

Goldstein, M. and Rabinowitz, T. (1975). Selection of Variates for the Two Group Classification Problem. *Journal of the American Statistical Association* 70, pp. 776-781

Gorr, W.L.; Nagin, D. and Szczypula, (1994). Comparative Study of Artificial Neural Network and Statistical Models for Predicting Student Grade Point Averages. *International Journal of forecasting*, **10**(2), 17-34.

Graham, L.D. (1991). Predicting Academic Success of Students in a Master of Business Administration Program. *Educational Psychology Measure*, **4**, 721-727.

Grossberg, Stephen, and Rudd, M.E. (1989). A Neural Network for Visual Motion Perception: Group and Element Apparent Motion. *Neural Networks*, Volume 2.

Gurney, K. (1997). *An Introduction to Neural Networks.* London: Routledge.

Hardgrave, B.C, Wilson, R.L. and Kent, K.A. (1994). Predicting Graduate Student Success: A comparison of Neural Networks and Traditional Techniques. *Computer and Operations Research*, **21**, 249-263.

Hartnett, T. (2000). Financing Trends and Expenditure Patterns in Nigerian Federal Universities: An Update. Unpublished Report. Washington, D.C.: Africa Region; The World Bank.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*, Prentice Hall,

Hebb, D. O. (1949). *The Organization of Behavior.* John Wiley, New York.

Hecht-Nielsen, Robert (1990). *Neurocomputing*. Addison-Wesley, ISBN 0-201-09255-3

Hertz, J., Palmer, R.G., Krogh. A.S. (1990). *Introduction to the theory of neural computation*. Perseus Books.

Hills, M. (1966). Allocation Rules and their Error Rates. *Journal of the Royal Statistical Society* B28, pg. 1.

Johnson, R.A. and Wichern, D.W. (1998). *Applied Multivariate Analysis,* 4th ed., Prentice Hall, Englewood Cliffs, New Jersey.

Khanna, T. (1990). *Foundations of Neural Networks.* Addison-Wesley Publishing Co. New York.

Krishnaswami, P. and Nath, R. (1968). Bias in Multinomial Classification, *Journal of the American Statistical Association* 63, pp. 298-303

Krzanowski, W.J. (1975). Discrimination and Classification Using Both Binary and Continuous Variables. *Journal of the American Statistical Association* 70, pp. 782-790

Kulkarni, A.D. (1993). *Artificial Neural Networks for Image Understanding*. Van Nostrand Reinhold, New York.

Lachenbruch, P.A. (1967). An Almost Unbiased Method of Obtaining Confidence Intervals for the Probability of Misclassification in Discriminant Analysis. *Biometrics* 23, pp. 639-645

Lachenbruch, P.A. and Mickey, M.R. (1968). Estimation of Error Rates in Discriminat Analysis. *Technometrics* 10, pp.1-11

Leyva López, J.C. (2000). A Genetic Algorithm Application for the Individual and Group Multicriteria Decision Making: PhD Thesis Resume. *Computación y Sistemas*, 4, 183-188.

Lievens, F. and Coetsier, P. (2002). Situational Tests in Student Selection: An Examination of Predictive Validity, Adverse Impact, and Construct Validity. *International Journal of Selection and Assessment*, 10, pp. 245-257.

Lindsey, C. S. and Lindblad, T. (1995). Survey of Neural Network Hardware. Proceedings of SPIE95.

Lindsey, C. S.; Denby, B. and Lindblad, T. (1996). *Neural Network Hardware*, [Online], Artificial Neural Networks in High Energy Physics, Available: http://www1.cern.ch/NeuralNets/nnwInHepHard.html, [1996, August 30].

Lippman, K. (1987). Neural Networks at Pacific Northwest National Laboratory. http://www.emsl.pnl.gov:2080/docs/cie/neural/neural.homepage.html

Looney, C. G. (1996). Advances in FeedForward Neural Networks: Demystifying Knowledge Acquiring Black Boxes, IEEE Educational Activities Department, Piscataway, NJ, USA.

Looney, C. G. (1997). *Pattern Recognition Using Neural Networks: Theory and Algorithms for Engineers and Scientists*. Oxford University Press, Inc. New York, USA.

Maduewesi, B.U. and Imhanlahimi, E.O. (2006). Implementing Panacea for Admission Crisis into Nigerian Universities: An Innovation Diffusion Plan. *College Student Journal.*
http://goliath.ecnext.com/coms2/summary_0199-5776175_ITM

Masters, T. (1994). *Signal and Image Processing with Neural Networks*. John Wiley & Sons, Inc.

Masters, T. (1993). *Practical Neural Network Recipes in C++*. Academic Press Inc., San Diego, CA.

McCartor, H. (1991). Back Propagation Implementation on the Adaptive Solutions CNAPS Neurocomputer. *Advances in Neural Information Processing Systems* 3, USA.

McCulloch, W. S. and Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity:, *Bulletin of Mathematical Biophysics*, pp. 5:115-33.

McCullock, P. A. and Pitts, K. (1949). Artificial Neural Network for Speech and Vision. *Neural Computing Series*, Mammone Ed, Chapman-Hall.

McLachlan, G.J. (1974). Estimation of the Errors of Misclassification on the Criterion of Asymptotic Mean Square Error. *Technometrics* 16 (May), pp.255-256.

McLachlan, G.J. (177). Estimation of the Linear Discriminat Function from Initial Samples Containing a Small Number of Unclassified Observations. *Journal of the American Statistical Association* 72, pp. 403-406.

Minsky K. and Papert W. (1969). *Neural Network and Applications.* Coastal Carolina University Press, USA

Minsky, M. L. and Papert, S. A. (1969). *Perceptrons*, MIT Press, Cambridge, MA, USA.

Minsky, Marvin L. and Papert, Seymour A. (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, USA.

Minsky, M. and Papert, S. A. (1988). *Perceptrons. Expanded Edition*, MIT Press, Cambridge, MA.

Morris, K. M and Gibsson, R. (2004). The Predictive Validity of Selected Admission Variables Relative to GPA in a Master of Business Administration Program. Education and Psychological Measurement, **42**, 1163-67

Nelson, C.V. and Henricksen, L.W. (1994). Using Neural Net Technology to Enhance the Efficiency of a Computer Adaptive Testing Application. Mid-Western Educational Research Association Annual Meeting, 1 – 40

Nelson, M. M. and Illingworth, W. T. (1991). *A Practical Guide to Neural Nets*. Addison-Wesley Publishing Company, Inc, Reading, MA.

Nilsson, N. J. (1990). *The Mathematical Foundations of Learning Machines*. Morgan Kaufmann Publishers.

Omoike, D. and Aluede, O. (2007). Appraisal of Production Function of Nigerian Universities: Admission Disparity in a Deregulated Context. *Anthropologist*, 9(2): 133-137.

Omolewa, M. (1981). Historical Antecedent of Correspondence Education in Nigeria. *West Africa Journal of Education.*

Pao, Yoh-Han, (1989). *Pattern Recognition and Neural Networks.* Addison-Wesley.

Pedhauzer, E. (1982). *Multiple Regression in Behavioral Research.* Fort Worth, TX:Harcourt Brace Jovanovich.

Powers, D.E. and Lehman, J. (1983). GRE Candidates' Perceptions of the Importance of Graduate Admission Factors. *Research in Higher Education*, **19**, 231-249.

Randles, R.H. Brofitt, J.D., Ramberg, J.S. and Hogg, R.V. (1978). Discriminant Analysis Based on Ranks. *Journal of the American Statistical Association* 73, pp. 376-384.

Renner, R.S. and Lacher, R.C. (2000). Combining Constructive Neural Networks for Ensemble Classification. *Proceedings of the Fifth Joint Conference on Information Science*, 887-891

Ripley, B. D. (1993). Statistical Aspects of Neural Networks, Networks and Chaos: Statistical and Probabilistic Aspects edited by Barndoff-Nielsen, O. E., Jensen, J.L. and Kendall, W.S., Chapman and Hall, London, United Kingdom.

Roger, M., Bruen, M. and Maystre, L. (2000). *Electre and Decision Support.* Kluwer, Academic Publishers.

Rosenblatt, F. (1958). The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, Vol. 65:pp.386-408.

Roy, B. (1996). *Multicriteria Methodology for Decision Aiding*. Kluwer.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning Internal Representations by Error Propagation. Institute for Cognitive Science Report 8506, San Diego, University of California.

Rumelhart, D.E., McClelland, J.L., editors, (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol.1,* pp. 216-271, MIT Press, Cambridge Mass., USA.

Sejnowski, T. J. and Rosenburg, C. R. (1987). Parallel Networks that Learn to Pronounce English Text. *Complex Systems*, No. 1, pp. 145-168.

Sheel, S.J, Renner R.S. and Dawsey, S. K. (2002). Alternatives to Math Placement Exams: A Look at Discriminant Analysis, Neural Networks and Ensembles of Networks, Department of Computer Science, Coastal Carolina University, Chico Conway, SC USA.

Shepherd, G.M. (1990). *The Synaptic Organization of the Brain*, 3rd Ed. Oxford University, Press.

Shih, Y. (1994). *Neuralyst* ä *User's Guide*, Cheshire Engineering Corporation, USA, p. 21.

Smith, M. (1993). *Neural Networks for Statistical Modeling*, Van Nostrand Reinhold

Simpson, P. K. (1990). *Artificial Neural Systems*, Pergamon Pr., USA.

Urbakh, V.U. (1971). Linear Discriminant Analysis: Loss of Discriminating Power When a Variate is Omitted. *Biometrics* 27, pp. 531-534.

Vanderpooten, D. (1990). The Construction of Prescriptions in Outranking Methods. **In**: *Reading in Multiple Criteria Decision Aid* [edited by C.A. Bana e Costa], Springer Verlag, Berlin, 184-215.

Wassermann, Philip D. (1989). *Neural Computing, Theory and Practice.* Van Nostrand, NY.

Wasserman Y. G. (1990). Neural Networks as Systems for Pattern Recognition: *A Review*, Pattern Recognition and Image Analysis, **5** (2): 254 – 293.

Wasserman, P. (1993). *Advanced Methods in Neural Computing*. Van Nostrand Reinhold.

Werbos, P. (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Ph.D. Thesis, Harvard University.

Widrow, B. and Hoff, M. D. (1960). Adaptive Switching Circuits. *IRE WESCON Convention Record*, Part 4, pp. 96-104.

Wikipedia, (1996), Online Encyclopaedia, http://en.wikipedia.org

Wilson, T. (1999). A Student Selection Method and Predictors of Success in a Graduate Nursing Program. *Journal of Nursing Education*, **38**, 183-187.

Wolming, S. (1999). Validity Issues in Higher Education Selection: A Swedish Example. *Studies in Educational Evaluation*, **25**, 335-351.

Wright, R.E. and Palmer, J.C. (1994). GMAT Scores and Undergraduate GPAs as Predictors of Performance in Graduate Business Programs. *Journal of Education for Business*, **70**, 344-348

Zador, A.M., Mainen, Z.F, Claiborne, B.J., Brown, T.H. (1990). *Self-Organization of Hebbian Synapses in Hippocampal Neurons*. Morgan Kaufmann Publishers Inc.

APPENDIX

SOURCE CODES

//The Home Page//

```
<?php
 session_start();
if(isset($_SESSION['valid_user']))
{
        //header("Location: home.php");
}
else{
header("Location: Notlog.php");
}?>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>AHMADU BELLO UNIVERSITY, ZARIA</title>
<style type="text/css">
<!--
#Layer1 {
        position:absolute;
        width:199px;
        height:430px;
        z-index:1;
        left: 77px;
        top: 329px;
        background-color: #009966;
}
#Layer2 {
        position:absolute;
        width:456px;
        height:84px;
        z-index:2;
        left: 749px;
        top: 163px;
        border-color: #FFFFFF;
        border-top-style: solid;
        border-right-style: solid;
        border-bottom-style: solid;
        border-left-style: solid;
        border-top-width: thick;
        border-right-width: thick;
        border-bottom-width: thick;
        border-left-width: thick;
```

```
        background-color: #FFFFFF;
}
body {
        margin-left: 5%;
        margin-top: 10%;
        background-color: #FFFFFF;
}
.aerocss {
        padding-right: 40px;
        padding-left: 0px;
        font-size: 14px;
        font-weight: bold;
}
#Layer3 {
        position:absolute;
        width:431px;
        height:383px;
        z-index:3;
        left: 302px;
        top: 210px;
        background-color: #FFFFFF;
        border-right-color: #FFFFFF;
        border-width: 10;
        outline: 10;
        border-top-style: solid;
        border-right-style: solid;
        border-bottom-style: solid;
        border-left-style: solid;
        border-top-width: thick;
        border-right-width: thick;
        border-bottom-width: medium;
        border-left-width: thin;
        border-top-color: #FFFFFF;
        border-bottom-color: #FFFFFF;
        border-left-color: #FFFFFF;
        outline-width: thick;
}
.style2 {font-family: Arial, Helvetica, sans-serif}
#Layer4 {
        position:absolute;
        width:542px;
        height:43px;
        z-index:4;
        left: 909px;
        top: 161px;
}
#Layer5 {
        position:absolute;
        width:327px;
        height:318px;
```

```
        z-index:5;
        left: 883px;
        top: 283px;
        visibility: visible;
        overflow: auto;
        border-right-style: solid;
        border-bottom-style: solid;
        border-left-style: solid;
        border-top-width: thick;
        border-right-width: thick;
        border-bottom-width: thick;
        border-left-width: thick;
        border-color: #FFFFFF;
        border-width: 20;
        border-top-color: #FFFFFF;
        border-top-style: solid;
        border-top: 10;
        border-right: 10;
        border-bottom: 10;
        border-left: 10;
        outline-color: #FFFFFF;
        outline-style: groove;
        outline-width: thick;
        outline: 20;
        background-color: #FF3366;
}
#Layer6 {
        position:absolute;
        width:1143px;
        height:186px;
        z-index:6;
        left: 75px;
        top: 47px;
}
.anytg18 {
        margin-left: 3.28cm;
}
.anytg15 {
        margin-left: 2.13cm;
}
.ANYTG {
        margin-left: 1.66cm;
}
.anyTSC {
        margin-left: 3.52cm;
}
.any14 {
        margin-left: 4.36cm;
        background-color: #FFFFFF;
        font-size: 14px;
```

```
        color: #FF6600;
}
.any36 {
        margin-left: 4.8cm;
}
.any42 {
        margin-left: 3.72cm;
        background-color: #FFFFFF;
        font-size: 14px;
        color: #FF6600;
}
.any366 {
        margin-left: 3.65cm;
        background-color: #FFFFFF;
        font-size: 14px;
        color: #FF6600;
}
.anytg32 {
        margin-left: 2.25cm;
}
.anytg38 {
        margin-left: 4.55cm;
}
.anytg41 {
        margin-left: 3.89cm;
}
.anytg27 {
        margin-left: 2.34cm;
}
.anyPRD {
        margin-left: 2cm;
}
.anytg34 {
        margin-left: 3cm;
}
.anyDRC {
        margin-left: 1.62cm;
}
.anytg4 {
        margin-left: 3.98cm;
}
.anytg388 {
        margin-left: 3.83cm;
}
#Layer7 {
        position:absolute;
        width:962px;
        height:65px;
        z-index:7;
        left: 742px;
```

```
        top: 139px;
}
.style7 {font-size: 36px; font-weight: bold; outline-color: #9999FF; font-family:
"Courier New", Courier, monospace; }
.style8 {font-size: 24px}
#Layer8 {
        position:absolute;
        width:388px;
        height:285px;
        z-index:8;
        left: 995px;
        top: 377px;
}
.imga380 {
        background-color: #FFFFFF;
        background-image: none;
        background-repeat: repeat;
}
#Layer9 {
        position:absolute;
        width:556px;
        height:111px;
        z-index:8;
        left: 160px;
        top: 132px;
}
#Layer10 {
        position:absolute;
        width:896px;
        height:63px;
        z-index:8;
        left: 201px;
        top: 89px;
}
#Layer11 {
        position:absolute;
        width:537px;
        height:432px;
        z-index:8;
        left: 306px;
        top: 328px;
}
-->
</style>
<script type="text/JavaScript">
<!--
function MM_timelinePlay(tmLnName, myID) { //v1.2
  //Copyright 1998, 1999, 2000, 2001, 2002, 2003, 2004 Macromedia, Inc. All rights
reserved.
```

```
  var
i,j,tmLn,props,keyFrm,sprite,numKeyFr,firstKeyFr,propNum,theObj,firstTime=false;
  if (document.MM_Time == null) MM_initTimelines(); //if *very* 1st time
  tmLn = document.MM_Time[tmLnName];
  if (myID == null) { myID = ++tmLn.ID; firstTime=true;}//if new call, incr ID
  if (myID == tmLn.ID) { //if Im newest
    setTimeout('MM_timelinePlay('"+tmLnName+'",'+myID+')',tmLn.delay);
    fNew = ++tmLn.curFrame;
    for (i=0; i<tmLn.length; i++) {
      sprite = tmLn[i];
      if (sprite.charAt(0) == 's') {
        if (sprite.obj) {
          numKeyFr = sprite.keyFrames.length; firstKeyFr = sprite.keyFrames[0];
          if (fNew >= firstKeyFr && fNew <= sprite.keyFrames[numKeyFr-1]) {//in range
            keyFrm=1;
            for (j=0; j<sprite.values.length; j++) {
              props = sprite.values[j];
              if (numKeyFr != props.length) {
                if (props.prop2 == null) sprite.obj[props.prop] = props[fNew-firstKeyFr];
                else        sprite.obj[props.prop2][props.prop] = props[fNew-firstKeyFr];
              } else {
                while (keyFrm<numKeyFr && fNew>=sprite.keyFrames[keyFrm])
keyFrm++;
                if (firstTime || fNew==sprite.keyFrames[keyFrm-1]) {
                  if (props.prop2 == null) sprite.obj[props.prop] = props[keyFrm-1];
                  else        sprite.obj[props.prop2][props.prop] = props[keyFrm-1];
        } } } } }
      } else if (sprite.charAt(0)=='b' && fNew == sprite.frame) eval(sprite.value);
      if (fNew > tmLn.lastFrame) tmLn.ID = 0;
  } }
}

function MM_timelineGoto(tmLnName, fNew, numGotos) { //v2.0
  //Copyright 1998, 1999, 2000, 2001, 2002, 2003, 2004 Macromedia, Inc. All rights
reserved.
  var i,j,tmLn,props,keyFrm,sprite,numKeyFr,firstKeyFr,lastKeyFr,propNum,theObj;
  if (document.MM_Time == null) MM_initTimelines(); //if *very* 1st time
  tmLn = document.MM_Time[tmLnName];
  if (numGotos != null)
    if (tmLn.gotoCount == null) tmLn.gotoCount = 1;
    else if (tmLn.gotoCount++ >= numGotos) {tmLn.gotoCount=0; return}
  jmpFwd = (fNew > tmLn.curFrame);
  for (i = 0; i < tmLn.length; i++) {
    sprite = (jmpFwd)? tmLn[i] : tmLn[(tmLn.length-1)-i]; //count bkwds if jumping
back
    if (sprite.charAt(0) == "s") {
      numKeyFr = sprite.keyFrames.length;
      firstKeyFr = sprite.keyFrames[0];
      lastKeyFr = sprite.keyFrames[numKeyFr - 1];
```

```
    if ((jmpFwd && fNew<firstKeyFr) || (!jmpFwd && lastKeyFr<fNew)) continue;
//skip if untouchd
    for (keyFrm=1; keyFrm<numKeyFr && fNew>=sprite.keyFrames[keyFrm];
keyFrm++);
    for (j=0; j<sprite.values.length; j++) {
      props = sprite.values[j];
      if (numKeyFr == props.length) propNum = keyFrm-1 //keyframes only
      else propNum = Math.min(Math.max(0,fNew-firstKeyFr),props.length-1); //or
keep in legal range
      if (sprite.obj != null) {
        if (props.prop2 == null) sprite.obj[props.prop] = props[propNum];
        else       sprite.obj[props.prop2][props.prop] = props[propNum];
    } }
   } else if (sprite.charAt(0)=='b' && fNew == sprite.frame) eval(sprite.value);
  }
  tmLn.curFrame = fNew;
  if (tmLn.ID == 0) eval('MM_timelinePlay(tmLnName)');
}

function MM_initTimelines() { //v4.0
   //MM_initTimelines() Copyright 1997 Macromedia, Inc. All rights reserved.
   var ns = navigator.appName == "Netscape";
   var ns4 = (ns && parseInt(navigator.appVersion) == 4);
   var ns5 = (ns && parseInt(navigator.appVersion) > 4);
   var macIE5 = (navigator.platform ? (navigator.platform == "MacPPC") : false) &&
(navigator.appName == "Microsoft Internet Explorer") &&
(parseInt(navigator.appVersion) >= 4);
   document.MM_Time = new Array(1);
   document.MM_Time[0] = new Array(2);
   document.MM_Time["Timeline1"] = document.MM_Time[0];
   document.MM_Time[0].MM_Name = "Timeline1";
   document.MM_Time[0].fps = 2;
   document.MM_Time[0][0] = new String("behavior");
   document.MM_Time[0][0].frame = 12;
   document.MM_Time[0][0].value = "MM_timelineGoto('Timeline1','1')";
   document.MM_Time[0][1] = new String("sprite");
   document.MM_Time[0][1].slot = 1;
   if (ns4)
      document.MM_Time[0][1].obj = document["Layer7"];
   else if (ns5)
      document.MM_Time[0][1].obj = document.getElementById("Layer7");
   else
      document.MM_Time[0][1].obj = document.all ? document.all["Layer7"] : null;
   document.MM_Time[0][1].keyFrames = new Array(1, 14);
   document.MM_Time[0][1].values = new Array(3);
   if (ns5 || macIE5)
      document.MM_Time[0][1].values[0] = new Array("120px", "174px", "228px",
"282px", "335px", "389px", "443px", "497px", "551px", "605px", "658px", "712px",
"766px", "820px");
   else
```

```
        document.MM_Time[0][1].values[0] = new
Array(120,174,228,282,335,389,443,497,551,605,658,712,766,820);
    document.MM_Time[0][1].values[0].prop = "left";
    if (ns5 || macIE5)
        document.MM_Time[0][1].values[1] = new Array("193px", "186px", "179px",
"172px", "165px", "158px", "151px", "144px", "137px", "130px", "123px", "116px",
"109px", "102px");
    else
        document.MM_Time[0][1].values[1] = new
Array(193,186,179,172,165,158,151,144,137,130,123,116,109,102);
    document.MM_Time[0][1].values[1].prop = "top";
    if (!ns4) {
        document.MM_Time[0][1].values[0].prop2 = "style";
        document.MM_Time[0][1].values[1].prop2 = "style";
    }
    if (ns5 || macIE5)
        document.MM_Time[0][1].values[2] = new Array("962px", "949px", "937px",
"925px", "913px", "901px", "889px", "876px", "864px", "852px", "840px", "828px",
"816px", "804px");
    else
        document.MM_Time[0][1].values[2] = new
Array(962,949,937,925,913,901,889,876,864,852,840,828,816,804);
    document.MM_Time[0][1].values[2].prop = "width";
    if (!ns4)
        document.MM_Time[0][1].values[2].prop2 = "style";
    document.MM_Time[0].lastFrame = 14;
    for (i=0; i<document.MM_Time.length; i++) {
        document.MM_Time[i].ID = null;
        document.MM_Time[i].curFrame = 0;
        document.MM_Time[i].delay = 1000/document.MM_Time[i].fps;
    }
}
//-->
</script>
</head>

<body onLoad="MM_timelinePlay('Timeline1')">
<div id="Layer1">
  <table width="200" height="427" border="2">
    <tr>
      <th height="30" scope="col"><div align="left"><a
href="home.php">ABOUT</a></div></th>
    </tr>
    <tr>
      <th height="30" scope="row"><div align="left"><a
href="umeservices.php">SERVICES</a></div></th>
    </tr>
    <tr>
      <th height="30" scope="row"><div align="left"><a
href="ADMumeservices.php">ADMINISTRATION</a></div></th>
```

```
    </tr>
    <tr>
      <th height="23" scope="row"><div align="left"><a
href="accebdeparture.php">COURSES </a></div></th>
    </tr>
    <tr>
      <th height="30" scope="row"><div align="left"><a
href="separation.php">REQUIREMENT </a></div></th>
    </tr>
    <tr>
      <th height="30" scope="row"><div align="left"><a
href="selfevaluation.php">SELF EVALUATION </a></div></th>
    </tr>
    <tr>
      <th height="30" scope="row"><div align="left"><a
href="ADMresult.php">RESULT</a></div></th>
    </tr>
    <tr>
      <th height="23" scope="row"><div align="left">NEWS &amp;amp; ABU </div></th>
    </tr>
    <tr>
      <th height="159" scope="row"> </th>
    </tr>
  </table>
</div>
<div class="style7" id="Layer7" style="left: 120px; top: 193px;">
  <div align="center">
    <p align="center" class="style8">A WEB BASED UNIVERSITY ADMISSION
AND PLACEMENT NEURAL NETWORK MODEL</p>
  </div>
</div>
<div id="Layer6">
  <div id="Layer10"><span class="style7">AHMADU BELLO UNIVERSITY,
ZARIA</span></div>
  <h1 align="center" class="style2"> </h1>
  <p class="style2"> </p>
  <p class="style2"> </p>
  <p class="style2"> </p>
  <p class="style2"> </p>
  <p class="style2"> </p>
  <table width="1133" height="43" border="4">
    <tr>
      <td width="188" height="10" bordercolor="#009966" bgcolor="#009966"
scope="col"><div align="right"></div></td>
      <td bordercolor="#009966" bgcolor="#009966" scope="col"><div
align="center"></div>        <div align="center"></div>        <div
align="center"></div></td>
    </tr>
  </table>
  <p class="style2"> </p>
```


ABSTRACT
Every year the number of applicants seeking admission into Nigerian Universities increases by leaps and bounds although the Universities lack the commensurate facilities to meet the challenges of admitting the high number of applicants. For this reason, the admission officers have to manually evaluate every candidate’s data against the set admission requirements to screen the applicants in order to select the quota that their universities can accommodate. The procedures involved are very cumbersome, time consuming and prone to a lot of human errors and irregularities. Many candidates miss out on the admission every year, and the most painful aspect of this manual process is that many who are not qualified for a particular course end up being given placement into such courses while the more qualified ones are left out. Consequently, for lack of aptitude for the course, the students struggle through and many even resort to cheating their way through examinations and then graduate out of the Universities ill-equipped for the job market and the society. On the other hand, some of the less fortunate but qualified ones who are not given University admission year after year, become so frustrated over time and end up in hideous lifestyles. Whichever way, the society suffers and national growth is hindered.

In this work, a web-based model was designed to considerably take care of the above problems. From a neural network background, the system is developed to provide a time-efficient, detailed and unbiased automated procedure for selecting the most qualified candidates for admission into universities; ensure that qualified candidates who fail to meet the requirements for a particular course, are automatically placed into other courses for which they meet the admission requirements and where vacancies exist. The model also provides an avenue for students self-screening admission system.

</body>
</html>

```
<?php
//require('sesuser.php');?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>AHMADU BELLO UNIVERSITY, ZARIA </title>
<style type="text/css">
<!--
#Layer1 {
	position:absolute;
	width:199px;
	height:430px;
	z-index:1;
	left: 75px;
	top: 183px;
	background-color: #009966;
}
#Layer2 {
	position:absolute;
	width:456px;
	height:84px;
	z-index:2;
	left: 450px;
	top: 43px;
	border-color: #FFFFFF;
	border-top-style: solid;
	border-right-style: solid;
	border-bottom-style: solid;
	border-left-style: solid;
	border-top-width: thick;
	border-right-width: thick;
	border-bottom-width: thick;
	border-left-width: thick;
	background-color: #FFFFFF;
}
body {
	margin-left: 5%;
	margin-top: 10%;
	background-color: #FFFFFF;
}
.aerocss {
	padding-right: 40px;
	padding-left: 0px;
	font-size: 14px;
	font-weight: bold;
}
#Layer3 {
	position:absolute;
```

```
        width:529px;
        height:512px;
        z-index:3;
        left: 335px;
        top: 228px;
        background-color: #669999;
        border-right-color: #FFFFFF;
        border-width: 10;
        outline: 10;
        border-top-style: solid;
        border-right-style: solid;
        border-bottom-style: solid;
        border-left-style: solid;
        border-top-width: thick;
        border-right-width: thick;
        border-bottom-width: medium;
        border-left-width: thin;
        border-top-color: #FFFFFF;
        border-bottom-color: #FFFFFF;
        border-left-color: #FFFFFF;
        outline-width: thick;
        border-color: #000000;
}
.style2 {font-family: Arial, Helvetica, sans-serif}
#Layer4 {
        position:absolute;
        width:542px;
        height:43px;
        z-index:4;
        left: 909px;
        top: 161px;
}
#Layer5 {
        position:absolute;
        width:387px;
        height:318px;
        z-index:5;
        left: 892px;
        top: 291px;
        visibility: visible;
        overflow: auto;
        outline-color: #FFFFFF;
        outline-style: groove;
        outline-width: thick;
        outline: 20;
        background-color: #FFCCFF;
        color: #FF6666;
        border: medium solid #003399;
}
#Layer6 {
```

```
        position:absolute;
        width:1143px;
        height:135px;
        z-index:6;
        left: 75px;
        top: 22px;
}
.anytg18 {
        margin-left: .41cm;
}
.anytg15 {
        margin-left: 1.3cm;
}
tanyag1 {
        clip: rect(auto,auto,auto,2.5cm);
}
.any14 {
        margin-left: 1.2cm;
        background-color: #FFFFFF;
        font-size: 14px;
        color: #FF6600;
        line-height: 5cm;
}
.any36 {
        margin-left: 0.35cm;
        line-height: normal;
        font-size: 14px;
        width: 3.5cm;
}
.any42 {
        margin-left: 3.77cm;
        background-color: #FFFFFF;
        font-size: 14px;
        color: #FF6600;
}
.any366 {
        margin-left: 3.67cm;
        background-color: #FFFFFF;
        font-size: 14px;
        color: #FF6600;
}
.anya1 {
        width: 5cm;
        clip: rect(auto,auto,auto,auto);
        margin-left: 1.08cm;
}
.anytag2 {
        width: 1.5cm;
}
.anytg32 {
```

```
	margin-left: 4.07cm;
}
.anytg38 {
	margin-left: 3.8cm;
}
.anytg41 {
	margin-left: 2.17cm;
}
.anytg27 {
	margin-left: 2.3cm;
}
.anytg34 {
	margin-left: 3.16cm;
}
.anytg4 {
	margin-left: 5cm;
	width: 2.5cm;
}
.anytg388 {
	margin-left: 2.5cm;
	width: 3.5cm;
	font-size: 14px;
}
.anygrd {
	width: 1.6cm;
	margin-left: 0.5cm;
}
.style6 {
	color: #FF6600;
	font-family: "Times New Roman", Times, serif;
	font-size: large;
}
#Layer7 {
	position:absolute;
	width:237px;
	height:37px;
	z-index:7;
	left: 978px;
	top: 130px;
}
.anysc {
	margin-left: 1cm;
	width: 1cm;
}
.anycs {
	width: 3.8cm;
}
.antgh {
	display: none;
	visibility: hidden;
```

```
}
-->
</style>
<script type="text/JavaScript">
<!--
function MM_timelinePlay(tmLnName, myID) { //v1.2
  //Copyright 1998, 1999, 2000, 2001, 2002, 2003, 2004 Macromedia, Inc. All rights
reserved.
  var
i,j,tmLn,props,keyFrm,sprite,numKeyFr,firstKeyFr,propNum,theObj,firstTime=false;
  if (document.MM_Time == null) MM_initTimelines(); //if *very* 1st time
  tmLn = document.MM_Time[tmLnName];
  if (myID == null) { myID = ++tmLn.ID; firstTime=true;}//if new call, incr ID
  if (myID == tmLn.ID) { //if Im newest
    setTimeout('MM_timelinePlay('"+tmLnName+'",'+myID+')',tmLn.delay);
    fNew = ++tmLn.curFrame;
    for (i=0; i<tmLn.length; i++) {
      sprite = tmLn[i];
      if (sprite.charAt(0) == 's') {
        if (sprite.obj) {
          numKeyFr = sprite.keyFrames.length; firstKeyFr = sprite.keyFrames[0];
          if (fNew >= firstKeyFr && fNew <= sprite.keyFrames[numKeyFr-1]) {//in range
            keyFrm=1;
            for (j=0; j<sprite.values.length; j++) {
              props = sprite.values[j];
              if (numKeyFr != props.length) {
                if (props.prop2 == null) sprite.obj[props.prop] = props[fNew-firstKeyFr];
                else        sprite.obj[props.prop2][props.prop] = props[fNew-firstKeyFr];
              } else {
                while (keyFrm<numKeyFr && fNew>=sprite.keyFrames[keyFrm])
keyFrm++;
                if (firstTime || fNew==sprite.keyFrames[keyFrm-1]) {
                  if (props.prop2 == null) sprite.obj[props.prop] = props[keyFrm-1];
                  else        sprite.obj[props.prop2][props.prop] = props[keyFrm-1];
        } } } } }
      } else if (sprite.charAt(0)=='b' && fNew == sprite.frame) eval(sprite.value);
      if (fNew > tmLn.lastFrame) tmLn.ID = 0;
  } }
}

function MM_timelineGoto(tmLnName, fNew, numGotos) { //v2.0
  //Copyright 1998, 1999, 2000, 2001, 2002, 2003, 2004 Macromedia, Inc. All rights
reserved.
  var i,j,tmLn,props,keyFrm,sprite,numKeyFr,firstKeyFr,lastKeyFr,propNum,theObj;
  if (document.MM_Time == null) MM_initTimelines(); //if *very* 1st time
  tmLn = document.MM_Time[tmLnName];
  if (numGotos != null)
    if (tmLn.gotoCount == null) tmLn.gotoCount = 1;
    else if (tmLn.gotoCount++ >= numGotos) {tmLn.gotoCount=0; return}
  jmpFwd = (fNew > tmLn.curFrame);
```

```
  for (i = 0; i < tmLn.length; i++) {
    sprite = (jmpFwd)? tmLn[i] : tmLn[(tmLn.length-1)-i]; //count bkwds if jumping
back
    if (sprite.charAt(0) == "s") {
      numKeyFr = sprite.keyFrames.length;
      firstKeyFr = sprite.keyFrames[0];
      lastKeyFr = sprite.keyFrames[numKeyFr - 1];
      if ((jmpFwd && fNew<firstKeyFr) || (!jmpFwd && lastKeyFr<fNew)) continue;
//skip if untouchd
      for (keyFrm=1; keyFrm<numKeyFr && fNew>=sprite.keyFrames[keyFrm];
keyFrm++);
      for (j=0; j<sprite.values.length; j++) {
        props = sprite.values[j];
        if (numKeyFr == props.length) propNum = keyFrm-1 //keyframes only
        else propNum = Math.min(Math.max(0,fNew-firstKeyFr),props.length-1); //or
keep in legal range
        if (sprite.obj != null) {
          if (props.prop2 == null) sprite.obj[props.prop] = props[propNum];
          else        sprite.obj[props.prop2][props.prop] = props[propNum];
      } }
    } else if (sprite.charAt(0)=='b' && fNew == sprite.frame) eval(sprite.value);
  }
  tmLn.curFrame = fNew;
  if (tmLn.ID == 0) eval('MM_timelinePlay(tmLnName)');
}

function MM_initTimelines() { //v4.0
    //MM_initTimelines() Copyright 1997 Macromedia, Inc. All rights reserved.
    var ns = navigator.appName == "Netscape";
    var ns4 = (ns && parseInt(navigator.appVersion) == 4);
    var ns5 = (ns && parseInt(navigator.appVersion) > 4);
    var macIE5 = (navigator.platform ? (navigator.platform == "MacPPC") : false) &&
(navigator.appName == "Microsoft Internet Explorer") &&
(parseInt(navigator.appVersion) >= 4);
    document.MM_Time = new Array(1);
    document.MM_Time[0] = new Array(2);
    document.MM_Time["Timeline1"] = document.MM_Time[0];
    document.MM_Time[0].MM_Name = "Timeline1";
    document.MM_Time[0].fps = 2;
    document.MM_Time[0][0] = new String("behavior");
    document.MM_Time[0][0].frame = 16;
    document.MM_Time[0][0].value = "MM_timelineGoto('Timeline1','1')";
    document.MM_Time[0][1] = new String("sprite");
    document.MM_Time[0][1].slot = 1;
    if (ns4)
        document.MM_Time[0][1].obj = document["Layer4"];
    else if (ns5)
        document.MM_Time[0][1].obj = document.getElementById("Layer4");
    else
        document.MM_Time[0][1].obj = document.all ? document.all["Layer4"] : null;
```

```
   document.MM_Time[0][1].keyFrames = new Array(1, 15);
   document.MM_Time[0][1].values = new Array(3);
   if (ns5 || macIE5)
      document.MM_Time[0][1].values[0] = new Array("75px", "123px", "170px",
"218px", "266px", "313px", "361px", "408px", "456px", "504px", "551px", "599px",
"647px", "694px", "742px");
   else
      document.MM_Time[0][1].values[0] = new
Array(75,123,170,218,266,313,361,408,456,504,551,599,647,694,742);
   document.MM_Time[0][1].values[0].prop = "left";
   if (ns5 || macIE5)
      document.MM_Time[0][1].values[1] = new Array("53px", "54px", "55px",
"56px", "57px", "58px", "59px", "60px", "60px", "61px", "62px", "63px", "64px",
"65px", "66px");
   else
      document.MM_Time[0][1].values[1] = new
Array(53,54,55,56,57,58,59,60,60,61,62,63,64,65,66);
   document.MM_Time[0][1].values[1].prop = "top";
   if (!ns4) {
      document.MM_Time[0][1].values[0].prop2 = "style";
      document.MM_Time[0][1].values[1].prop2 = "style";
   }
   if (ns5 || macIE5)
      document.MM_Time[0][1].values[2] = new Array("1040px", "1001px", "962px",
"923px", "885px", "846px", "807px", "769px", "730px", "691px", "652px", "614px",
"575px", "536px", "498px");
   else
      document.MM_Time[0][1].values[2] = new
Array(1040,1001,962,923,885,846,807,769,730,691,652,614,575,536,498);
   document.MM_Time[0][1].values[2].prop = "width";
   if (!ns4)
      document.MM_Time[0][1].values[2].prop2 = "style";
   document.MM_Time[0].lastFrame = 16;
   for (i=0; i<document.MM_Time.length; i++) {
      document.MM_Time[i].ID = null;
      document.MM_Time[i].curFrame = 0;
      document.MM_Time[i].delay = 1000/document.MM_Time[i].fps;
   }
}
//-->
</script>
<link href="Confirm.css" rel="stylesheet" type="text/css">
<style type="text/css">
<!--
.style8 {font-size: 24px}
#Layer8 {
        position:absolute;
        width:200px;
        height:50px;
        z-index:8;
```

```
        left: 744px;
        top: 209px;
}
#Layer9 {
        position:absolute;
        width:311px;
        height:41px;
        z-index:9;
        left: 288px;
        top: 82px;
}
.style9 {font-size: xx-large}
#Layer10 {
        position:absolute;
        width:200px;
        height:268px;
        z-index:9;
        left: 844px;
        top: 373px;
}
.style10 {
        color: #993333;
        font-weight: bold;
}
#Layer11 {
        position:absolute;
        width:200px;
        height:30px;
        z-index:10;
        left: 361px;
        top: 115px;
        background-color: #CCFF66;
}
#Layer12 {
        position:absolute;
        width:200px;
        height:35px;
        z-index:10;
        left: 210px;
        top: -4px;
}
#Layer13 {
        position:absolute;
        width:165px;
        height:31px;
        z-index:10;
        left: 702px;
        top: 260px;
        background-color: #CCFF66;
}
```

```
#Layer14 {
	position:absolute;
	width:200px;
	height:32px;
	z-index:11;
	left: 121px;
	top: 193px;
}
#Layer15 {
	position:absolute;
	width:200px;
	height:29px;
	z-index:11;
	left: 435px;
	top: 428px;
}
#Layer16 {
	position:absolute;
	width:200px;
	height:28px;
	z-index:11;
}
#Layer17 {
	position:absolute;
	width:366px;
	height:31px;
	z-index:11;
	left: 336px;
	top: 260px;
	background-color: #CCFF66;
}
.style12 {color: #993333}
#Layer18 {
	position:absolute;
	width:271px;
	height:309px;
	z-index:12;
	left: 3px;
	top: 150px;
}
#Layer19 {
	position:absolute;
	width:245px;
	height:309px;
	z-index:12;
	left: 277px;
	top: 150px;
}
#Layer20 {
	position:absolute;
```

```
        width:200px;
        height:115px;
        z-index:12;
        left: -234px;
        top: 401px;
}
-->
</style>
</head>

<body onLoad="MM_timelinePlay('Timeline1')">
<div id="Layer4" style="left: 75px; top: 53px; width: 1040px;"><span
class="style8">A WEB BASED UNIVERSITY ADMISSION AND PLACEMENT
NEURAL NETWORK MODEL</span></div>
<div id="Layer6">
  <h1 align="center" class="style2">AHMADU BELLO UNIVERSITY, ZARIA </h1>
  <p class="style2"> </p>
  <div class="style9" id="Layer9">O LEVEL  RESULT </div>
  <p class="style2">  </p>
  <table width="1133" height="27" border="4">
    <tr>
      <td width="188" height="10" bordercolor="#009966" bgcolor="#009966"
scope="col"><div align="right"></div></td>
      <td width="204" bordercolor="#009966" bgcolor="#009966" scope="col"><div
align="center"><a href="umeservices.php">SUBMIT UME RESULT  </a></div></td>
      <td width="233" bordercolor="#009966" bgcolor="#009966" scope="col"><div
align="center"><a href="olservices.php">SUBMIT O LEVEL RESULT</a><a
href="approdeparturea.php"></a></div></td>
      <td width="160" bordercolor="#009966" bgcolor="#009966" scope="col"><a
href="postume.php">SUBMIT POST UME </a></td>
      <td width="134" bordercolor="#009966" bgcolor="#009966"
scope="col"> </td>
      <td width="168" bordercolor="#009966" bgcolor="#009966" scope="col"><div
align="center"></div></td>
    </tr>
  </table>
  <p class="style2"> </p>
</div>
<p> </p>
<div id="Layer3">
  <p>
    <?php
/*
$db = mysql_connect("localhost", "******t", "*********");
mysql_select_db("stadm",$db) or die('<hr />MySQL Error: ' .mysql_error(). '<hr />');


if(isset($_POST["CS"])) $CS = $_POST["CS"];
if (isset($_POST['update']))
    {
```

```
   $ETA = $_POST['ETA'];
   $TAOVCLR = $_POST['TAOVCLR'];
   $FL  = $_POST['FL'];
   $CS = $_POST['CS'];
        $TAS = $_POST['TAS'];
   $TAC = $_POST['TAC'];
   $ATL = $_POST['ATL'];
   $LA = $_POST['LA'];
        $RP = $_POST['RP'];
   $EAT = $_POST['EAT'];
   $CPT = $_POST['CPT'];
   $DU = $_POST['DU'];
            $sql = "UPDATE aeroarrival SET ETA ='$ETA',TAOVCLR
='$TAOVCLR',FL ='$FL',CS ='$CS', TAS ='$TAS',TAC ='$TAC', ATL ='$ATL',LA
='$LA', RP ='$RP',EAT ='$EAT', CPT ='$CPT',DU ='$DU' WHERE CS ='$CS'";
   $result = mysql_query($sql);
        mysql_query($sql) or die('<hr />MySQL Error: ' .mysql_error(). '<hr />');

   echo "Thank you! Information updated.\n";
   }

        if (isset($_POST['addnew']))
{
        $ETA = $_POST['ETA'];
   $TAOVCLR = $_POST['TAOVCLR'];
   $FL  = $_POST['FL '];
   $CS = $_POST['CS'];
        $TAS = $_POST['TAS'];
   $TAC = $_POST['TAC'];
   $ATL = $_POST['ATL'];
   $LA = $_POST['LA'];
        $RP = $_POST['RP'];
   $EAT = $_POST['EAT'];
   $CPT = $_POST['CPT'];
   $DU = $_POST['DU'];

        $sql = "INSERT INTO aeroarrival (ETA ,TAOVCLR,FL ,CS , TAS,TAC ,
ATL,LA ,RP ,EAT, CPT ,DU) VALUES
        ('$ETA','$TAOVCLR','$FL','$CS','$TAS','$TAC','$ATL','$LA','$RP','$EAT',
'$CPT','$DU')";
        $result = mysql_query($sql);

        echo "Thank you! Information entered.\n";
}

if (isset($_POST['delete']))
{
        $sql = "DELETE FROM aeroarrival WHERE CS='$CS'";
        $result = mysql_query($sql);
```

```
        echo " Record deleted!<p>";
}

if (isset($CS))
   {
          $sql = "SELECT * FROM aeroarrival WHERE CS='$CS'";
   $result = mysql_query($sql);
        mysql_query($sql) or die('<hr />MySQL Error: ' .mysql_error(). '<hr />');
   $myrow = mysql_fetch_array($result);
        $ETA = $myrow['0'];
   $TAOVCLR = $myrow['1'];
   $FL  = $myrow['2'];
   $CS = $myrow['3'];
        $TAS = $myrow['4'];
   $TAC = $myrow['5'];
   $ATL = $myrow['6'];
   $LA = $myrow['7'];
        $RP = $myrow['8'];
   $EAT = $myrow['9'];
   $CPT = $myrow['10'];
   $DU = $myrow['11'];
        }
        //if(empty($cs))
        //{
        //        echo " Record not Available!<p>";
        //}
        */
        ?>
   <?php
        $db = mysql_connect("localhost", "******", "**********");
mysql_select_db("stadm",$db) or die('<hr />MySQL Error: ' .mysql_error(). '<hr />');

//if(isset($_POST["dept"])) $dept = $_POST["CS"];

  if(isset($_POST['SUBMIT']))
{
        $UMENO = $_POST['UMENO'];
        $YEXAMS = $_POST['YEXAMS'];
        $EXAMNO = $_POST['EXAMNO'];
        $NECOM = $_POST['NECOM'];
        $NECON = $_POST['NECON'];
        $WASCM = $_POST['WASCM'];
        $WASCN = $_POST['WASCN'];
        $COS1 = $_POST['COS1'];
        $COS2 = $_POST['COS2'];
        $COS3 = $_POST['COS3'];
        $COS4 = $_POST['COS4'];
        $COS5 = $_POST['COS5'];
        $GRADE1 = $_POST['GRADE1'];
        $GRADE2 = $_POST['GRADE2'];
```

```
$GRADE3 = $_POST['GRADE3'];
$GRADE4 = $_POST['GRADE4'];
$GRADE5 = $_POST['GRADE5'];
$YEXAMS2 = $_POST['YEXAMS2'];
$EXAMNO2 = $_POST['EXAMNO2'];
$NECOM2 = $_POST['NECOM2'];
$NECON2 = $_POST['NECON2'];
$WASCM2 = $_POST['WASCM2'];
$WASCN2 = $_POST['WASCN2'];
$COS21 = $_POST['COS21'];
$COS22 = $_POST['COS22'];
$COS23 = $_POST['COS23'];
$COS24 = $_POST['COS24'];
$COS25 = $_POST['COS25'];
$GRADE21 = $_POST['GRADE21'];
$GRADE22 = $_POST['GRADE22'];
$GRADE23 = $_POST['GRADE23'];
$GRADE24 = $_POST['GRADE24'];
$GRADE25 = $_POST['GRADE25'];
$PNT = $_POST['PNT'];
if($COS1 == "Select Subject"){
 $COS1 = " ";
 $GRADE1 = " ";
 }
 if($COS2 == "Select Subject"){
 $COS2 = " ";
 $GRADE2 = " ";
 }
 if($COS3 == "Select Subject"){
 $COS3 = " ";
 $GRADE3 = " ";
 }
 if($COS4 == "Select Subject"){
 $COS4 = " ";
 $GRADE4 = " ";
 }
  if($COS5 == "Select Subject"){
 $COS5 = " ";
 $GRADE5 = " ";
 }
 if($COS21 == "Select Subject"){
 $COS21 = " ";
 $GRADE21 = " ";
 }
 if($COS22 == "Select Subject"){
 $COS22 = " ";
 $GRADE22 = " ";
 }
 if($COS23 == "Select Subject"){
 $COS23 = " ";
```

```
$GRADE23 = " ";
}
if($COS24 == "Select Subject"){
$COS24 = " ";
$GRADE24 = " ";
}
 if($COS25 == "Select Subject"){
$COS25 = " ";
$GRADE25 = " ";
}


$sql = "INSERT INTO
oresult(UMENO,YEXAMS,EXAMNO,NECOM,NECON,WASCM,WASCN,COS1,C
OS2,COS3,COS4,COS5,GRADE1,GRADE2,GRADE3,GRADE4,GRADE5,YEXAMS
2,EXAMNO2,NECOM2,NECON2,WASCM2,WASCN2,COS21,COS22,COS23,COS2
4,COS25,GRADE21,GRADE22,GRADE23,GRADE24,GRADE25,PNT)
VALUES('$UMENO','$YEXAMS','$EXAMNO','$NECOM','$NECON','$WASCM','$
WASCN','$COS1','$COS2','$COS3','$COS4','$COS5','$GRADE1','$GRADE2','$GRA
DE3','$GRADE4','$GRADE5','$YEXAMS2','$EXAMNO2','$NECOM2','$NECON2','$
WASCM2','$WASCN2','$COS21','$COS22','$COS23','$COS24','$COS25','$GRADE21
','$GRADE22','$GRADE23','$GRADE24','$GRADE25','$PNT') " or die('<hr />MySQL
Error: ' .mysql_error(). '<hr />');
$result = mysql_query($sql);

//$sql = "INSERT INTO
olresult(UMENO,YEXAMS,EXAMNO,NECOM,NECON,WASCM,WASCN,COS1,C
OS2,COS3,COS4,COS5,GRADE1,GRADE2,GRADE3,GRADE4,GRADE5,YEXAMS
2,EXAMNO2,NECOM2,NECON2,WASCM2,WASCN2,COS21,COS22,COS23,COS2
4,COS25,GRADE21,GRADE22,GRADE23,GRADE24,GRADE25,PNT)VALUES
        ('$UMENO','$YEXAMS','$EXAMNO','$NECOM','$NECON','$WASCM','$W
ASCN','$COS1','$COS2','$COS3','$COS4','$COS5','$GRADE1','$GRADE2','$GRADE
3','$GRADE4','$GRADE5','$YEXAMS2','$EXAMNO2','$NECOM2','$NECON2','$W
ASCM2','$WASCN2','$COS21','$COS22','$COS23','$COS24','$COS25','$GRADE21','
$GRADE22','$GRADE23','$GRADE24','$GRADE25','$PNT')";

//$result = mysql_query($sql);

//$sql = "INSERT INTO olevel2s(
YEXAMS2,EXAMNO2,NECOM2,NECON2,WASCM2,WASCN2,COS21,COS22,CO
S23,COS24,COS25,GRADE21,GRADE22,GRADE23,GRADE24,GRADE25,)VALUE
S
('$YEXAMS2','$EXAMNO2','$NECOM2','$NECON2','$WASCM2','$WASCN2','$CO
S21','$COS22','$COS23','$COS24','$COS25','$GRADE21','$GRADE22','$GRADE23','
$GRADE24','$GRADE25')";

//$result = mysql_query($sql);

//$sql = "INSERT INTO oresult(UMENO, PNT) VALUES('$UMENO','$PNT');
```

```
if(isset($_POST['RESET'])){
  //if(isset($_POST['SUBMIT']))

$UMENO = $_POST['UMENO'];
        $YEXAMS = $_POST['YEXAMS'];
        $EXAMNO = $_POST['EXAMNO'];
        $NECOM = $_POST['NECOM'];
        $NECON = $_POST['NECON'];
        $WASCM = $_POST['WASCM'];
        $WASCN = $_POST['WASCN'];
        $COS1 = $_POST['COS1'];
        $COS2 = $_POST['COS2'];
        $COS3 = $_POST['COS3'];
        $COS4 = $_POST['COS4'];
        $COS5 = $_POST['COS5'];
        $GRADE1 = $_POST['GRADE1'];
        $GRADE2 = $_POST['GRADE2'];
        $GRADE3 = $_POST['GRADE3'];
        $GRADE4 = $_POST['GRADE4'];
        $GRADE5 = $_POST['GRADE5'];
        $YEXAMS2 = $_POST['YEXAMS2'];
        $EXAMNO2 = $_POST['EXAMNO2'];
        $NECOM2 = $_POST['NECOM2'];
        $NECON2 = $_POST['NECON2'];
        $WASCM2 = $_POST['WASCM2'];
        $WASCN2 = $_POST['WASCN2'];
        $COS21 = $_POST['COS21'];
        $COS22 = $_POST['COS22'];
        $COS23 = $_POST['COS23'];
        $COS24 = $_POST['COS24'];
        $COS25 = $_POST['COS25'];
        $GRADE21 = $_POST['GRADE21'];
        $GRADE22 = $_POST['GRADE22'];
        $GRADE23 = $_POST['GRADE23'];
        $GRADE24 = $_POST['GRADE24'];
        $GRADE25 = $_POST['GRADE25'];
        $PNT = $_POST['PNT'];

if ($GRADE1 == "A") $PNT1 =  5;
elseif ($GRADE1 == "B") $PNT1 =  4;
elseif ($GRADE1 == "C") $PNT1 =  3;
elseif ($GRADE1 == "D") $PNT1 =  2;
elseif ($GRADE1 == "E") $PNT1 =  1;
else $PNT1 =  0;
        echo $PNT1;

if ($GRADE2 == "A") $PNT2 =  5;
elseif ($GRADE2 == "B") $PNT2 =  4;
elseif ($GRADE2 == "C") $PNT2 =  3;
elseif ($GRADE2 == "D") $PNT2 =  2;
```

```
elseif ($GRADE2 == "E") $PNT2 =  1;
else $PNT2 =  0;
        echo $PNT2;

if ($GRADE3 == "A") $PNT3 =  5;
elseif ($GRADE3 == "B") $PNT3 =  4;
elseif ($GRADE3 == "C") $PNT3 =  3;
elseif ($GRADE3 == "D") $PNT3 =  2;
elseif ($GRADE3 == "E") $PNT3 =  1;
else $PNT3 =  0;
        echo $PNT3;

if ($GRADE4 == "A") $PNT4 =  5;
elseif ($GRADE4 == "B") $PNT4 =  4;
elseif ($GRADE4 == "C") $PNT4 =  3;
elseif ($GRADE4 == "D") $PNT4 =  2;
elseif ($GRADE4 == "E") $PNT4 =  1;
else $PNT4 =  0;
        echo $PNT4;

if ($GRADE5 == "A") $PNT5 =  5;
elseif ($GRADE5 == "B") $PNT5 =  4;
elseif ($GRADE5 == "C") $PNT5 =  3;
elseif ($GRADE5 == "D") $PNT5 =  2;
elseif ($GRADE5 == "E") $PNT5 =  1;
else $PNT5 =  0;
        echo $PNT5;


if ($GRADE21 == "A") $PNT21 =  5;
elseif ($GRADE21 == "B") $PNT21 =  4;
elseif ($GRADE21 == "C") $PNT21 =  3;
elseif ($GRADE21 == "D") $PNT21 =  2;
elseif ($GRADE21 == "E") $PNT21 =  1;
else $PNT21 =  0;
        echo $PNT21;

if ($GRADE22 == "A") $PNT22 =  5;
elseif ($GRADE22 == "B") $PNT22 =  4;
elseif ($GRADE22 == "C") $PNT22 =  3;
elseif ($GRADE22 == "D") $PNT22 =  2;
elseif ($GRADE22 == "E") $PNT22 =  1;
else $PNT22 =  0;
        echo $PNT22;

if ($GRADE23 == "A") $PNT23 =  5;
elseif ($GRADE23 == "B") $PNT23 =  4;
elseif ($GRADE23 == "C") $PNT23 =  3;
elseif ($GRADE23 == "D") $PNT23 =  2;
elseif ($GRADE23 == "E") $PNT23 =  1;
```

```
else $PNT23 =  0;
        echo $PNT23;

if ($GRADE24 == "A") $PNT24 =  5;
elseif ($GRADE24 == "B") $PNT24 =  4;
elseif ($GRADE24 == "C") $PNT24 =  3;
elseif ($GRADE24 == "D") $PNT24 =  2;
elseif ($GRADE24 == "E") $PNT24 =  1;
else $PNT24 =  0;
        echo $PNT24;

if ($GRADE25 == "A") $PNT25 =  5;
elseif ($GRADE25 == "B") $PNT25 =  4;
elseif ($GRADE25 == "C") $PNT25 =  3;
elseif ($GRADE25 == "D") $PNT25 =  2;
elseif ($GRADE25 == "E") $PNT25 =  1;
else $PNT25 =  0;
        echo $PNT25;
        $PNT =
$PNT1+$PNT2+$PNT3+$PNT4+$PNT5+$PNT21+$PNT22+$PNT23+$PNT24+$PN
T25;
        echo $PNT;
}
}
 //}
        ?>


  <form name="form1" method="post" action="">

   <p>UME Number
     :
    <label>
    <input name="UMENO" type="text" class="anytg18" value="<?php echo $UMENO
?>">
    </label>
    <br>
    <br>
    <label></label>
    <label></label>
    <br>
    <br>
  </p>
    <p align= "left">Year of Exams  :
      <label>
      <select name="YEXAMS" class="any36">
        <option>Select Year of Exams</option>
        <option>1995</option>
        <option>1996</option>
        <option>1997</option>
```

```
      <option>1998</option>
      <option>1999</option>
      <option>2000</option>
      <option>2001</option>
      <option>2002</option>
      <option>2003</option>
      <option>2004</option>
      <option>2005</option>
      <option>2006</option>
      <option>2007</option>
      <option>2008</option>
    </select>
    </label>
    <label>
    <select name="YEXAMS2" class="anytg388">
          <option>Select Year of Exams</option>
      <option>1995</option>
      <option>1996</option>
      <option>1997</option>
      <option>1998</option>
      <option>1999</option>
      <option>2000</option>
      <option>2001</option>
      <option>2002</option>
      <option>2003</option>
      <option>2004</option>
      <option>2005</option>
      <option>2006</option>
      <option>2007</option>
      <option>2008</option>
    </select>
    </label>
    <br>
   Exam no. :
    <input name="EXAMNO" type="text" class="any14" value="<?php echo
$EXAMNO ?>">
    <label>
    <input name="EXAMNO2" type="text" class="anytg41" value="<?php echo
$EXAMNO2 ?>">
    </label>

   <div id="Layer19">

    <p>Name of Exam </p>
    <p>
      <label>
      <input name="NECOM2" type="radio" value="NECO May/June">
 NECO May/June   </label>
    <br>
      <label>
```

```
      <input name="NECON2" type="radio" value="NECO Nov/Dec">
      </label>
      NECO Nov/Dec     <br>
          <label>
      <input name="WASCM2" type="radio" value="WASC May/June">
      </label>
     WASC May/June
      <label>     <br>
      <input name="WASCN2" type="radio" value="WASC Nov/Dec">
WASC Nov/Dec      </label>     <br>
  Subjects
    <p>
      <label>
      <select name="COS21" class="anycs">
                  <option>Select Subject</option>
        <option>English</option>
        <option>Mathematics</option>
        <option>Physics</option>
        <option>Biology</option>
        <option>Chemistry</option>
        <option>Geography</option>
        <option>Economics</option>
        <option>Agric Science</option>
      </select>
      <select name="GRADE21" class="anygrd">
        <option>Grade</option>
        <option>A</option>
        <option>B</option>
        <option>C</option>
        <option>D</option>
        <option>E</option>
      </select>
      </label>
        <br>
      <label>
      <select name="COS22" class="anycs">
                  <option>Select Subject</option>
        <option>English</option>
        <option>Mathematics</option>
        <option>Physics</option>
        <option>Biology</option>
        <option>Chemistry</option>
        <option>Geography</option>
        <option>Economics</option>
        <option>Agric Science</option>
      </select>
      <select name="GRADE22" class="anygrd">
        <option>Grade</option>
                   <option>A</option>
        <option>B</option>
```

```
  <option>C</option>
  <option>D</option>
  <option>E</option>
</select>
</label>
  <br>
<label>
<select name="COS23" class="anycs">
            <option>Select Subject</option>
  <option>English</option>
  <option>Mathematics</option>
  <option>Physics</option>
  <option>Biology</option>
  <option>Chemistry</option>
  <option>Geography</option>
  <option>Economics</option>
  <option>Agric Science</option>
</select>
<select name="GRADE23" class="anygrd">
  <option>Grade</option>
            <option>A</option>
  <option>B</option>
  <option>C</option>
  <option>D</option>
  <option>E</option>
</select>
</label>
  <br>
<label>
<select name="COS24" class="anycs">
            <option>Select Subject</option>
  <option>English</option>
  <option>Mathematics</option>
  <option>Physics</option>
  <option>Biology</option>
  <option>Chemistry</option>
  <option>Geography</option>
  <option>Economics</option>
  <option>Agric Science</option>
</select>
<select name="GRADE24" class="anygrd">
  <option>Grade</option>
            <option>A</option>
  <option>B</option>
  <option>C</option>
  <option>D</option>
  <option>E</option>
</select>
</label>
  <br>
```

```
      <label>
      <select name="COS25" class="anycs">
                  <option>Select Subject</option>
        <option>English</option>
        <option>Mathematics</option>
        <option>Physics</option>
        <option>Biology</option>
        <option>Chemistry</option>
        <option>Geography</option>
        <option>Economics</option>
        <option>Agric Science</option>
      </select>
      <select name="GRADE25" size="1" class="anygrd">
        <option>Grade</option>
                   <option>A</option>
        <option>B</option>
        <option>C</option>
        <option>D</option>
        <option>E</option>
      </select>
      </label>
  </div>
  <p> </p>
  <div id="Layer18">
    <p>Name of Exam </p>
    <p>
      <label>
      <input name="NECOM" type="radio" value="NECO May/June">
NECO May/June   </label>
   <br>
      <label>
      <input name="NECON" type="radio" value="NECO Nov/Dec">
      </label>
      NECO Nov/Dec     <br>
          <label>
      <input name="WASCM" type="radio" value="WASC May/June">
      </label>
    WASC May/June
      <label>     <br>
      <input name="WASCN" type="radio" value="WASC Nov/Dec">
WASC Nov/Dec      </label>     <br>

    <p>
      <label>
      <select name="COS1" class="anycs">
        <option>Select Subject</option>
        <option>English</option>
        <option>Mathematics</option>
        <option>Physics</option>
        <option>Biology</option>
```

```
  <option>Chemistry</option>
  <option>Geography</option>
  <option>Economics</option>
  <option>Agric Science</option>
</select>
<select name="GRADE1" class="anygrd">
  <option>Grade</option>
  <option>A</option>
  <option>B</option>
  <option>C</option>
  <option>D</option>
  <option>E</option>
</select>
</label>
  <br>
<label>
<select name="COS2" class="anycs">
  <option>Select Subject</option>
              <option>English</option>
  <option>Mathematics</option>
  <option>Physics</option>
  <option>Biology</option>
  <option>Chemistry</option>
  <option>Geography</option>
  <option>Economics</option>
  <option>Agric Science</option>
</select>
<select name="GRADE2" class="anygrd">
  <option>Grade</option>
              <option>A</option>
  <option>B</option>
  <option>C</option>
  <option>D</option>
  <option>E</option>
</select>
</label>
   <br>
<label>
<select name="COS3" class="anycs">
              <option>Select Subject</option>
             <option>English</option>
  <option>Mathematics</option>
  <option>Physics</option>
  <option>Biology</option>
  <option>Chemistry</option>
  <option>Geography</option>
  <option>Economics</option>
  <option>Agric Science</option>
  <option>Select Subject</option>
</select>
```

```
<select name="GRADE3" class="anygrd">
  <option>Grade</option>
              <option>A</option>
  <option>B</option>
  <option>C</option>
  <option>D</option>
  <option>E</option>
</select>
</label>
   <br>
<label>
<select name="COS4" class="anycs">
              <option>Select Subject</option>
              <option>English</option>
  <option>Mathematics</option>
  <option>Physics</option>
  <option>Biology</option>
  <option>Chemistry</option>
  <option>Geography</option>
  <option>Economics</option>
  <option>Agric Science</option>
  <option>Select Subject</option>
</select>
<select name="GRADE4" class="anygrd">
  <option>Grade</option>
              <option>A</option>
  <option>B</option>
  <option>C</option>
  <option>D</option>
  <option>E</option>
</select>
</label>
   <br>
<label>
<select name="COS5" class="anycs">
<option>Select Subject</option>
              <option>English</option>
  <option>Mathematics</option>
  <option>Physics</option>
  <option>Biology</option>
  <option>Chemistry</option>
  <option>Geography</option>
  <option>Economics</option>
  <option>Agric Science</option>
  <option>Select Subject</option>
</select>
<select name="GRADE5" size="1" class="anygrd">
  <option>Grade</option>
              <option>A</option>
  <option>B</option>
```

```
          <option>C</option>
          <option>D</option>
          <option>E</option>
        </select>
        </label>
    </div>
    <p> </p>
    <label></label>
    <label><br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
    <br>
  </label>
    <label>    <br>
    <br>
    <br>
    </label>
    <p>
          <input type="Submit" name="addnew" value="ADDNEW">
          <input type="Submit" name="SUBMIT" value="SUBMIT">
          <input type="Submit" name="delete" value="DELETE">
          <input type="Submit" name="RESET" value="RESET">
       </p>
  </form>
</div>
<div id="Layer7">
  <p>
    <?PHP
echo DATE('H : i : s,  jS F Y');

?>
  </p>
  <p>     </p>
</div>
<div id="Layer1">
  <table width="200" height="427" border="2">
    <tr>
      <th height="30" scope="col"><div align="left"><a
href="home.php">ABOUT</a></div></th>
    </tr>
    <tr>
      <th height="30" scope="row"><div align="left"><a
href="umeservices.php">SERVICES</a></div></th>
    </tr>
    <tr>
```

```
    <th height="30" scope="row"><div align="left"><a
href="ADMumeservices.php">ADMINISTRATION</a></div></th>
  </tr>
  <tr>
    <th height="23" scope="row"><div align="left"><a
href="accebdeparture.php">COURSES </a></div></th>
  </tr>
  <tr>
    <th height="30" scope="row"><div align="left"><a
href="separation.php">REQUIREMENT </a></div></th>
  </tr>
  <tr>
    <th height="30" scope="row"><div align="left"><a
href="selfevaluation.php">SELF EVALUATION </a></div></th>
  </tr>
  <tr>
    <th height="30" scope="row"><div align="left"><a
href="ADMresult.php">RESULT</a></div></th>
  </tr>
  <tr>
    <th height="23" scope="row"><div align="left">NEWS &amp;amp; ABU </div></th>
  </tr>
  <tr>
    <th height="159" scope="row"> </th>
  </tr>
 </table>
</div>
<div class="style10" id="Layer13">
 <div align="left">Second Sitting </div>
</div>
<div id="Layer17">
 <div align="center" class="style10">
  <div align="left">First Sitting </div>
 </div>
</div>
</body>
</html>
```