TITLE PAGE

AN ONLINE-APPLICATION FOR INDUSTRIAL TRAINING

BY

HAJARA JUMMAI MOHAMMED - IBRAHIM

PGD/MCS/2004/2005/1164

SUBMITTED TO THE POST GRADUATE SCHOOL, IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF POST GRADUATE DIPLOMA IN COMPUTER SCIENCE IN THE DEPARTMENT OF MATHEMATICS/COMPUTER SCIENCE, FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, NIGER STATE, NIGERIA.

CERTIFICATION

This project titled: **AN ONLINE-APPLICATION FOR INDUSTRIAL TRAINING** by **HAJARA JUMMAI MOHAMMED – IBRAHIM** meets the regulation governing the award of post graduate Diploma in Computer Science, Federal University of Technology, Minna and is approved for its contribution to knowledge and literary presentation.

.... Date......................

Mr. V. J. Akinola
Supervisor

.... Date......................

Dr. L. N. Ezeako
Head of Department

.... Date......................

Prof J. A. Abalaka
Dean, Postgraduate School

DEDICATION

To my one-in-a-million Hubby, Ibrahim Jibril Sheikh. Aint no me without you, Love.

You are the best.

Thanks for letting me be me, and loving me with a passion, just the way I am.

ACKNOWLEDGEMENT

This has certainly been a hectic but most exciting and fulfilling task. I must thank my God for all the inspiration and strength. THANK you Almighty ALLAH. You are my everything.

I owe immense gratitude to many people that in their different capacities aided the success of the work.

I thank my supervisor, Mr Victor Akinola. His words have encouraged and spurred me on. I deeply appreciate this.

I am also pleased to acknowledge the encouragement and guidance of the Head of Department Dr. L. N. Ezeako and other lecturers in the department.

I wish to thank my Husband Ibrahim Jibrin Sheikh for his support. I am also grateful to my brothers and sisters: Mohammed, Babangida, Abdullahi, Abdulsalami, aunty Safiya, Fatima Umar Kago, Rakiya and Hadiza all for your understanding and moral support.

I wish to thank my lovely parents Alhaji Mohammed and Hauwa Kulu for being there for me always.

My heartfelt thanks go to Alex Iyaji for your personal effort.

TABLE OF CONTENTS

CHAPTER ONE - INTRODUCTION

CHAPTER TWO - LITERATURE REVIEW

CHAPTER THREE - JAVA TECHNOLOGY FOR WEB APPLICATION DEVELOPMENT

CHAPTER FOUR - ON-LINE WEB APPLICATION

CHAPTER FIVE – CONCLUSION AND RECOMMENDATION

ABSTRACT

It is now an unmistakable fact that the level of success of any system or organization is dependent on how much of Information Technology is applied in the organization. Application of well articulated information technology to the operations of the Industrial Training Fund (ITF) by means of a client/server web application that connects to an Oracle database server will enable students, who are due for industrial training, to visit Internet cafes, log-on to the ITF website and apply for industrial training online by filling forms. These forms are analyzed at the other end by the ITF employees who in - turn assign placements to the students and communicate back to the students via e-mails.

CHAPTER ONE
INTRODUCTION

1.1 BACKGROUND TO THE STUDY

The rise of server-side Java applications is one of the latest and most exciting trends in Java programming. The Java language was originally intended for use in small, embedded devices. It was first hyped as a language for developing elaborate client-side web content in the form of applets. Until recently, Java's potential as a server-side development platform had been sadly overlooked. Now, Java is coming into its own as a language ideally suited for server-side development. Businesses in particular have been quick to recognize Java's potential on the server—Java is inherently suited for large client/server applications. The crossplatform nature of Java is extremely useful for organizations that have a heterogeneous collection of servers running various flavors of the Unix and Windows operating systems. Java's modern, object-oriented, memory-protected design allows developers to cut development cycles and increase reliability. In addition, Java's built-in support for networking and enterprise APIs provides access to legacy data, easing the transition from older client/server systems.

Java servlets are a key component of server-side Java development. A servlet is a small, pluggable extension to a server that enhances the server's functionality. Servlets allow developers to extend and customize any Java-enabled server—a web server, a mail server, an application server, or any custom server—with a hitherto unknown degree of portability, flexibility, and ease.

1.2 HISTORY OF THE INTERNET AND CLIENT/SERVER-SIDE WEB APPLICATIONS

In the beginning, the Internet was quite literally for rocket scientists. The Net began as a system of interconnected computers used by government and academic researchers in many parts of the world. By adhering to some standard ways of transferring data among varied computer systems, these researchers were able to trade raw data and to share computing resources.

When connected systems follow mutually agreed upon standards, the users know how to share data and computing resources among the systems.

Consider the following scenario, as an example of the sharing made possible by Internet standards.

> A Scientist at Minna University connects through his university computing center to a statistical analysis program that's installed on a computer at the Federal Ministry of Finance data center. After analyzing his data, the scientist sends the result to the computer account of a colleague at Zaria University.
>
> The Zaria colleague, in turn, can access the raw data at the Minna computing center and perform her own analysis with a program that she developed on the Zaria computer. Or she can use the Federal Ministry of Finance program.

Because they are connected through the Internet, the three computers (at Minna, Federal Ministry of Finance, and Zaria) work together. The Internet now connects most of the large public computing facilities in the world, and computing projects often use the combined resources of many computers in different locations.

The Internet is both a physical reality and an idea. The physical reality is that most large public computing centers in the world, such as university and government computing centers, are linked by a cross-connected web of data connections. Although each center may be directly connected to a few other centers, the interconnections enable any center to connect indirectly with any other center. The idea is that after a number of computers are connected; passing information from any one computer is easy, provided that users can agree on a single standard way to communicate. In the Internet community, all the participants have agreed to accept and forward messages by using mutually agreed-upon technical specifications and ways of addressing the messages.

At first, if you wanted an Internet address (the Internet equivalent of a telephone number), you had to claim some sort of connection with the government and the academic research community. But the attractions of the Internet were irresistible. College students discovered that being able to sent e-mail to friends on other campuses was very convenient. Professors discovered that they could publish a research proposal or a simple request for information to the Internet community and receive valuable help from unexpected sources.

As Internet users moved on to different places and other jobs, they found ways to maintain their Internet accounts. Eventually, the Internet was opened to the public. Now, millions of individuals in the general public have Internet accounts. The communications links to the Internet have extended and merged with telephone communications so that connecting any personal computer to the Internet is a simpler matter.

1.3 STATEMENT OF THE PROBLEM

With the advent of Internet Technology, the use of the Internet has been recognized by various organizations as a better means of communicating, processing and retrieving information for the day-to-day running of the organization. However, to properly utilize the Internet to achieve the desired organizational goals, the problems existing in the current process of seeking for Industrial Training placement in Nigeria should be identified and clearly stated. The identified problems have been categorized as follows:

i. Students encounter a lot of difficulties in communicating with organizations of their choice.

ii. If they secure placements in organizations far away from an ITF area office, they find it difficult to communicate immediately with the ITF area office for necessary documentations

iii. If they secure placements in organizations far away from their institutions, they find it difficult to communicate immediately with their institutions for necessary documentations.

1.4 OBJECTIVES OF THE STUDY

It is now an unmistakable fact that the level of success of any system or organization is dependent on how much of Information Technology is applied in the organization. Application of well articulated information technology to the operations of an organization is a sure way to its success.

This project is intended to make possible the application of information technology by developing a web application that connects to an Oracle database server for the Industrial Training Fund (ITF). This is done with the view to enabling students, who are due for industrial training to visit Internet cafes, log-on to the ITF website and apply for industrial training online by filling forms. These forms are analyzed at the other end by the ITF employees who in - turn assign placements to the students and communicate back to the students via e-mails.

1.5 JUSTIFICATION FOR THE STUDY

It is obvious that the current system of seeking for an industrial training placement is generally inefficient and tasking on the students. This also tells on the image of Nigeria's industrial training scheme. Developing an alternative method of seeking for an industrial training placement will enhance the image of the Nations industrial training scheme, while the problems inherent in the current process is eliminated. The proposed method will provide easier and more convenient application as well as faster feedbacks to the students, organizations and institutions.

1.6 SCOPE AND LIMITATIONS OF THE PROJECT

The scope of this project is restricted to developing a web application for the Industrial training Fund (ITF) that will enable students to apply for their Industrial training online. The Abuja area office is being considered as a case study. However, with some adjustment in the program source codes and the database files, it could be utilized to cover some other Internet applications.

1.7 DEFINITION OF TERMS

The terminologies used and applied in this project work will reflect mainly on web applications, Java technology, Oracle Relational Database Management System (RDMS), Java Database Connectivity (JDBC) and Java Naming Directory Interface (JNDI).

JAVA CLASS LIBRARIES

Java programs consist of pieces called classes. Classes consist of pieces called methods that perform tasks and return information when they complete their tasks. You can program each piece you may need to form a Java program. But most Java programmers take advantage of rich collections of existing classes in Java class libraries. The class libraries are also known as the Java APIs (Applications Programming Interfaces). Thus, there are really two pieces to learning the Java "world". The first is learning the Java language itself so that you can program your own classes and the second is learning how to use the classes in the extensive Java class libraries. Class libraries are provided primarily by compiler vendors, but many class libraries are supplied by independent software vendors. Also, many class libraries are available from the Internet and World Wide Web as shareware (products you can download for a small fee) and freeware (products you can download for free).

JAVA APPLICATION

An application is a program that executes using the java interpreter. An application is a program such as a word processor program, a spreadsheet program, a drawing program, an email program, etc. that is normally stored and executed from the user's local computer.

JAVA APPLET

Unlike a Java application that executes from a command window, an applet is a Java program that runs in the appletviewer (a test utility for applets that is included with the J2SDK) or a World Wide Web browser such as Netscape Communicator or Microsoft Internet Explorer. The appletviewer (or browser) executes an applet when a Hypertext Mark-up Language (HTML) document containing the applet is opened in the appletviewer (or browser). Hence applets are small programs that are normally stored on a remote computer that users connect to via a World Wide Web browser.

BASICS OF A TYPICAL JAVA ENVIRONMENT

Java systems generally consist of several parts: an environment, the language, the Java Applications Programming Interface (API), and various class libraries. An explanation of a typical Java program development environment is given below.

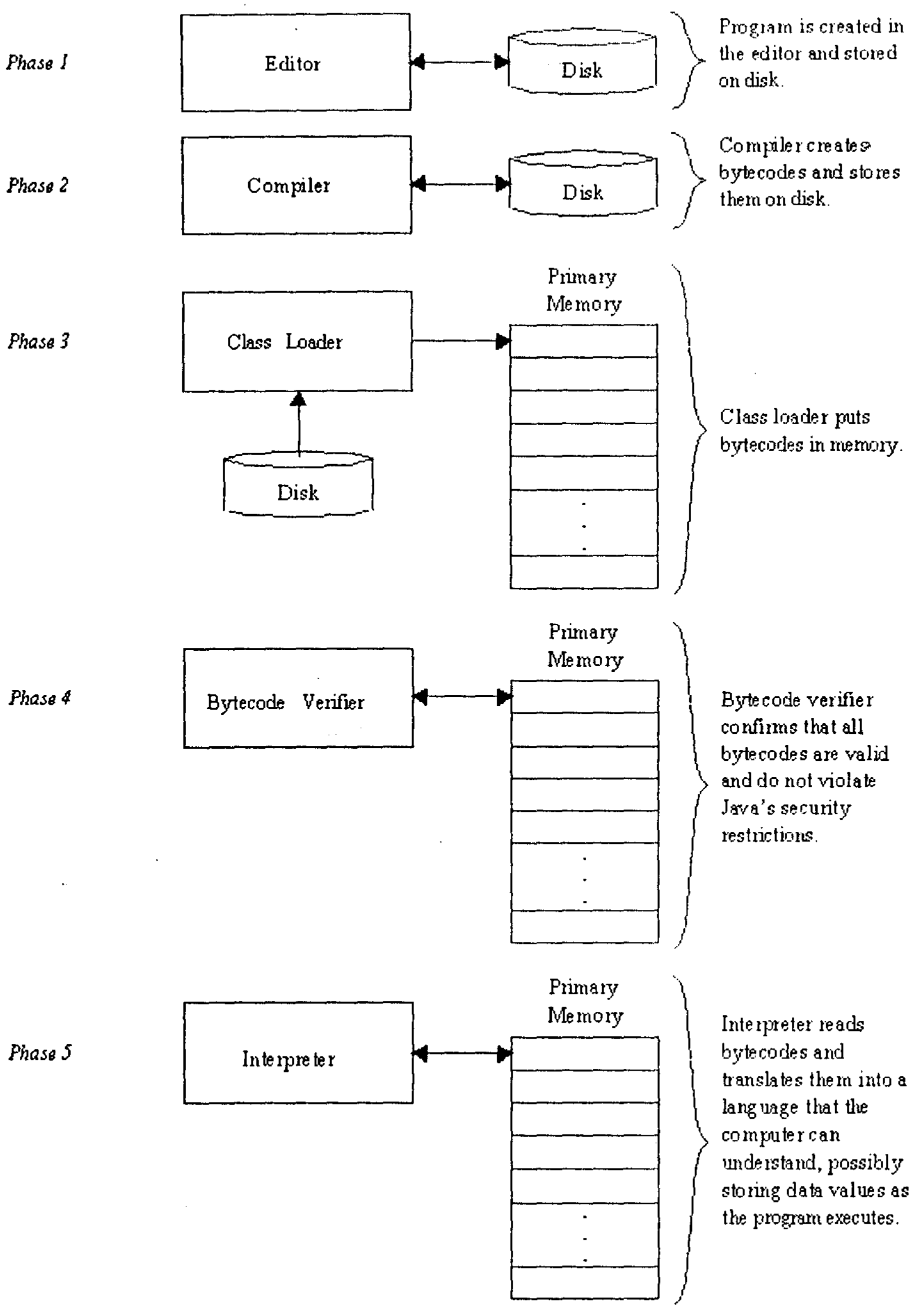

Phase 1
Editor
Disk
Program is created in the editor and stored on disk.
Phase 2
Compiler
Disk
Compiler creates bytecodes and stores them on disk.
Phase 3
Class Loader
Disk
Primary Memory
Class loader puts bytecodes in memory.
Phase 4
Bytecode Verifier
Primary Memory
Bytecode verifier confirms that all bytecodes are valid and do not violate Java's security restrictions.
Phase 5
Interpreter
Primary Memory
Interpreter reads bytecodes and translates them into a language that the computer can understand, possibly storing data values as the program executes.

Figure 1.1 – A typical Java environment.

Java programs normally go through five phases to be executed (Figure 1.1). These are: **edit, compile, load, verify** and **execute**.

Phase 1 consists of editing a file. This is accomplished with an editor program. The programmer types a Java program using the editor and makes corrections if necessary. When the programmer specifies that the file in the editor should be saved, the program is stored on a secondary storage device such as a disk. Java program file names end with the .java extension. Java integrated development environments (IDEs) such as Borland's JBuilder, Eclipse, and NetBeans have built-in editors that are smoothly integrated into the programming environment.

In **Phase 2**, the programmer gives the command **javac** to *compile* the program. The Java compiler translates the Java program into ***bytecodes*** the language understood by the Java interpreter.

If the program compiles correctly, a file with class extension is produced. This is the file containing the bytecodes that will be interpreted during the execution phase.

Phase 3 is called *loading*. The program must first be placed in memory before it can be executed. This is done by the class loader, which takes the class file (or files) containing the bytecodes and transfers it to memory. The .class file can be loaded from a disk on your system or over a network (such as your local university or company network or even the Internet). There are two types of programs for which the class loader loads **.class** files – applications and applets.

Applets are loaded from a remote computer into the browser, executed in the browser and discarded when execution completes. To execute an applet again, the user must point their browser at the appropriate location on the World Wide Web and reload the program into the browser.

Applications are loaded into memory and executed using the Java interpreter via the command **java**.

The class loader also is executed when a Java applet is loaded into a World Wide Web browser such as Netscape's Communicator or Microsoft's Internet Explorer. Browsers are used to view documents on the World Wide Web called HTML (Hypertext Mark-up Language) documents. HTML is used to format a document in a manner that is easily understood by the browser application. An HTML document may refer to a Java applet.

When the browser sees an applet referenced in an HTML document, the browser launches the Java class loader to load the applet (normally from the location where the HTML document is stored). Browsers that support Java each have a built-in Java interpreter. Once the applet is loaded, the browser's Java interpreter executes the applet.

Before the bytecodes in an applet are executed by the Java interpreter built into a browser, they are verified by the *bytecode verifier* in **Phase 4** (this also happens in applications that download bytecodes from a network). This ensures that the bytecodes for classes that are loaded from the Internet (referred to as downloaded classes) are valid and that they do not violate Java's security restrictions. Java enforces strong security because Java programs arriving over the network should not be able to cause damage to your files and your system (as computer viruses might).

Finally, in **Phase 5**, the computer, under the control of its CPU, interprets the program one bytecode at a time, thus performing the actions specified by the program.

Programs may not work on the first try. Each of the preceding phases can fail because of various errors. For example, an executing program might attempt to divide by zero (an illegal operation in Java just as it is in arithmetic). This would cause the Java program to print an error message. The programmer would return to the edit phase, make the necessary corrections and proceed through the remaining phases again to determine if the corrections work properly.

CHAPTER TWO
LITERATURE REVIEW

2.1 Concept of Online Applications

Online applications are programmes capable of communicating with or being controlled by a computer. For example, a printer is online when it can be used for printing; a database is online when it can be used by a person who connects with the computer on which it is stored. In the early 1990s online applications made it possible for discussion groups to share information and discuss ideas in topics ranging from music and art to sports and politics. Today, millions of people who are involved in online e-commerce can visit World Wide Web sites to buy books or CDs, order flowers, or check their bank accounts. Banking is becoming a lot easier with the implementation of online applications designed for banking. International-Broadcasting Corporations like the British Broadcasting Corporation (BBC), Produces radio programmes for broadcast around the world in 43 languages, including English, all of which are also available online. [See 3].

Today in Nigeria, a lot of online application packages have been designed and deployed over the Internet for the purpose of cutting cost and ensuring efficiency. Some of these applications have been designed to carter for the W.A.E.C, N.E.C.O, and JAMB exams, some have been designed to aid job seekers in their quest of gaining employment. It is intended that at the end of this project, an online package that will aid students seeking for industrial training placement would be made available.

2.1.1 Tools for Online Applications

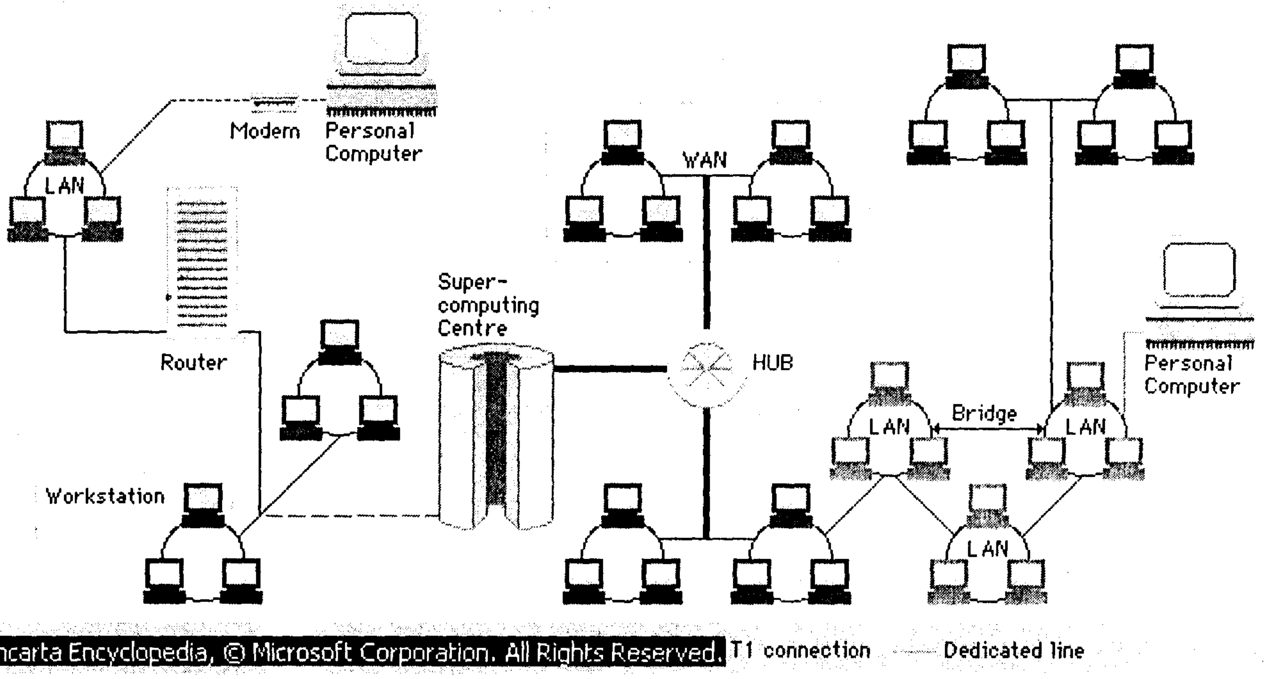
The tools for online applications include the following: Internet, World Wide Web, Electronic mails, Networking, Client and Server.

a. Internet

The Internet is a collection of computer networks that operate to common standards and enable the computers and the programs they run to communicate directly. There are many small-scale, controlled-access "enterprise Internets", but the term is usually applied to the global, publicly accessible network, called simply the Internet or Net. Vinton Cerf created the Internet technology in early 1973 as part of a project headed by Robert Kahn and conducted by the Advanced Research Projects Agency, part of the United States Department of Defense. Thereafter, Cerf led many efforts to build, scale, and standardize the Internet. In

84 the technology and the network were turned over to the private sector and to vernment scientific agencies for further development. The growth has continued ponentially. Service-provider companies that make "gateways" to the Internet available to me and business users enter the market in ever-increasing numbers. By early 2000, access as available in over 200 countries and encompassed around 100 million users. The Internet d its technology continue to have a profound effect in promoting the sharing of formation, making possible rapid transactions among businesses, and supporting global llaboration among individuals and organizations.

METHODS OF CONNECTING



ncarta Encyclopedia, © Microsoft Corporation. All Rights Reserved.

ere are four ways to connect to the public Internet.

i) **Host access** is usually carried out via dial-up telephone lines and modems, combined with Internet software on a personal computer, and allows the computer that is accessed to function fully as an Internet host.

ii) **Network access** is similar to host access, but is done via a leased line or an "always-on" link such as Digital Subscriber Line (DSL) or Ether loop. In this case, all the attached computers are made into Internet hosts.

iii) **Terminal access** is usually carried out via dial-up telephone lines and modems combined with terminal emulation software on a personal computer; it allows interaction with another computer that is an Internet host.

iv) **Gateway access** is similar to terminal access, but is provided via on-line or similar proprietary services that give the user the ability to exchange e-mail with the Internet.

b. World Wide Web (WWW)

The World Wide Web was developed in 1989 by Timothy Berners-Lee, a British computer scientist at the European Center for Nuclear Research (CERN) facility near Geneva, Switzerland, to allow information-sharing among internationally dispersed teams of high-energy physics researchers. It subsequently became a platform for related software development, and the numbers of linked computers and users grew very rapidly to support a wide variety of endeavours, including a large business marketplace. Its further development is guided by the WWW Consortium based at Massachusetts Institute of Technology. The World Wide Web allows the seamless creation and use of elegant point-and-click hypermedia presentations, linked across the Internet in a way that creates a vast open knowledge repository, through which users can easily browse. It is a library of resources available to computer users through the global Internet. It enables users to view a wide variety of information, including magazine archives, public and college library resources, and current world and business news.

c. Electronic Mail (E-Mail)

E-mails are electronic messages transmitted between computers via a network. Millions of e-mails are exchanged every day—virtually every business relies on it and many people see it as their preferred method of communication. For all its ubiquity and importance e-mail is very simple and has humble roots. An e-mail message is nothing more than a piece of text and the sending of a mail message requires no more than the simple attachment of this text to a special file, known as a mailbox. The first e-mail message was sent in 1971 by an engineer named Ray Tomlinson, one of the pioneers of the Internet. He showed how a messaging facility that could be used by several users on a single computer could be extended so that it can work between a number of computers. Tomlinson decided that the @ sign should be used to designate the receiving machine, and so e-mail as we know it was born. Practical e-mail systems have only a few, simple components. At the user end is a piece of software known as an "**e-mail client**"; at the other end from the client is the **e-mail server**.

2.2 Concept of Training

Different people in circumstances occasioned by organizational needs and goals have defined training variously. Training could mean, "teaching operational or technical employees how to do the job which they are hired. An online dictionary defines training "as the act of passing on Knowledge freely to employees with the hope that they can independently utilize, convert, renew and extend it". A Military journal defines training "as the systematic development of the knowledge, skills, and attitudes required by an individual to perform adequately a given task or job". Training involves various learning process in various situations. The basic premise of training is that it concerns people already employed and whose jobs are sufficiently defined and whose job behaviours can be identified. It is primarily concerned with the steps taking by employers to develop the potentials of employees to ensure maximum performance for the benefit of the organization.

The availability of a corps of well-trained workforce is a prerequisite for the attainment of enhanced efficiency in the implementation of policies and programmes in any organization. It is through training that people acquire knowledge and/or skill, for a definite purpose. The objective of training is to achieve a change in the behaviour of those trained. In the industrial situation, this means that the trainees shall acquire new manipulated skills, technical knowledge, and problem solving ability or attitudes. It must be noted that a lot of knowledge could be impacted through informal training by a supervisor in the field. On the other hand, a new entrant into an organization can learn a lot through observations of superiors and colleagues.

2.3 Industrial Training Overview

"Training is the acquisition and development of knowledge to achieve a set level of proficiency". In concept and practice, it is the enhancement of efficiency and effectiveness in any system. In the industry, it remains the cornerstone for building a strong and virile organization. Industries offer training that would ensure the proper utilisation of synergies for creating common aids and standards. Industrial training according to the British government is "The preparation of students to effectively serve their fatherland". American Industrial Training News defines industrial training as "a process that should result in an increased skill level for an individual or group ability. It is therefore a positive process aimed at building confidence through learning".

Bill Gates, the founder of Microsoft Corporation states that training in the industry "welds competent individuals into effective units and organizations and make possible the creation of an efficient and competent work force to meet public demands". Indeed, training in the industry is a continuous process aimed at making the individual and hence the organization always ready through acquisition of skill and proficiency on characteristics and tactics of available hardware's, software's and other equipments. Industrial training therefore is a pre-planned and well-organized activity and success in a country's economic advancement amongst other factors depends on it. Michael sums up the overall objective of industrial training as "the achievement of tactical superiority over competitors through adequate knowledge of equipments, the effective coordination of their use and the ability to appreciate the working environment".

Industrial training is provided to the employees in the industry, students on attachment, managerial staff who wish to acquire competences to enable them perform the leadership and managerial tasks required at their functional level, technical staff who require knowledge and expertise to perform their functions along the entire value-creation chain, company managers and decision makers who have to evaluate the technologies to be used and specify the functional requirements of the systems and products, operators who monitor the equipment when it is in operation and have to respond to its displays and messages, and various categories of customers who require information and knowledge to fulfil all the functions of their professional lives. This training helps the employees and students through the stages leading up to and including the operation and maintenance of the equipments and facilities in the industry. It helps develop their technical competence and provides them with the ability to handle technical knowledge, processes and problems.

2.4 Industrial Training in Nigeria

The training concept of industries in Nigeria is not common, that is, each industry structures its training as it wishes. Each industry operates as an independent training field, and is responsible for the customer satisfaction and profitability of the training it offers. Industrial Training takes account of all the areas of competence of professional life. Due to the rapid obsolescence of technical knowledge, ever-changing and increasingly complex work structures and processes, as well as modified human values and self-images. Education and training must cover not only methodological and technical competences but also those of a behavioural nature. (Competence = Capacity to act). For this reason the government of Nigeria in 1971 under the leadership of General Yakubu Gowan decided to enact Decree No. 47 which saw the setting up of the Industrial Training Fund (ITF) with headquarters at Jos, the Plateau state capital, with the hope of building capacity for industries and commerce in the Nigerian economy through manpower training.

2.4.1 Establishment of Industrial Training Fund (ITF)

The Industrial Training Fund (ITF) was established in 1971 amidst the nation-wide climate of accelerated economic activities of the early 1970s, as a Manpower Development Agency of the Federal Government of Nigeria, specifically to provide services and facilities for the development of skilled manpower, particularly, in the industrial and commercial sectors of the economy.

The ITF's enabling Decree No. 47 of 8th October, 1971 was, therefore, one of the major leverages by which the government hoped to transform the economy of the nation from its predominant dependence on foreign expertise to a state of self-reliance, through the training and development of Nigerians that would be competent to perform the specialised skills required to manage essential sectors of the national economy.

2.4.1.1 Mission Statement of the ITF

The provision of Decree 47 of 8th October 1971 empowers the ITF to "promote and encourage the acquisition of skills in industry and commerce with a view to generating a pool of indigenous trained manpower sufficient to meet the needs of the Nigerian economy".

2.4.1.2 Organogram and Branch Network of ITF

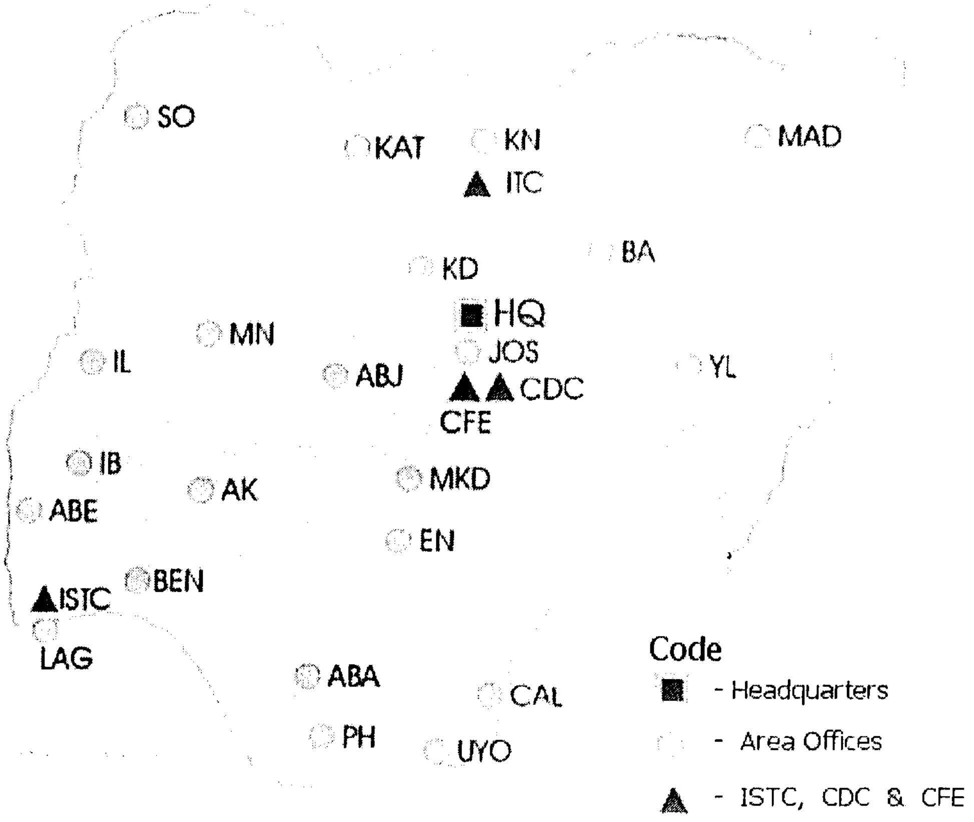
The Industrial Training Fund is a “Group A” parastatal. As prescribed by statute, the ITF operates under the aegis of the Federal Ministry of Industry. The ITF, therefore, receives ministerial directives and guidance from time to time on training and development matters through the Governing Council and reports to the ministry the progress attained in the accomplishment of its activities and services, through the Governing Council.

The ITF currently operates in accordance with the following structure: -

- The Governing council
- The Directorate; (i.e. office of the Director General and Chief Executive)
- Eight Headquarters Departments each headed by a Director
- Twenty-six Area Offices (nation-wide);
- Three Training Centres; viz:-
    - Industrial Skills Training Centre, Ikeja
    - Industrial Skills Training Centre, Kano
    - Centre for Excellence, Bukuru, Jos

2.4.1.3 Branch Network

BRANCH NETWORK OF ITF



ITF is Headquartered at Jos and has 26 area offices in the following cities: Abuja, Aba, Abeokuta, Akure, Bauchi, Benin, Calabar, Enugu, Ilorin, Apapa, Lagos Island, Ibadan, Jos, Kaduna, Kano, Katsina, Ikeja, Lokoja, Makurdi, Maiduguri, Minna, Port-Harcourt, Uyo, Sokoto, Yola, Awka.

2.4.1.4 Students Industrial Work Experience Scheme (SIWES).

The **SIWES** was established by the **ITF** in 1973 to solve the problem of lack of adequate practical skills preparatory for employment in industries by Nigerian graduates of tertiary institutions. The Scheme exposes students to industry-based skills necessary for a smooth transition from the classroom to the world of work. It affords students of tertiary institutions the opportunity of being familiarized and exposed to the needed experience in handling machinery and equipment, which are usually not available in the educational institutions. Participation in **SIWES** has become a necessary pre-condition for the award of Diploma and

Degree certificates in specific disciplines in most institutions of higher learning in the country, in accordance with the education policy of government.

- **Operators** – The operators of SIWES are the ITF, the coordinating agencies (NUC, NBTE), employers of labour and the institutions.
- **Funding** - The Federal Government of Nigeria funds the SIWES programme through the ITF.
- **Beneficiaries** – Beneficiaries of SIWES are the undergraduate students of the following disciplines: Agriculture, Engineering, Technology, Environmental, Science, Education, Medical Science and Pure and Applied Sciences.
- **Duration** - Four months for Colleges of Education, one year for Polytechnics and Six months for the Universities.
- Highlight Number of Participating Institutions:

Universities	34
Polytechnics	65
Colleges of Education	59
Total	**158**

Average number of students scheduled to participate in SIWES from Universities, Polytechnics, and Colleges of Education annually is about 78,000.

CHAPTER THREE

JAVA TECHNOLOGY FOR WEB APPLICATION DEVELOPMENT

3.1 Java Servlets

A Java servlet is a generic server extension—a Java class that can be loaded dynamically to expand the functionality of a server. Servlets are commonly used with web servers. A servlet is similar to a proprietary server extension, except that it runs inside a Java Virtual Machine (JVM) on the server (see Figure 3.2), so it is safe and portable. Servlets operate solely within the domain of the server: unlike applets, they do not require support for Java in the web browser.

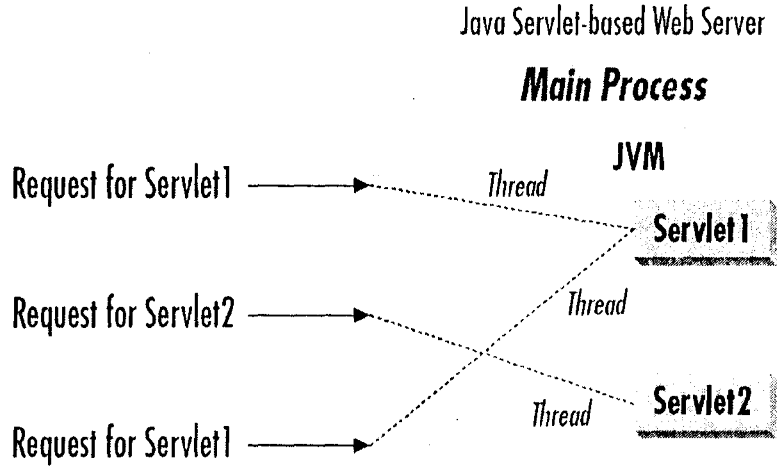


Figure 3.2 The servlet life cycle

Another advantage of servlets is that they are portable: both across operating systems with Java and also across web servers. All of the major web servers support servlets. Java servlets offer the best possible platform for web application development.

3.1.1 The Power of Servlets

Servlets are portrayed as an alternative to other dynamic web content technologies. They are a viable choice for web development and offer a number of advantages over other approaches, including: portability, power, efficiency, endurance, safety, elegance, integration, extensibility, and flexibility.

i. Portability

Because servlets are written in Java and conform to a well-defined and widely accepted API, they are highly portable across operating systems and across server implementations. One can develop a servlet on a Windows NT machine running the Java Web Server and later deploy it effortlessly on a high-end Unix server running Apache. With servlets, one can truly "write once, and serve everywhere." Servlet portability is not the stumbling block it so often is with applets, for two reasons. First, servlet portability is not mandatory. Unlike applets, which have to be tested on all possible client platforms, servlets have to work only on the server machines that one is using for development and deployment.

ii. Power

Servlets can harness the full power of the core Java APIs: networking and URL access, multithreading, image manipulation, data compression, database connectivity, internationalization, remote method invocation (RMI), CORBA connectivity, and object serialization, among others. If one wants to write a web application that allows employees to query a corporate legacy database, one can take advantage of all of the Java Enterprise APIs in doing so. Or, if you need to create a web-based directory lookup application, one can make use of the JNDI API. As a servlet author, one can also pick and choose from a plethora of third-party Java classes and JavaBeans components. In the future, one may even be able to use newly introduced Enterprise JavaBeans components.

iii. Efficiency and Endurance

Servlet invocation is highly efficient. Once a servlet is loaded, it generally remains in the server's memory as a single object instance. Thereafter, the server invokes the servlet to handle a request using a simple, lightweight method invocation. Multiple, concurrent requests are handled by separate threads, so servlets are highly scalable. Servlets, in general, are naturally enduring objects. Because a servlet stays in the server's memory as a single object instance, it automatically maintains its state and can hold on to external resources, such as database connections, that may otherwise take several seconds to establish.

iv. Safety

Servlets support safe programming practices on a number of levels. Because they are written in Java, servlets inherit the strong type safety of the Java language. In addition, the Servlet API is implemented to be type-safe. While most values in a CGI program, including a numeric item like a server port number, are treated as strings, values are manipulated by the Servlet API using their native types, so a server port number is represented as an integer.

v. Elegance

The elegance of servlet code is striking. Servlet code is clean, object oriented, modular, and amazingly simple. One reason for this simplicity is the Servlet API itself, which includes methods and classes to handle many of the routine chores of servlet development. Even advanced operations, like cookie handling and session tracking, are abstracted into convenient classes.

vi. Integration

Servlets are tightly integrated with the server. For example, a servlet can use the server to translate file paths, perform logging, check authorization, perform MIME type mapping, and, in some cases, even add users to the server's user database. Server-specific extensions can do much of this, but the process is usually much more complex and error-prone.

vii. Extensibility and Flexibility

The Servlet API is designed to be easily extensible. As it stands today, the API includes classes that are optimized for HTTP servlets. But at a later date, it could be extended and optimized for another type of servlets, either by Sun Microsystems or by a third party. It is also possible that its support for HTTP servlets could be further enhanced. Servlets are also quite flexible. An HTTP servlet can be used to generate a complete web page; it can be added to a static page using a <SERVLET> tag in what's known as a server-side include; and it can be used in cooperation with any number of other servlets to filter content in something called a servlet chain. In addition, Sun Microsystems introduced Java Server Pages, which offer a way to write snippets of servlet code directly within a static HTML page, using a syntax that is curiously similar to Microsoft's Active Server Pages (ASP).

3.2 Java Server Pages

Sun Microsystems recently announced a new way to use servlets, called Java Server Pages (commonly referred to as JSP). JSP's functionality and syntax bear a remarkable resemblance to Active Server Pages (ASP). JSP operates in many ways like server-side includes. The main difference is that instead of embedding a <SERVLET> tag in an HTML page, JSP embeds actual snippets of servlet code. It's an attempt by Sun to separate content from presentation, more convenient than server-side includes for pages that have chunks of dynamic content intermingled with static content in several different places. Just like server-side includes and servlet chaining, JSP doesn't require any changes to the Servlet API. But it does require special support in your web server.

3.2.1 Using Java Server Pages

At its most basic, JSP allows for the direct insertion of servlet code into an otherwise static HTML file. Each block of servlet code (called a *scriptlet*) is surrounded by a leading <% tag and a closing %> tag.

If one has a server that supports Java Server Pages and want to test this page; the file should be placed under the server's document root (probably *server_root/public_html*) and save it with a special extension. By default, this extension for JSP pages is *.jsp*. The first time you access a JSP page, one may notice that it takes a short time to respond. This is the time necessary for the server to create and compile the background servlet.

3.2.2 Java Server Pages and Java Beans

One of the most interesting and powerful ways to use Java Server Pages is in cooperation with Java Beans components. Java Beans are reusable Java classes whose methods and variables follow specific naming conventions to give them added abilities. They can be embedded directly in a JSP page using <BEAN> tags. A Java Bean component can perform a well-defined task (execute database queries, connect to a mail server, maintain information about the client, etc.) and make its resulting information available to the JSP page through simple access or methods. The difference between a Java Beans component embedded in a JSP page and a normal third-party class used by the generated servlet is that the web server can give Java Beans special treatment.

3.3 MACROMEDIA DREAMWEAVER MX 2004

Macromedia Dreamweaver MX 2004 is a professional HTML editor for designing, coding, and developing websites, web pages, and web applications. Whether one enjoys the control of hand-coding HTML or prefer to work in a visual editing environment, Dreamweaver provides helpful tools to enhance web creation experience. The visual editing features in Dreamweaver let one quickly create pages without writing a line of code. It makes it possible to view all site elements or assets and drag them from an easy-to-use panel directly into a document. One can streamline development workflow by creating and editing images in Macromedia Fireworks or another graphics application, then importing them directly into Dreamweaver, or by adding Macromedia Flash objects.

3.4 Relational Database

The concept of relational database is now new: it was originally developed back in 1970 by Dr. E. F. Codd, and he laid down the theory of relational databases in his seminal paper entitled "A Relational Model of Data for Large Shared Data Banks" published in *Communications of the ACM* (Association for Computing Machinery). Vol. 13, No. 6, june 1970.

The basic concepts of a relational database are fairly easy to understand. A *relational database* is a collection of related information that has been organized into structures known as *tables*. Each table contains *rows* that are further organized into *columns*. These tables are stored in the database in structures known as *schemas*, which are areas where database users may store their tables.

In the case of a database consisting of pieces of paper, the database management system might be a set of alphabetically indexed cards in a filing cabinet; for a database accessed using a computer, the database management system is the software that manages the files stored in the file system of the computer. The Oracle database is one such piece of software; other examples include SQL Server, DB2, and MySQL.

3.5 Structured Query Language (SQL)

The structured Query Language (SQL) is the standard language designed to access relational databases. SQL is pronounced either as the word "sequel" or as the letters "ess-que-ell." SQL is based on the groundbreaking work of Dr. E. F. Codd, with the first implementation of SQL being developed by IBM in the mid-1970s. Later in 1979, a company known as Relational Software Inc. (known today as Oracle Corporation) released the first commercial version of SQL. SQL is now fully standardized and recognized by the American National Standard Institute (ANSI). SQL can be used to access an Oracle database, SQL Server, DB2, and MySQL, to name just a few of the many database systems that support SQL.

The syntax used by SQL is quite easy to learn and master. There are several types of SQL statements; the two most common used types are:

- Data Manipulation Language (DML) statements
- Data Definition Language (DDL) statements

A DML statement allows one to retrieve, add, modify, and delete rows stored in tables. DDL statements are used to create database users, tables, and many other types of database structures that make up a database.

3.6 Common Oracle Database Types

While there are many types that may be used to handle data in an Oracle database, the most commonly used types are as follows:

- **CHAR(length)** Stores strings of a fixed length. The length parameter specifies the length of the string. If a string of a smaller length is stored, then it is padded with spaces at the end.
- **VARCHAR2(length)** Stores strings of a variable length. The length parameter specifies the maximum length of the string.
- **DATE** Stores dates and times. It stores all four digits of a year, along with the month, the day, the hour (in 24-hour format), the minute, and the second.
- **INTEGER** Stores numbers. An integer number doesn't contain a floating point: it is a whole number, such as 1, 10, and 115, for example.

- □ **NUMBER(precision, scale)** Stores floating point numbers but may also be used to store integer numbers. *Precision* is the maximum number of digits (in front of and behind of a decimal point, if used) that may be used for the number.

3.7 The Oracle JDBC Drivers

JDBC stands for *Java Database Connectivity*. It is through a JDBC driver that your JDBC statements access a database. There are four Oracle JDBC drivers:

- □ Thin driver
- □ OCI driver
- □ Server-side internal driver
- □ Server-side Thin driver

3.7.1 The Thin Driver

The Thin driver has the smallest footprint of all the drivers, meaning that it requires the least amount of system resources to run, and it is written entirely in Java. Thin drivers are used for written Java applets.

3.7.2 The OCI Driver

The OCI drivers require more resources than the Thin driver but generally has better performance than the Thin driver. The OCI driver is suitable for programs deployed on the middle tier – a web server, for example.

3.7.3 The Server-Side Internal Driver

The server-side internal driver provides direct access to the database, and is used by the Oracle JVM to communicate with that database. The Oracle JVM is a Java Virtual Machine that is integrated with the database – users can load a Java class into the database, and then publish and run methods contained in that class using the Oracle JVM.

3.7.4 The Server-Side Thin Driver

The server-side Thin driver is also used by the Oracle JVM, and provides access to remote databases. Like the Thin driver, this driver is also written entirely in Java.

3.7.5 Importing the JDBC Packages

In order to use JDBC in programs, users must import the required JDBC packages into their Java programs. There are two sets of JDBC packages: the standard JDBC packages enable Java programs to access the basic features of most databases, including the Oracle database, SQL Server, DB2, and MySQL. The Oracle extensions to JDBC enable programs to access all of the Oracle specific features, as well as the Oracle specific performance extensions.

3.7.6 Registering the Oracle JDBC Drivers

Before opening a database connection, the Oracle JDBC drivers with the java program must be registered. The JDBC drivers are the software "glue" that allows JDBC statements to actually access the database. There are two ways to register the Oracle JDBC drivers: the first is to use forName () method of the class java.lang.class, and the second is to use the registerDriver () method of the JDBC DriverManager class.

3.7.7 Opening a Database Connection

Before performing any SQL statements in Java programs, a database connection must first be opened. There are two main ways to open a database connection. The first way is to use the getConnection () method of the DriverManager class. The second way uses an Oracle data source object, which must first be created and then connected to. Using an Oracle data source employs a standardized way of setting database connection details, and an Oracle data source object may be used with the *Java Naming and Directory Interface* (JNDI).

3.7.8 Creating a JDBC Statement Object

Before issuing SQL statements using JDBC, a JDBC Statement object of the class java.sql.Statement needs to be created. A Statement object is used to represent a SQL statement, such as a DML statement (SELECT, INSERT, UPDATE, or DELETE) or a DDL statement (such as CREATE TABLE). The createStatement () method of a connection object is used to create a statement object.

The method in the statement class to run the SQL depends on the SQL statement to be performed. To perform a SELECT statement, the executeQuery () method is used. To perform an INSERT, UPDATE or DELETE statement, the executeUpdate () method is used. The execute () method can also be used to perform DDL statements.

3.7.9 Retrieving Rows from the Database

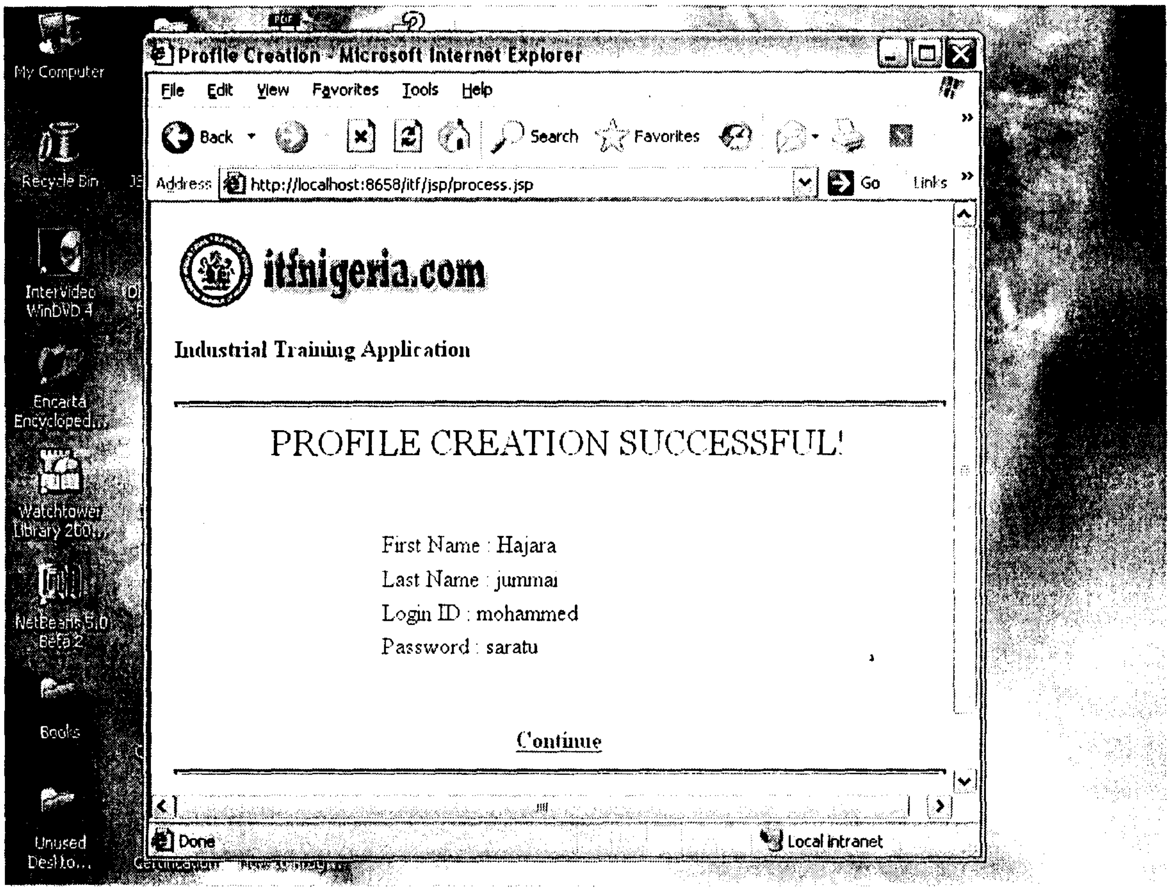
The SQL SELECT statement is used to retrieve informatiom from the database. To perfoprm a SELECT statement using JDBC, the executeQuery () method of the statement object can be used, because it accepts a Java string containing the text for the SELECT statement. Because a SELECT statement may return more than one row, the executeQuery () method returns an object that stores the row(s) returned by the SELECT statement. This object is known as a JDBC result set and is of the java.sql.ResultSet class. When using a ResultSet object to read rows from the database, there are three steps to be followed:

1. Create a ResultSet object and populate it using a SELECT statement.
2. Read the column values from the ResultSet object using get methods.
3. Close the ResultSet object.

CHAPTER FOUR

ON-LINE WEB APPLICATION

4.1 STUDENT'S PROFILE CREATION

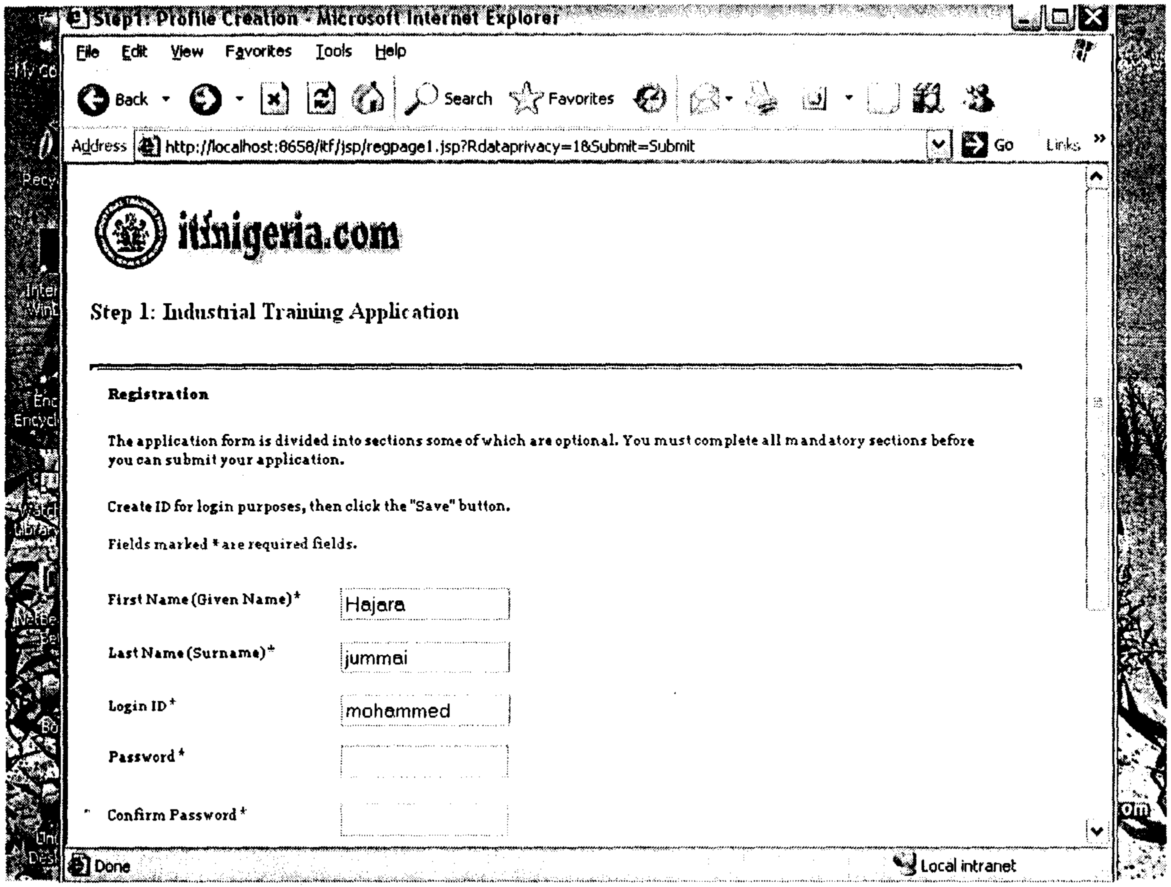


Profile Creation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8658/itf/jsp/process.jsp
Go Links
itfnigeria.com
Industrial Training Application
Continue
Done
Local intranet
My Computer
Recycle Bin
InterVideo WinDVD 4
Encarta Encycloped...
Watchtower Library 200...
NetBeans 5.0 Beta 2
Books
Unused Deskto...

PROFILE CREATION SUCCESSFUL!

First Name : Hajara
Last Name : jummai
Login ID : mohammed
Password : saratu

4.2 REGISTRATION PAGE


Step1: Profile Creation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8658/itf/jsp/regpage1.jsp?Rdataprivacy=1&Submit=Submit
Go Links
Done
Local intranet

itfnigeria.com

Step 1: Industrial Training Application

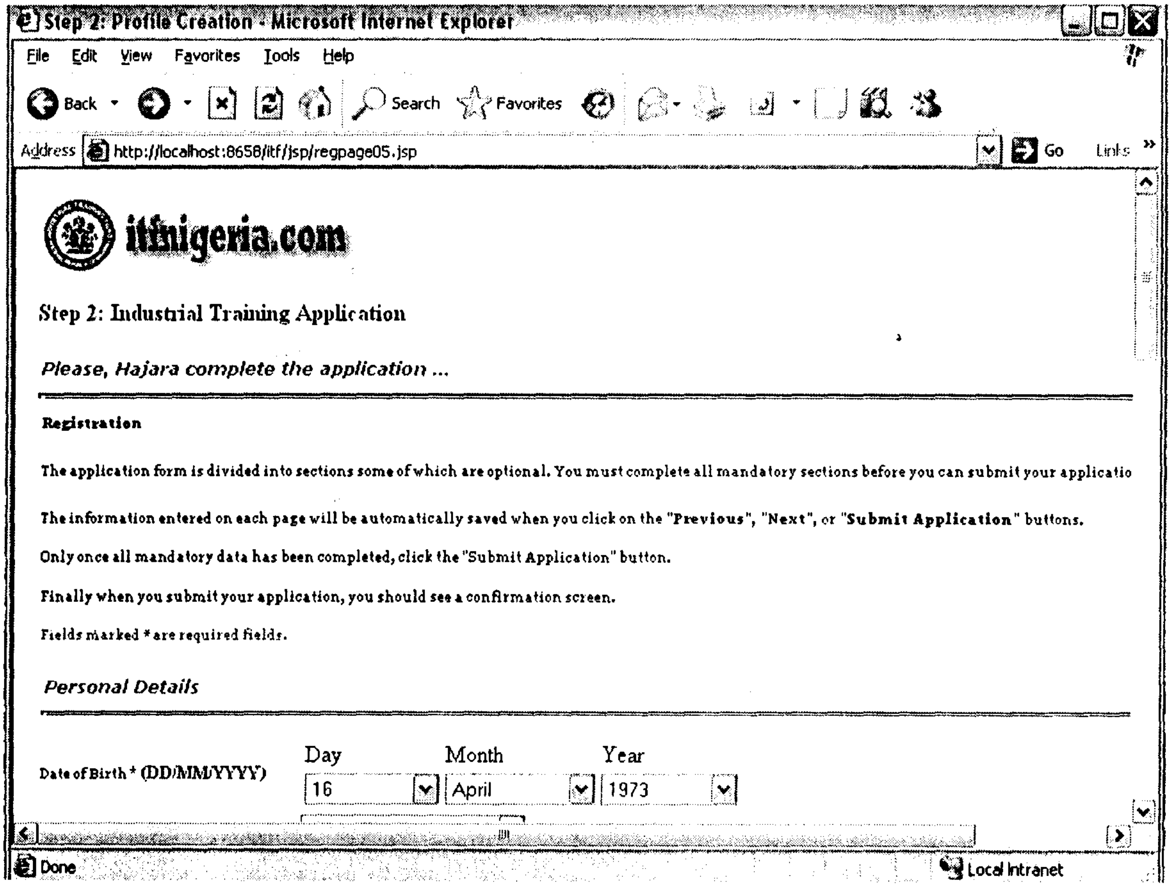
Registration

The application form is divided into sections some of which are optional. You must complete all mandatory sections before you can submit your application.

Create ID for login purposes, then click the "Save" button.

Fields marked * are required fields.

First Name (Given Name)*	Hajara
Last Name (Surname)*	jummai
Login ID*	mohammed
Password*	
Confirm Password*	


Step 2: Profile Creation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8658/itf/jsp/regpage05.jsp
Go Links
Done
Local intranet

itfnigeria.com

Step 2: Industrial Training Application

Please, Hajara complete the application ...

Registration

The application form is divided into sections some of which are optional. You must complete all mandatory sections before you can submit your applicatio

The information entered on each page will be automatically saved when you click on the "**Previous**", "**Next**", or "**Submit Application**" buttons.

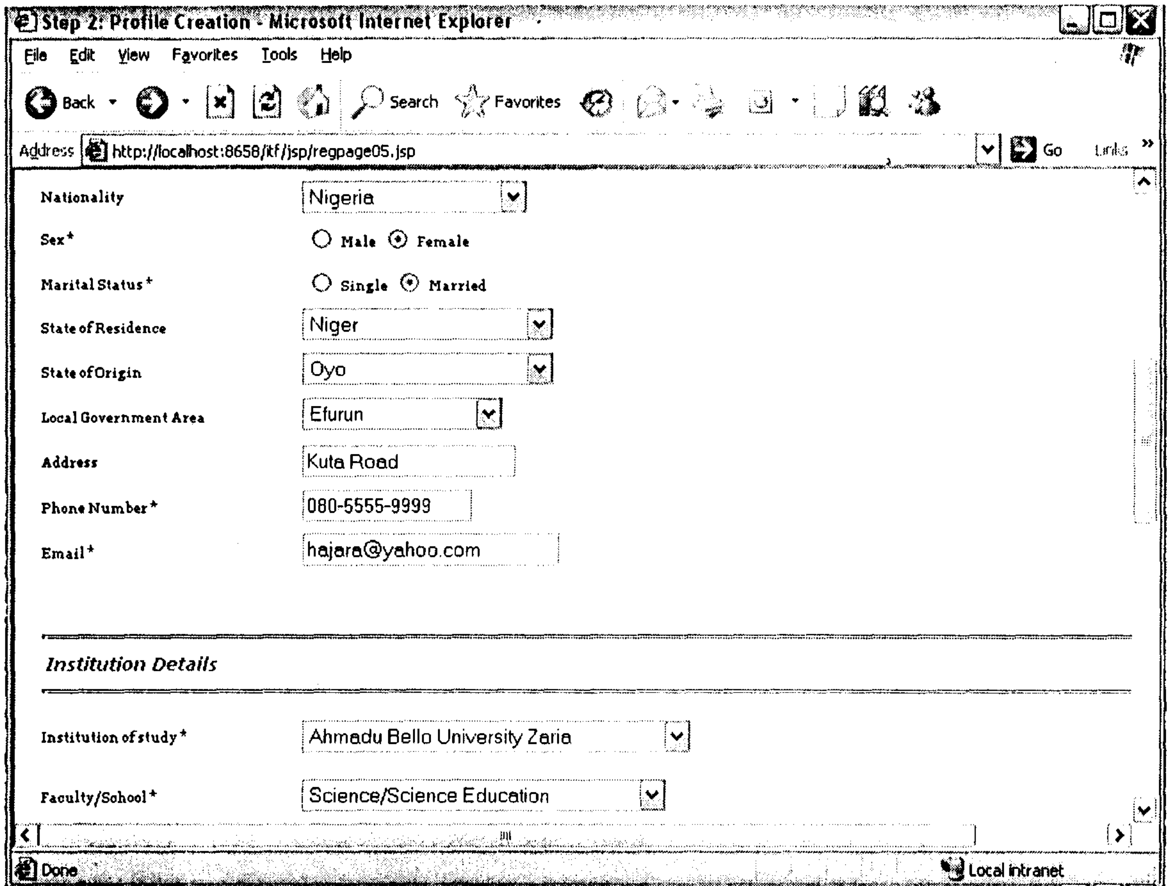
Only once all mandatory data has been completed, click the "Submit Application" button.

Finally when you submit your application, you should see a confirmation screen.

Fields marked * are required fields.

Personal Details

	Day	Month	Year
Date of Birth* (DD/MM/YYYY)	16	April	1973



http://localhost:8658/itf/jsp/regpage05.jsp

Nationality: Nigeria

Sex*: ○ Male ● Female

Marital Status*: ○ Single ● Married

State of Residence: Niger

State of Origin: Oyo

Local Government Area: Efurun

Address: Kuta Road

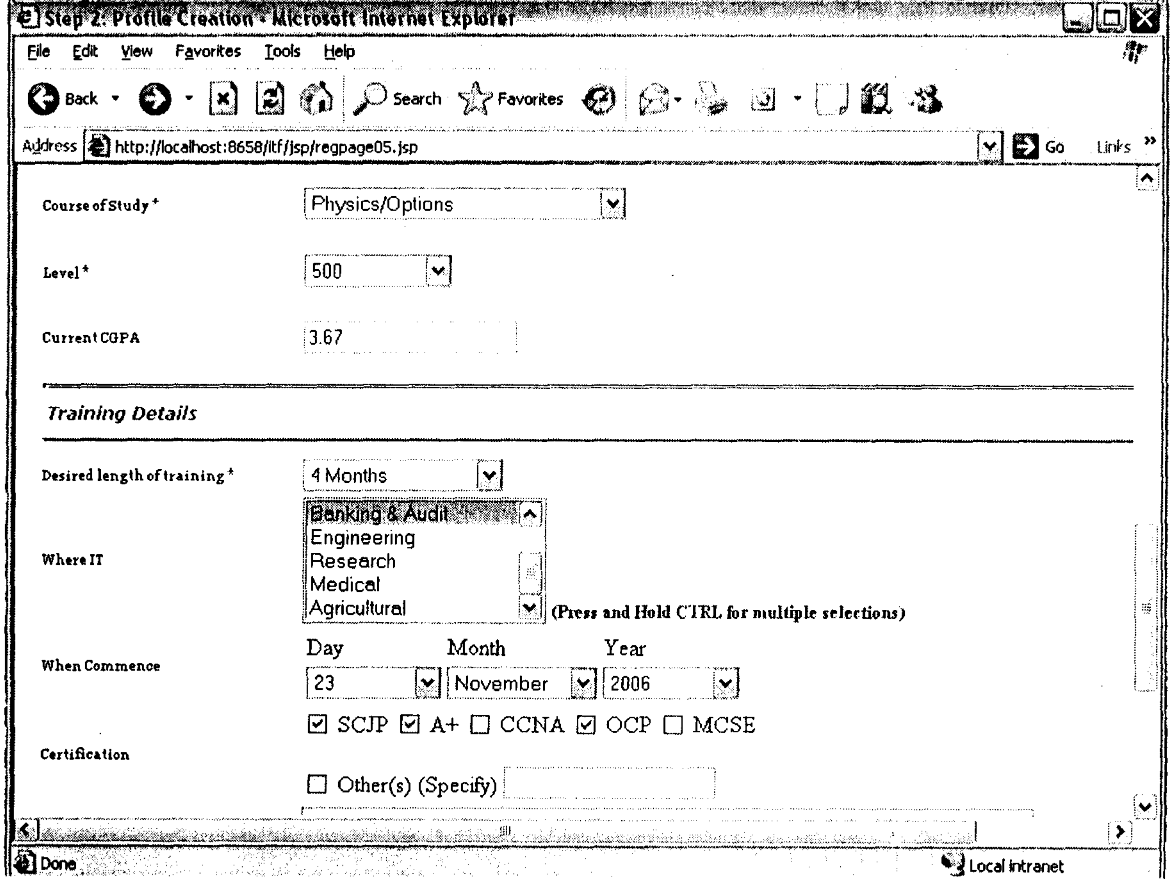
Phone Number*: 080-5555-9999

Email*: hajara@yahoo.com

Institution Details

Institution of study*: Ahmadu Bello University Zaria

Faculty/School*: Science/Science Education



http://localhost:8658/itf/jsp/regpage05.jsp

Course of Study*: Physics/Options

Level*: 500

Current CGPA: 3.67

Training Details

Desired length of training*: 4 Months

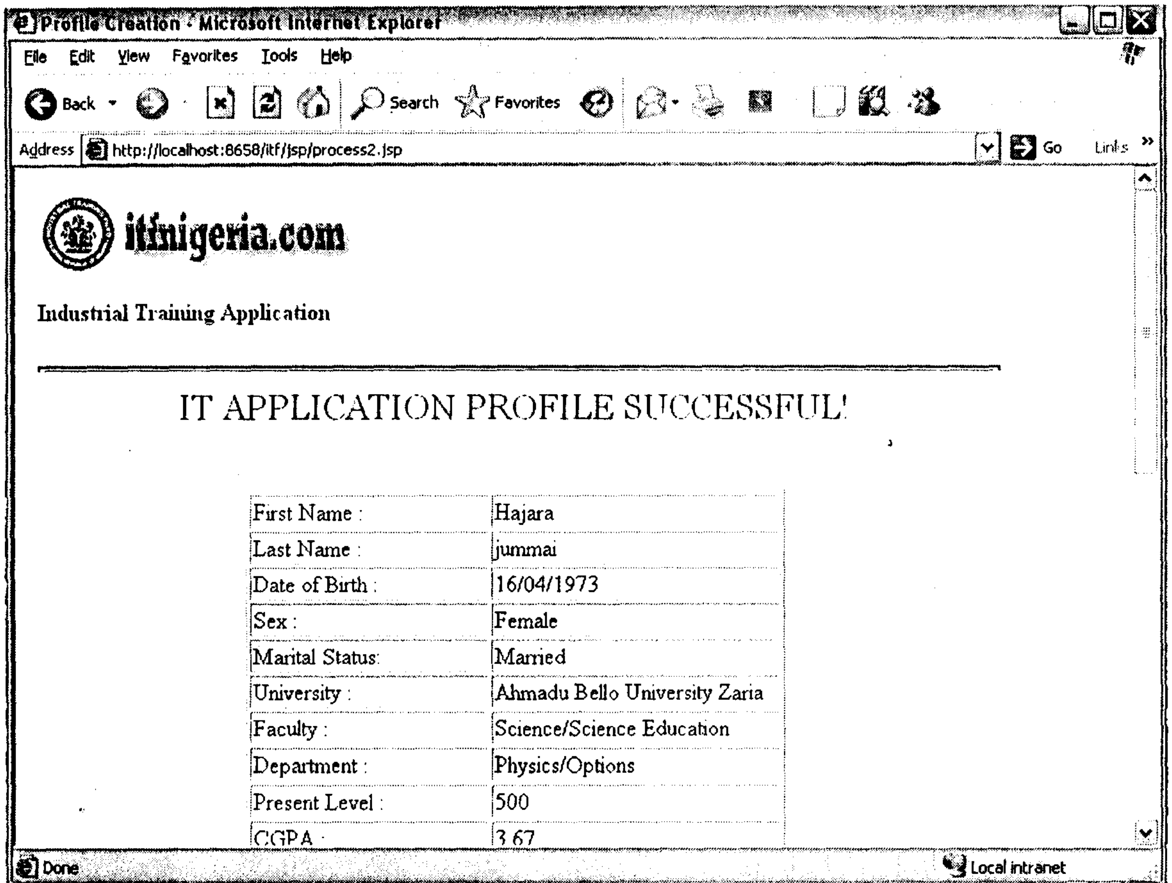
Where IT:
Banking & Audit
Engineering
Research
Medical
Agricultural
(Press and Hold CTRL for multiple selections)

When Commence: Day 23 Month November Year 2006

Certification: ☑ SCJP ☑ A+ □ CCNA ☑ OCP □ MCSE

□ Other(s) (Specify)

4.3 APPLICATION COMPLETION SUMMARY

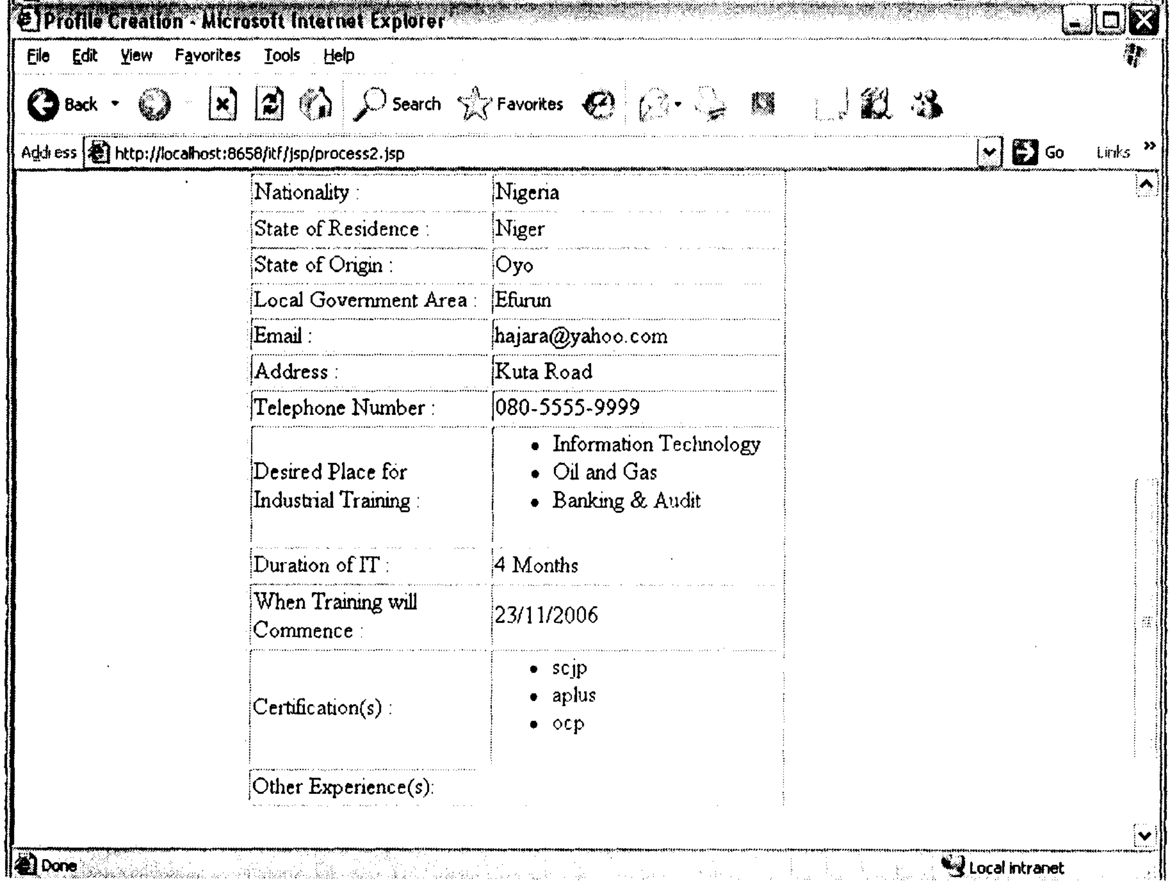

Profile Creation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://localhost:8658/itf/jsp/process2.jsp
Go
Links
Done
Local intranet

itfnigeria.com

Industrial Training Application

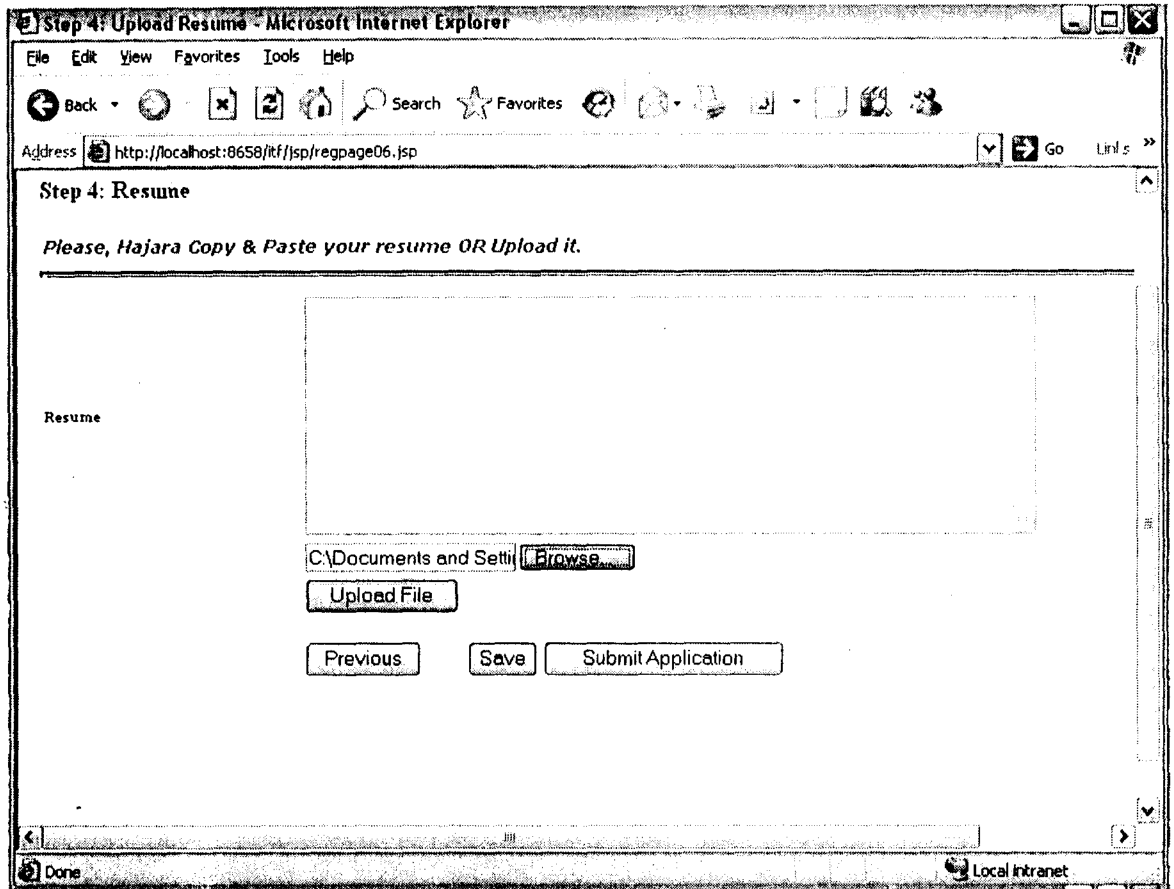
IT APPLICATION PROFILE SUCCESSFUL!

First Name :	Hajara
Last Name :	jummai
Date of Birth :	16/04/1973
Sex :	Female
Marital Status:	Married
University :	Ahmadu Bello University Zaria
Faculty :	Science/Science Education
Department :	Physics/Options
Present Level :	500
CGPA :	3.67


Profile Creation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
http://localhost:8658/itf/jsp/process2.jsp
Go
Links
Done
Local intranet

Nationality :	Nigeria
State of Residence :	Niger
State of Origin :	Oyo
Local Government Area :	Efurun
Email :	hajara@yahoo.com
Address :	Kuta Road
Telephone Number :	080-5555-9999
Desired Place for Industrial Training :	• Information Technology • Oil and Gas • Banking & Audit
Duration of IT :	4 Months
When Training will Commence :	23/11/2006
Certification(s) :	• scjp • aplus • ocp
Other Experience(s):	

4.4 RESUME UPLOAD



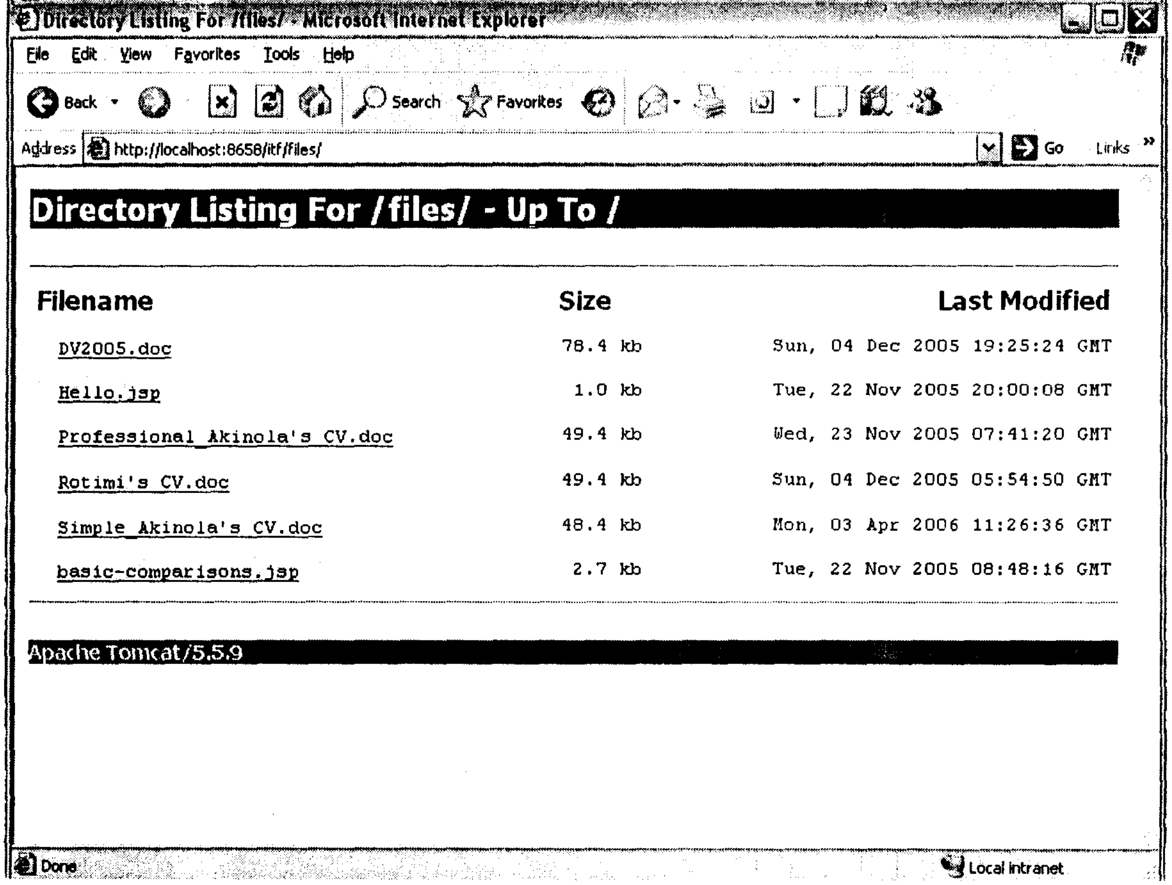
http://localhost:8658/itf/jsp/regpage06.jsp

Step 4: Resume

Please, Hajara Copy & Paste your resume OR Upload it.

Resume

C:\Documents and Setti


Directory Listing For /files/ - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
Done
Local intranet

http://localhost:8658/itf/files/

Directory Listing For /files/ - Up To /

Filename	Size	Last Modified
DV2005.doc	78.4 kb	Sun, 04 Dec 2005 19:25:24 GMT
Hello.jsp	1.0 kb	Tue, 22 Nov 2005 20:00:08 GMT
Professional_Akinola's_CV.doc	49.4 kb	Wed, 23 Nov 2005 07:41:20 GMT
Rotimi's_CV.doc	49.4 kb	Sun, 04 Dec 2005 05:54:50 GMT
Simple_Akinola's_CV.doc	48.4 kb	Mon, 03 Apr 2006 11:26:36 GMT
basic-comparisons.jsp	2.7 kb	Tue, 22 Nov 2005 08:48:16 GMT

Apache Tomcat/5.5.9

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

The Internet today is becoming the backbone and foundation stone for developing significant information technology (IT) and it is very important for the Industrial Training Fund (ITF), to stay connected or be on the same pedestal with other industrial training institutions in the developing and developed countries.

The Students' Industrial Training Online Application is web application software that connects to an Oracle database server. If adopted and implemented by ITF (the organizers of industrial training in Nigeria), students who are due for industrial training would be able to apply for their training on-line by visiting Internet cafes and logging-on to the ITF website to fill the necessary forms.

The adoption and implementation of this software by ITF would require some financing. It is indeed necessary for the Federal Government of Nigeria to increase its investment in information technology. Furthermore, the implementation of this software will bring about the following benefits: -

1) It will enhance the image of the nation's industrial training scheme.
2) It will eliminate the problems inherent in the current process.
3) It will provide an easier and more convenient application for students.
4) It will provide faster feedbacks to students, organizations and institutions.
5) It will eliminate the problems which students face in communicating with organizations of their choice.

5.2 RECOMMENDATION

With the advent of the Internet Technology, the use of the Internet has been recognized by various organizations as a better means of communicating, processing and retrieving data or information for the day-to-day running of the organization. The proper utilization of the Internet can achieve any desired organizational goals. However, the goals of the ITF can be achieved with the proper utilization of the Internet.

Education philanthropists, private enterprises and the Federal Government can contribute by sponsoring the ITF with logistic support for the development of information systems, creating efficient awareness on information technology (IT), donating computer systems and computer hardware facilities required to the various branches of ITF.

The ITF needs to co-ordinate students' industrial training application online, to enhance its efficiency in this regard, and ease the difficulties encountered by students in securing placements.

There should be computer appreciation training for all the staff concerned as well as recruitment of operators and web administrators or training of some technical assistance.

In addition to this, there should be a programmer engaged for the purpose of program maintenance and development. The programmer should be fully knowledgeable in the concept of Internet and server side application and be able to write programs in Java and Oracle.

REFERENCES

APTEC World Wide, Web Page Designing with DreamWeaver 4, (Unpublished) P.7-35.

Codd, E.F. (June 1970): A Relational Model of Data for Large Shared Data Banks: Seminar Paper, vol 13, No.6.

Debra Schinder. Computer Networking Essentials. www.e-Learningcenter.com.

Donald Koosis., David Koosis. (1998): Java Programming for Dummies, 3rd P. 9-17.

Industrial Training Fund branch office, Abuja, Nigeria.

Jason Hunter., William Crawford. (1998): Java Servlet Programming, First Edition: P.17-29.

Michael, I.A. (1997): Handbook on Personnel Management and Training Practice: P.126.

Microsoft Corporation. Encarta Premium Suite 2004 Encyclopedia.

The Encyclopedia Americana (Gronlier Incorporated 1988) vol 15, P.144.

www.google.com\industrialtraining.

APPENDIX: PROGRAM LISTING

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.*;
import java.io.*;
import java.sql.*;
import java.util.Date;
import javax.mail.*;
import javax.mail.internet.*;
import javax.activation.*;
import org.apache.commons.mail.*;

import com.bean.StudentBean;
import com.jsp.connectionpool.*;

public class DBHandler1 extends HttpServlet {
  private Connection con = null;
  private Statement statement;
  private ConnectionPool pool;

  // set up database connection and create SQL statement
   public void init( ServletConfig config ) throws ServletException   {
        pool = new ConnectionPool();
    // attempt database connection (using connection pooling) and create Statement
    try {
      // initialize the pool
        pool.setURL("jdbc:oracle:oci8:@dbavj");
        pool.setUsername("administrator");
        pool.setPassword("akinola");
        pool.setSize(5);
        pool.initializePool();
        // Get a connection from the ConnectionPool
        con = pool.getConnection();
    }

    // for any exception throw an UnavailableException to
    // indicate that the servlet is not currently available
    catch ( Exception exception ) {
      exception.printStackTrace();
      throw new UnavailableException( exception.getMessage() );
    }

  } // end of init method


  public void doPost (HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    response.setContentType( "text/html" );
    PrintWriter out = response.getWriter();
    HttpSession session = request.getSession();

    // start XHTML document
    out.println( "<?xml version = \"1.0\"?>" );

    out.println( "<!DOCTYPE html PUBLIC \"-//W3C//DTD " +
      "XHTML 1.0 Strict//EN\" \"http://www.w3.org" +
      "/TR/xhtml1/DTD/xhtml1-strict.dtd\">" );

    out.println(
      "<html xmlns = \"http://www.w3.org/1999/xhtml\">" );

    // head section of document
   out.println( "<head>" );
   out.println( "<title>Profile Creation</title>");
   out.println( "<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-
        8859-1\">");

        out.println( "<style type=\"text/css\">");
        out.println( "<!--");
        out.println( ".HoriRule {");
        out.println( "  background-color: #006699;");
        out.println( "  color: #006699;");
        out.println( "  font-weight: bolder;");
        out.println( "  border: thick double #006699;");
        out.println( "}");
        out.println( ".Horule {");
        out.println( "  background-color: #006699;");
        out.println( "  padding: 0em;");
        out.println( "}");
    out.println( ".textbody {");
    out.println( "  font-family: Georgia, \"Times New Roman\", Times, serif;");
    out.println( "  font-size: 10px;");
    out.println( "  font-style: normal;");
    out.println( "  color: #000000;");
    out.println( "  text-decoration: blink;");
    out.println( "  font-weight: normal;");
    out.println( "  font-variant: normal;");
    out.println( "}");
    out.println( ".unnamed1 {");
    out.println( "  font-family: \"Times New Roman\", Times, serif;");
    out.println( "  font-size: 12px;");
    out.println( "  font-style: normal;");
    out.println( "  font-weight: bold;");
    out.println( "  color: #FF0000;");
    out.println( "}");
    out.println( ".style1 {");
    out.println( "  color: #0000FF;");
    out.println( "  font-weight: bold;");
    out.println( "}");
    out.println( ".style2 {color: #0000FF}");
    out.println( ".style4 {");
    out.println( "  font-size: 16px;");
    out.println( "  color: #FF0000;");
    out.println( "}");
    out.println( "-->");
    out.println( "</style>");
    out.println( "</head>");

try {
 StudentBean sb = (StudentBean)session.getAttribute("formHandler");
 boolean userExists = false;
 String query;

 // create Statement to query database
 statement = con.createStatement();

 // Perform a database query ensuring that the username does not exist
 // First obtain a list of Users
```

```
ResultSet rs = statement.executeQuery(
  "SELECT loginID FROM tblUsers" );

// get row data and set userExists=true if user is found in database
while ( rs.next() ) {
  if (rs.getString("loginID") != null)
      userExists = true;
}

rs.close();

if (userExists) { //connect user and perform update
    // Print Success Message and Just created profile
    String DOBDay = "";
    String DOBMonth = "";
    String DOBYear = "";
    String WhenDay = "";
    String WhenMonth = "";
    String WhenYear = "";

    out.println( "<body class=\"sub\">");
    out.println( "<table width=90% border=0 cellspacing=2 cellpadding=4
   style=text-align: center>");
    out.println( "<tr>");
    out.println( "<td width=100% align=left valign=bottom
   class=StoryContentColor style=vertical-align: bottomp><h4
   class=TitleColor><img src=itfnig.gif alt=itflogo name=itflogo width=225
   height=50 id=itflogo /></h4>");
    out.println( "<h4 class=TitleColor>Industrial Training Application</h4>");
    out.println( "<hr align=center width=100% size=4 noshade class=HoriRule
   style=text-align: center> <table width=529 border=0 align=center>");
    out.println( "<tr align=center>");
    out.println( "<td colspan=2 valign=top>");
    out.println( "<p><span class=textbody><font size=5>IT APPLICATION
   PROFILE SUCCESSFUL!</font></span></p>");
    out.println( "<p>&nbsp;</p></td>");
    out.println( "</tr>");
    out.println( "<tr align=center bgcolor=#ffffff>");
    out.println( "<td height=35><table width=370 border=1>");
```

```
 out.println( "<tr><td width=175><span class=style2>First Name :
</span></td>");
 out.println("<td width=195>" + sb.getFirstName() + "</td></tr>");

 out.println( "<tr><td><span class=style2>Last Name : </span></td>");
 out.println("<td>" + sb.getLastName() + "</td></tr>");

 out.println( "<tr><td><span class=style2>Date of Birth : </span></td>");

 DOBDay = request.getParameter("dDay");
 DOBMonth = request.getParameter("dMonth");
 DOBYear = request.getParameter("dYear");

 if(DOBDay.equals("Pick a Day")) DOBDay = "00";
 if(DOBYear.equals("Pick a year")) DOBYear = "0000";

 if(DOBMonth.equals("January")) DOBMonth = "01";
 else if(DOBMonth.equals("February")) DOBMonth = "02";
 else if(DOBMonth.equals("March")) DOBMonth = "03";
 else if(DOBMonth.equals("April")) DOBMonth = "04";
 else if(DOBMonth.equals("May")) DOBMonth = "05";
 else if(DOBMonth.equals("June")) DOBMonth = "06";
 else if(DOBMonth.equals("July")) DOBMonth = "07";
 else if(DOBMonth.equals("August")) DOBMonth = "08";
 else if(DOBMonth.equals("September")) DOBMonth = "09";
 else if(DOBMonth.equals("October")) DOBMonth = "10";
 else if(DOBMonth.equals("November")) DOBMonth = "11";
 else if(DOBMonth.equals("December")) DOBMonth = "12";
 else DOBMonth = "00";

String dob = DOBDay + "/" + DOBMonth + "/" + DOBYear;
 sb.setDOB(dob);
 out.println("<td>" + dob + "<br /></td></tr>");

 String sex = request.getParameter("sex");
 out.println( "<tr><td><span class=style2>Sex : </span></td>");
 out.println("<td>"+ sex + "</td></tr>");
 sb.setSex(sex);
```

```
String maritalSta = request.getParameter("maritalSta");
out.println( "<tr><td><span class=style2>Marital Status: </span></td>");
out.println("<td>"+ maritalSta + "</td></tr>");
sb.setMarital(maritalSta);

String university = request.getParameter("university");
out.println( "<tr><td><span class=style2>University : </span></td>");
out.println("<td>"+ university + "</td></tr>");
sb.setUniversity(university);

String faculty = request.getParameter("faculty");
out.println( "<tr><td><span class=style2>Faculty : </span></td>");
out.println("<td>"+ faculty + "</td></tr>");
sb.setFaculty(faculty);

String dept = request.getParameter("dept");
out.println( "<tr><td><span class=style2>Department : </span></td>");
out.println("<td>"+ request.getParameter("dept") + "</td></tr>");
sb.setDept(dept);

String level = request.getParameter("intLevel");
out.println( "<tr><td><span class=style2>Present Level : </span></td>");
out.println("<td>"+ level + "</td></tr>");
sb.setLevel(level);

String cgpa = request.getParameter("cgpa");
out.println( "<tr><td><span class=style2>CGPA : </span></td>");
out.println("<td>"+ cgpa + "</td></tr>");
sb.setCgpa(cgpa);

String nationality = request.getParameter("nationality");
out.println( "<tr><td><span class=style2>Nationality : </span></td>");
out.println("<td>"+ nationality + "</td></tr>");
sb.setNationality(nationality);

String stateResident = request.getParameter("stateResident");
out.println( "<tr><td><span class=style2>State of Residence :
</span></td>");
```

```
 out.println("<td>"+ stateResident + "</td></tr>");
 sb.setStateResident(stateResident);

 String StateOrigin = request.getParameter("StateOrigin");
 out.println( "<tr><td><span class=style2>State of Origin : </span></td>");
 out.println("<td>"+ StateOrigin + "</td></tr>");
 sb.setStateOrigin(StateOrigin);

 String lga = request.getParameter("LGA");
 out.println( "<tr><td><span class=style2>Local Government Area :
</span></td>");
 out.println("<td>"+ lga + "</td></tr>");
 sb.setLGA(lga);

 String email = request.getParameter("email");
 out.println( "<tr><td><span class=style2>Email : </span></td>");
 out.println("<td>"+ email + "</td></tr>");
 sb.setEmail(email);

 String address = request.getParameter("address");
 out.println( "<tr><td><span class=style2>Address : </span></td>");
 out.println("<td>"+ address + "</td></tr>");
 sb.setAddress(address);

 String PhoneNo = request.getParameter("PhoneNo");
 out.println( "<tr><td><span class=style2>Telephone Number :
</span></td>");
 out.println("<td>"+ PhoneNo + "</td></tr>");
 sb.setPhone(PhoneNo);

 out.println( "<tr><td><span class=style2>Desired Place for Industrial Training
: </span></td>");
 String places = "";
out.println("<td valign=bottom>");
 String[] itplaces = request.getParameterValues("whereIT");
 if (!itplaces[0].equals("1")) {
        out.println("<ul>");
        for (int i=0; i<itplaces.length; i++) {
               out.println("<li>" + itplaces[i]);
```

```
                places += "*" + itplaces[i];
            }
            out.println("</ul>");
    } else out.println("Nothing was selected");
  out.println("</td>");

    // AN ALTERNATIVE FOR THE ABOVE !!!!!!
    //get the request dispatcher
  //RequestDispatcher                              dispatcher                                   =
   request.getRequestDispatcher("/jsp/itplaces.jsp");
  //include jsp file to display choice of IT place(s)
  //dispatcher.include(request, response);

    String duration = request.getParameter("duration");
    out.println( "<tr><td><span class=style2>Duration of IT : </span></td>");
    out.println("<td>" + duration + "</td></tr>");
    sb.setDuration(duration);

    out.println( "<tr><td><span class=style2>When Training will Commence :
   </span></td>");

    WhenDay = request.getParameter("wDay");
    WhenMonth = request.getParameter("wMonth");
    WhenYear = request.getParameter("wYear");

    if(WhenDay.equals("Pick a Day")) WhenDay = "00";
    if(WhenYear.equals("Pick a year")) WhenYear = "0000";

    if(WhenMonth.equals("January")) WhenMonth = "01";
    else if(WhenMonth.equals("February")) WhenMonth = "02";
    else if(WhenMonth.equals("March")) WhenMonth = "03";
    else if(WhenMonth.equals("April")) WhenMonth = "04";
    else if(WhenMonth.equals("May")) WhenMonth = "05";
    else if(WhenMonth.equals("June")) WhenMonth = "06";
    else if(WhenMonth.equals("July")) WhenMonth = "07";
    else if(WhenMonth.equals("August")) WhenMonth = "08";
    else if(WhenMonth.equals("September")) WhenMonth = "09";
    else if(WhenMonth.equals("October")) WhenMonth = "10";
    else if(WhenMonth.equals("November")) WhenMonth = "11";
  else if(WhenMonth.equals("December")) WhenMonth = "12";
  else WhenMonth = "00";

  String WhenCommence = WhenDay + "/" + WhenMonth + "/" + WhenYear;
  sb.setWhenComm(WhenCommence);

  out.println("<td>" + WhenCommence + "<br /></td></tr>");

  out.println( "<tr><td><span class=style2>Certification(s) : </span></td>");

  out.println("<td valign=bottom>");
  String certificates = "";
  String[] cert = request.getParameterValues("certifications");
  if (!cert[0].equals("1")) {
          out.println("<ul>");
          for (int i=0; i<cert.length; i++) {
                  out.println("<li>"+cert[i]);
                  certificates += "*" + cert[i];
          }
          out.println("</ul>");
  } else out.println("Nothing was selected");
  out.println("</td>");

  String experience = request.getParameter("experience");
out.println( "<tr><td><span class=style2>Other Experience(s): </span></td>");
  out.println("<td>"+ experience + "</td></tr>");
  sb.setExperience(experience);

  out.println( "</table>");
  out.println( "<tr align=center>");
  out.println( "<td height=35 colspan=2><p>&nbsp;</p>");
  out.println( "<p><a href=\"regpage06.jsp\" class=\"unnamed1
 style4\">Continue</a></p></td></tr>");
  out.println( "</table>");
  out.println( "<hr align=center width=100% size=4 noshade class=\"HoriRule\"
 style=\"text-align: center\"></td></tr>");
  out.println( "<tr>");
  out.println( "<td align=left valign=bottom class=\"StoryContentColor\"
 style=\"vertical-align: bottomp\">&nbsp;</td></tr>");
```

```
    out.println( "</table>");
    out.println( "</body>" );
    out.println( "</html>" );
    out.close();

  // do insertion of records
  //retrieve the bean properties and store them into the database.
  query = "INSERT INTO StudentData
   (loginID,dob,sex,maritalSta,university,faculty,dept,intLevel,cgpa, nationality,"
   +

   "stateResident,StateOrigin,LGA,email,address,PhoneNo,experience,whereIT,
   " +
                        "duration,whencommence,certification) VALUES ('" +
   sb.getLoginID() + "', " + "TO_DATE('" + dob + "', 'DD/MM/YYYY'), '" +
   sb.getSex() +
                                          "', '" + sb.getMarital() + "', '" +
   sb.getUniversity() + "', '" + sb.getFaculty() + "', '" + sb.getDept() +
                                      "', '" + sb.getLevel() + "', " +
   Double.parseDouble(cgpa) + ", '" + sb.getNationality() + "', '" +
   sb.getStateResident() +
                                          "', '" + sb.getStateOrigin() + "', '" +
   sb.getLGA() + "', '" + sb.getEmail() + "', '" + sb.getAddress() + "', '" +
   sb.getPhone()+
                                      "', '" + sb.getExperience() + "', '" + places + "', '"
   + sb.getDuration() + "', " + "TO_DATE('" + WhenCommence + "',
   'DD/MM/YYYY'), '" + certificates + "' )";
  statement.executeUpdate( query );

  // Send message to email Address
String student = sb.getFirstName() + sb.getLastName();
String message = "<html> Here are the information you provided <table
   width=370 border=1>" +
         "<tr><td width=175><span class=style2>First Name : </span></td>" +
         "<td width=195>" + sb.getFirstName() + "</td></tr>" +
         "<tr><td><span class=style2>Last Name : </span></td>" +
         "<td>" + sb.getLastName() + "</td></tr>" +
         "<tr><td><span class=style2>Date of Birth : </span></td>" +
         "<td>" + dob + "<br /></td></tr>" +
        "<tr><td><span class=style2>Sex : </span></td>" +
        "<td>"+ sex + "</td></tr>" +
        "<tr><td><span class=style2>Marital Status: </span></td>" +
        "<td>"+ maritalSta + "</td></tr>" +
        "<tr><td><span class=style2>University : </span></td>" +
        "<td>"+ university + "</td></tr>" +
        "<tr><td><span class=style2>Faculty : </span></td>" +
        "<td>"+ faculty + "</td></tr>" +
        "<tr><td><span class=style2>Department : </span></td>" +
        "<td>"+ request.getParameter("dept") + "</td></tr>" +
        "<tr><td><span class=style2>Present Level : </span></td>" +
        "<td>"+ level + "</td></tr>" +
        "<tr><td><span class=style2>CGPA : </span></td>" +
        "<td>"+ cgpa + "</td></tr>" +
        "<tr><td><span class=style2>Nationality : </span></td>" +
        "<td>"+ nationality + "</td></tr>" +
        "<tr><td><span class=style2>State of Residence : </span></td>" +
        "<td>"+ stateResident + "</td></tr>" +
        "<tr><td><span class=style2>State of Origin : </span></td>" +
        "<td>"+ StateOrigin + "</td></tr>" +
        "<tr><td><span class=style2>Local Government Area : </span></td>"
+
        "<td>"+ lga + "</td></tr>" +
        "<tr><td><span class=style2>Email : </span></td>" +
        "<td>"+ email + "</td></tr>" +
        "<tr><td><span class=style2>Address : </span></td>" +
        "<td>"+ address + "</td></tr>" +
        "<tr><td><span class=style2>Telephone Number : </span></td>" +
        "<td>"+ PhoneNo + "</td></tr>" +
        "<tr><td><span class=style2>Desired Place for Industrial Training :
</span></td>" +
      "<td valign=bottom>" + places + "</td>" +
        "<tr><td><span class=style2>Duration of IT : </span></td>" +
        "<td>" + duration + "</td></tr>" +
        "<tr><td><span class=style2>When Training will Commence :
</span></td>" +
        "<td>" + WhenCommence + "<br /></td></tr>" +
        "<tr><td><span class=style2>Certification(s) : </span></td>" +
        "<td valign=bottom>" + certificates + "</td>" +
                "<tr><td><span class=style2>Other Experience(s): </span></td>" +
                  "<td>"+ experience + "</td></tr></table>You can Log back to edit your
        Application</html>";

   HtmlEmail mail = new HtmlEmail();
   mail.setHostName("mail.yahoo.com");
   mail.addTo(email, student);
   mail.setFrom("jasonva2000@yahoo.com", "Victor Akinola");
   mail.setSubject("Industrial Training Application");
   mail.setHtmlMsg(message);
   mail.setTextMsg("Your email Client deos not support HTML messages");
   mail.send();

    } // end if

    /*
    //get the request dispatcher
    RequestDispatcher                    dispatcher                    =
        request.getRequestDispatcher("/jsp/success.jsp");
    //forward to the jsp file to display retry message
    dispatcher.forward(request, response);
    */
    else { // user does not exist
         sb.setErrors("loginID","User does not Exist: Create a Profile for this user");
         //get the request dispatcher
         RequestDispatcher                    dispatcher                    =
        request.getRequestDispatcher("/jsp/nouser.jsp");
            //forward to the jsp file to display retry message
         dispatcher.forward(request, response);
    }
  } catch (Exception ex) {
    ex.printStackTrace();
  }
} // end doPost

// close SQL statements and database when servlet terminates
 public void destroy()   {
    // attempt to close statements and database connection
    try {
      statement.close();
      con.close();
    }

    // handle database exceptions by returning error to client
    catch( SQLException sqlException ) {
      sqlException.printStackTrace();
    }
    finally {
        try {
          if ( con != null ) {
            // release the connection no matter what
            pool.releaseConnection(con);
          }
        }
        catch (Exception e) {
          e.printStackTrace();
        }
    }
  } // end destroy()
}

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class FileUpload extends HttpServlet {
    public void doPost(HttpServletRequest request, HttpServletResponse
    response)  throws IOException, ServletException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        out.println("<html>");
        out.println("File upload success. <a
    href=\"http://localhost:8658/itf/jsp/files/");
        out.println("\">Click here to browse through all uploaded ");
        out.println("files.</a><br>");

        ServletInputStream sis = request.getInputStream();
    StringWriter sw = new StringWriter();
    int i = sis. read();
    for(; i != -1 && i != '\r'; i=sis.read()) {
        sw.write(i);
    }
    sis.read(); // ditch '\n'
    String delimiter = sw.toString();
    int count = 0;
    while(true) {
        StringWriter h = new StringWriter();
        int[] temp = new int[4];
        temp[0] = (byte)sis.read();
        temp[1] = (byte)sis.read();
        temp[2] = (byte)sis.read();
        h.write(temp[0]);
        h.write(temp[1]);
        h.write(temp[2]);
        //  read buffer
        for(temp[3]=sis.read();temp[3] != -1; temp[3]=sis.read()) {
            if(temp[0] == '\r' && temp[1] == '\n' && temp[2] == '\r' &&
temp[3] == '\n') {
                break;
            }
            h.write(temp[3]);
            temp[0] = temp[1];
            temp[1] = temp[2];
            temp[2] = temp[3];
        }
        String header = h.toString();

        int startName = header.indexOf("name=\"");
        int endName = header.indexOf("\"", startName + 6);
        if(startName == -1 || endName == -1) {
            break;
        }
        String name = header.substring(startName+6, endName);
        if(name.equals("file")) {
            startName = header.indexOf("filename=\"");
            endName = header.indexOf("\"", startName+10);
            String filename = header.substring(startName+10,
endName);
            ServletContext sc =
request.getSession().getServletContext();

            File file = new File(sc.getRealPath("/files"));
            file.mkdirs();
            FileOutputStream fos = new FileOutputStream(filename);
//FileOutputStream fos = new FileOutputStream(sc.getRealPath("/") + filename);

            // write whole file to disk
            int length = 0;
            delimiter = "\r\n"+delimiter;
            byte[] body = new byte[delimiter.length()];
            for(int j = 0; j < body.length; j++) {
                body[j] = (byte)sis.read();
            }
            // check it wasn't a 0 length file
            if(!delimiter.equals(new String(body))) {
                int e = body.length - 1;
                i = sis.read();
                for(;i != -1; i = sis.read()) {
                    fos.write(body[0]);
                    for(int l = 0; l < body.length-1;l++) {
                        body[l] = body[l+1];
                    }
                    body[e] = (byte)i;
                    if(delimiter.equals(new String(body))) {
                        break;
                    }
                    length++;
                }
            }
            fos.flush();
            fos.close();
        }
        if(sis.read() == '-' && sis.read() == '-') {
            break;
        }
```

```
                }
                out.println("</html>");
        }

        public void doGet(HttpServletRequest request, HttpServletResponse
        response) throws IOException, ServletException {
                doPost(request, response);
        }
}

// JavaBean to store data for a student in the ITF database.
package com.bean;

import java.util.*;
import java.util.regex.*;

public class StudentBean {
  private String firstName;
  private String lastName;
  private String email;
  private String loginID;
  private String password1;
  private String password2;
  private Hashtable<String, String> errors;
  private String loggedin;

  private String middlename;
  private String dob;
  private String whencommence;
  private String sex;
  private String maritalSta;
  private String address;
  //picture      BLOB NOT NULL,
  private String university;
  private String faculty;
  private String dept;
  private String intLevel;
  private String cgpa;
  private String nationality;
private String stateResident;
private String City;
private String StateOrigin;
private String LGA;
private String PhoneNo;
private String experience;
private String[] whereIT;
private String duration;
private String[] certification;

// For Date of Birth (dob)
private String dMonth, dDay, dYear;
//For whencommence Date
private String wMonth, wDay, wYear;

// Bean Constructor
public StudentBean() {
  firstName = "";
  lastName = "";
  loginID = "";
  password1 = "";
  password2 = "";
  loggedin = "";


  dob = "";
  nationality = "";
  sex = "";
  maritalSta = "";
  university = "";
  faculty = "";
  dept = "";
  intLevel = "";
  stateResident = "";
  City = "";
  StateOrigin = "";
  LGA = "";
  PhoneNo = "";
  email = "";
  cgpa = "";
  address = "";
  duration = "";
  whereIT = new String[] { "1" };
  certification = new String[] { "1" };
  errors = new Hashtable<String, String> ();

  dMonth = "";
  dDay = "";
  dYear = "";
  wMonth = "";
  wDay = "";
  wYear = "";;
}

// For Validation checks !!!!

public boolean validateprofile() {
  boolean allOk = true;
      if (firstName.equals("")) {
        errors.put("firstName","Please enter your first name");
        firstName = "";
        allOk = false;
      }

    if (lastName.equals("")) {
        errors.put("lastName","Please enter your last name");
        lastName = "";
        allOk = false;
      }

      if (loginID.equals("")) {
        errors.put("loginID","Please enter a Login ID");
        loginID = "";
        allOk = false;
      }

      if (password1.equals("") ) {
        errors.put("password1","Please enter a valid password");
```

```
       password1 = "";
       allOk = false;
      }

      if (!password1.equals("") && (password2.equals("") ||
      !password1.equals(password2))) {
       errors.put("password2","Please confirm your password");
       password2 = "";
       allOk = false;
      }

    return allOk;
   }

  public boolean validate() {
   boolean allOk = true;

      /*if (email.equals("") || (email.indexOf('@') == -1)) {
       errors.put("email","Please enter a valid email address");
       email = "";
       allOk = false;
      }

      //if (!PhoneNo.matches("[0-9]\\d{2}-[2-6]\\d{3}-\\d{4}") ) {
      if (!PhoneNo.matches("[0][8][0]-[2-6]\\d{3}-\\d{4}") ) {
       errors.put("PhoneNo","Please confirm your phone Number: Use Format::
      080-9999-9999");
       PhoneNo = "";
       allOk = false;
      }

      if (!dob.matches("[0-3][0-9]-[0-1][1-9]-\\d{4}") ) {
       errors.put("dob","Please Use Format: DD-MM-YYYY");
       dob = "";
       allOk = false;
      }

      if (nationality.equals("Select a country")) {
       errors.put("nationality","Please pick a Nationality");
```

```
 nationality = "Select a country";
 allOk = false;
}

if (sex.equals("")) {
 errors.put("sex","Please choose a Gender");
 sex = "";
 allOk = false;
}

if (maritalSta.equals("")) {
 errors.put("maritalSta","Please pick a Marital Status");
 maritalSta = "";
 allOk = false;
}

if (university.equals("Please select one")) {
 errors.put("university","Please Pick a University");
 university = "Please select one";
 allOk = false;
}

if (faculty.equals("Please Select one")) {
 errors.put("faculty","Please pick a faculty");
 faculty = "Please select one";
 allOk = false;
}

if (dept.equals("Please Select one")) {
 errors.put("dept","Please pick a Department");
 dept = "Please select one";
 allOk = false;
}

if (intLevel.equals("Select Level")) {
 errors.put("intLevel","Please pick a Level");
 intLevel = "Select Level";
 allOk = false;
}
```

```
   if (stateResident.equals("Please Select one")) {
    errors.put("stateResident","Please pick a state of Residence");
    stateResident = "Please Select one";
    allOk = false;
   }

   if (City.equals("Please Select one")) {
    errors.put("City","Please pick a City");
    City = "Please Select one";
    allOk = false;
   }

   if (StateOrigin.equals("Please Select one")) {
    errors.put("StateOrigin","Please pick a State of Origin");
    StateOrigin = "Please Select one";
    allOk = false;
   }

   if (LGA.equals("Please Select one")) {
    errors.put("LGA","Please pick a Local Government Area");
    LGA = "Please Select one";
    allOk = false;
   }

   if (duration.equals("Please Select one")) {
    errors.put("duration","Please choose duration of IT"); ,
    duration = "Please Select one";
    allOk = false;
   } */
 return allOk;
}


public String getErrorMsg(String s) {
 String errorMsg =(String)errors.get(s.trim());
 return (errorMsg == null) ? "":errorMsg;
}
```

```
// Accessors
// Getters
// get the student's first name
public String getFirstName() {
  return firstName;
}

// get the student's last name
public String getLastName() {
  return lastName;
}

// get the student's email address
public String getEmail() {
  return email;
}

// get the student's userID
public String getLoginID() {
  return loginID;
}

public String getPassword1() {
 return password1;
}

public String getPassword2() {
 return password2;
}

public String getPhone() {
 return PhoneNo;
}

public String getDOB() {
 return dob;
}

public String getLoggedin() {
        return loggedin;
    }

    public String getStateResident() {
        return stateResident;
    }

    public String getStateOrigin() {
        return StateOrigin;
    }

    public String getCity() {
        return City;
    }

    public String getLGA() {
        return LGA;
    }

    public String getNationality() {
        return nationality;
    }

    public String getSex() {
        return sex;
    }

ublic String getMarital() {
eturn maritalSta;

blic String getUniversity() {
turn university;

lic String getFaculty() {
urn faculty;
```

```
public String getDept() {
  return dept;
}

public String getLevel() {
  return intLevel;
}

public String getCgpa() {
  return cgpa;
}

public String getAddress() {
  return address;
}

public String getDuration() {
  return duration;
}

public String getWhenComm() {
  return whencommence;
}

public String[] getWhereIT() {
  return whereIT;
}

public String getExperience() {
  return experience;
}

public String[] getCertification() {
  return certification;
}

  ublic String isSexRbSelected(String s) { // Sex Radio Buttons
   eturn (sex.equals(s))? "checked" : "";
```

```
public String isMaritalStaRbSelected(String s) { // Marital Status Radio Buttons
 return (maritalSta.equals(s))? "checked" : "";
}

public String isCbSelected(String s) { // Check Boxes
 boolean found=false;
  if (!certification[0].equals("1")) {
   for (int i = 0; i < certification.length; i++) {
     if (certification[i].equals(s)) {
      found=true;
      break;
     }
    }
    if (found) return "checked";
  }
  return "";
}

public String isLbSelected(String s) { // List Boxes
 boolean found = false;
  if (!whereIT[0].equals("1")) {
   for (int i = 0; i < whereIT.length; i++) {
     if (whereIT[i].equals(s)) {
      found = true;
      break;
     }
    }
    if (found) return "selected";
  }
  return "";
}

// Setters
// set the student's first name
public void setFirstName( String fname ) {
  firstName = fname;
}
```

```
// set the student's last name
public void setLastName( String lname ) {
  lastName = lname;
}

// set the student's email address
public void setEmail( String address ) {
  email = address;
}

public void setUserName( String uname ) {
  loginID = uname;
}

public void  setPassword1(String pswd1) {
 password1 = pswd1;
}

public void  setPassword2(String pswd2) {
 password2 = pswd2;
}

public void  setLoginID(String id) {
 loginID = id;
}

public void  setPhone(String phone) {
 PhoneNo = phone;
}

public void  setDOB(String DOB) {
 dob = DOB;
}

public void  setLoggedin(String lod) {
 loggedin = lod;
}

public void setStateResident(String dres) {
  stateResident = dres;
}

public void setStateOrigin(String sorigin) {
  StateOrigin = sorigin;
}

public void setCity(String c) {
  City = c;
}

public void setLGA(String lga) {
  LGA = lga;
}

public void setNationality(String nat) {
  nationality = nat;
}

public void setSex(String s) {
  sex = s;
}

public void setMarital(String mrt) {
  maritalSta = mrt;
}

public void setUniversity(String uni) {
  university = uni;
}

public void setFaculty(String fac) {
  faculty = fac;
}

public void setDept(String d) {
  dept = d;
}
```

```
  public void setLevel(String l) {
    intLevel = l;
  }

  public void setAddress(String adrs) {
    address = adrs;
  }

  public void setCgpa(String cg) {
    cgpa = cg;
  }

  public void setDuration(String d) {
    duration = d;
  }

  public void setWhenComm(String wc) {
    whencommence = wc;
  }

  public void setCertification(String[] cert) {
                certification = cert;
  }

  public void setWhereIT(String[] wit) {
                whereIT = wit;
  }

  public void setExperience(String expe) {
                experience = expe;
  }

  public void setErrors(String key, String msg) {
    errors.put(key,msg);
  }
}
```