

# **Web-Enabled Information Management System for the National Apprentice and Industrial Training Authority**

BY

**B.T.G.S.KUMAR**  
(01/AS/049)

This thesis is submitted in partial fulfillment of the requirements for the degree of  
Bachelor of Science in Physical Sciences

Department of Physical Sciences,  
Faculty of Applied Sciences,  
Sabaragamuwa University of Sri Lanka.  
Buttala.

August 2006

## Declaration

The content described in this thesis was practically implemented by me at the National Apprenticeship and Industrial Training Authority and the Faculty of Applied Sciences under the supervision of Mr. S.D.Wijewardhana and Mr. C.P. Wijesiriwardhana and the report described on this thesis has not been submitted by any one for another degree.

B.T.G.S.Kumara  
(01/AS/049)

.....  
Signature

Date: ..11.09.2006.....

Certified By,

Mr. S.D.Wijewardhana,  
External Supervisor,  
Provincial Director/NAITA,  
Sabaragamuwa Provincial Office,  
New Town,  
Ratnapura,  
Sri Lanka.

.....  
Signature

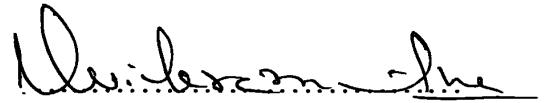
Date: ..11.09.2006.....

Mr. C.P. Wijesiriwardhana,  
Internal Supervisor,  
Lecturer/ Department of Physical Sciences,  
Faculty of Applied Sciences,  
Sabaragamuwa University of Sri Lanka,  
Buttala.

.....  
Signature

Date: ..15/09/2006.....

Dr (Mrs.) N. Wickramarathne,  
Head/Department of Physical Sciences,  
Faculty of Applied Sciences,  
Sabaragamuwa University of Sri Lanka,  
Buttala.



Signature

Date: 15/09/2006

## **Acknowledgement**

First I should express my sincere thanks to my internal supervisor Mr. C.P. Wijesiriwardhana , Lecturer, Department of Physical Sciences, Faculty of Applied Sciences for his encouragement and guidance through this project.

And also I wish to express my deepest gratitude to my external supervisor Mr. S.D. Wijewardhana, Provincial Director, Sabaragamuwa provincial office, NAITA, for his advice, encouragement and guidance through the study and for sparing his valuable time in bringing this study to a successful completion.

I express my sincere gratitude to all staff members of National Apprenticeship and Industrial Training Authority for there advice, encouragement and guidance and for sparing their valuable time in bringing this study to a successful completion.

And I express my sincere gratitude to Prof. Mahinda Wickramaratne, The Dean, Faculty of Applied Sciences, Sabaragamuwa University of Sri Lanka, and Dr (Mrs.) Nirmali Wickramaratne, Head, Department of Physical Sciences, Faculty of Applied Sciences, Sabaragamuwa University of Sri Lanka, for guiding me toward a successful completion.

Finally, I express my heart-felt gratitude towards the lectures for their cooperation throughout my study and my colleagues for their individual help and guidance at all times.



## **Abstract**

The National Apprentice and Industrial Training Authority (NAITA) performs a vital functions over the years in selecting and training thousands of youth in industrial and business undertakings throughout Sri Lanka. NAITA carry out several management activities such as recruiting beginners and releasing exam results. There are several disadvantages of this system mainly on entering and editing data. And also there is a possibility of forgetting to publish the new vacancies and about the services provided by the NAITA. This existing system is time consuming and inaccurate as it is a totally manual system. Therefore a software project was proposed to develop a web-based management information system for The National Apprenticeship Training and Industrial Training Authority.

The Linear sequential model was used for the system development process and it was started with feasibility study. Then the requirement analysis part was carried out. At this phase information was collected by interviews with potential users and by observing day-to-day documents. In the designing stage user interfaces, use case diagrams & databases of the system were designed according to the user requirements. Then, the system was implemented using a sever side scripting language called PHP: Hyper Text Preprocessor (PHP) and JavaScript as the scripting language. The database was implemented using MySQL and manipulate (feed, retrieve, store) data using Structured Query Language (SQL). User Interfaces, links and images were implemented using Photo Shop and Dream Viewer. Apache was used as the Web Server in this project. After coding, components were tested individually and finally the integrated system was tested.

At the end of the development process the main purpose of the project was achieved and the client was satisfied with the functionalities, usability, security and reliability of the system.

# CONTENT

Declaration .....	I
Acknowledgement .....	III
Abstract .....	IV
List of Abbreviations .....	V
List of Figures .....	VI
List of Tables .....	VII
Content .....	VIII

## **Chapter 1: Introduction ..... 1**

1.1 About NAITA .....	1
1.1.1 About Existing System.....	1
1.1.2 Web-Base Information Management System for the NAITA .....	2
1.2 Objectives .....	3
1.2.1 Overall Objective .....	3
1.2.2 Specific Objectives .....	3

## **Chapter 2: Literature Review ... ..4**

2.1 Web Database Application .....	4
2.1.1 Web server .....	4
2.1.2 Application Server .....	4
2.1.3 Internet Information Services .....	5
2.1.4 Dynamic Web page .....	5
2.1.5 Server Technologies for Web Applications .....	5
2.2 PHP: Hyper Text Preprocessor (PHP) .....	6
2.2.1 PHP in Object-Oriented Programming .....	10
2.2.1.1 Using Objects .....	10
2.2.1.2 Autoload .....	11
2.2.1.3 Data Encapsulation .....	12

2.2.1.4 Constructors and Destructors .....	12
2.3 Apache Web Server .....	13
2.4 Databases .....	13
2.4.1 Accessing a Database .....	13
2.4.2 ODBC/JDBC versus native API .....	14
2.4.3 MySQL .....	15
2.4.4 PHP with MySQL .....	16
2.4.5 To connect and select a database .....	18
2.5 HTML .....	18
2.6 Java script .....	19
2.7 Cascading Style Sheets (CSS) .....	19
2.7.1 How to Insert a Style Sheet .....	20
2.8 Software Engineering .....	20
2.8.1 Software Development Methods .....	21
2.8.2. Linear Sequential Model (Waterfall Approach) .....	21
2.8.2.1 Software Requirements Analysis .....	22
2.8.2.2 System Analysis and Design .....	22
2.8.2.3 Code Generation .....	23
2.8.2.4 Testing .....	23
2.8.2.5 Maintenance .....	24
<b>Chapter 3: Methodology .....</b>	<b>26</b>
3.1 Requirements Analysis .....	26
3.1.1 Requirement Capture .....	26
3.1.2 Functional Requirements Analysis .....	27
3.1.3 Non-functional requirement .....	28
3.2 System Analysis and Design .....	29
3.3 Coding and Implementation .....	30
3.3.1 Environment Used .....	31
3.3.2 Configuration of Apache web Server .....	40
3.4 Testing .....	40

3.4.1 Unit Testing .....	40
3.4.2 System Testing .....	41
<b>Chapter 4: Result and Discussion .....</b>	<b>42</b>
<b>Chapter 5: Conclusion .....</b>	<b>43</b>
<b>References.....</b>	<b>44</b>
<b>Appendix .....</b>	<b>45</b>

## List of Abbreviations

HTTP	Hyper Text Transfer Protocol
HTML	Hyper Text Markup Language
IIS	Internet Information Services
ASP	Active Server Pages
JSP	Java Server Pages
OOP	Object Oriented Programming
GUI	Graphical User Interface
ODBC	Open Database Connection
NCSA	National Center for Supercomputing Application
API	Application Programming Interfaces
JDBC	Java Database Connectivity
CSS	Cascading Style Sheets
JAD	joint application development
RAD	rapid application development
NCSA	National Center for Supercomputing Applications
SQL	Structured Query Language

# List of Figures

Page Number

## Chapter 2: Web Database Application

Figure 2.1	How PHP fits into the client-server model when user Request a Web page .....	9
Figure 2.2	The Process Involved With a Request for Static HTML Page.....	9
Figure 2.3	Dynamic web applications, using both PHP and MySQL.....	18
Figure 2.4	Waterfall Model.....	22

## Chapter 3 : Methodology

Figure 3.1	Designs for the Login Page to the Administration Module.....	29
Figure 3.2	Design of the web page .....	30
Figure 3.2	courses table .....	36

# List of Tables

	Page Number
<b>Chapter 2: Literature Review</b>	
Table 2.1 Most Popular Languages for Server Technologies .....	6
Table 2.2 Databases Support by The-PHP .....	8
Table 2.3 Database Interfaces for Web Application .....	15
<b>Chapter 3 : Methodology</b>	
Table 3.1 User Activities Table .....	27

# **Chapter 1**

## **Introduction**

### **1.1 About NAITA**

The more recent history of Vocational Training in Sri Lanka has been from its beginning inseparably connected with development of NAITA, the former National Apprenticeship Board (NAB). NAB was set up on 16<sup>th</sup> of December 1971, under the National Apprenticeship Act No. 49 of 1971, to formulate implement and administer the apprenticeship training in Sri Lanka. The NAB having served the nation almost 20 years in evolving the essential manpower requirements in industry went a step further in June 1990 to become National Apprenticeship and Industrial Authority (NAITA) under the Tertiary and Vocational Education Act No.20 of 1990. There are several national level training centers under the NAITA. Apprenticeship Training Institute, Technical Training Institute (TTI) and Automobile Engineering Training Institute are some of them. Main office of the NAITA is situated at Rajagiriya and it also decentralized the training function through 9 offices in 9 provinces.

#### **1.1.1 about Existing System**

The existing system carried out in the NAITA is totally manual system. When NAITA wants to recruit apprentices' .They publish about that through news papers, advertisement, etc. Applicants apply for the courses by register post or they handover applications to the office by themselves. According to the applicants qualifications they are selected to courses. Details about the apprentices are stored in documents by the clerk. Exam results of the apprentices are put on the office notice board. Some company recruit new employee through NAITA. They give details about vacancies to the NAITA .If there is no any suitable apprentices for the job under NAITA vacancies are put on notice board.. Vocational guidance is one of another service that is provided by the NAITA. If someone wants to get that service he or she must register by sending application to the NAITA.



There will be several disadvantages of this system. The main one is, mistakes can be heavily happened by the staff in entering, editing data and sometime they couldn't publish about vacancies and about services that provide by the NAITA. When applicants use register post in sending application to the NAITA. They couldn't send it before the closing date. This existing system is also very time consuming.

### **1.1.2 Web-Base Information Management system for the NAITA.**

NAITA publish its,

- News
- About vacancies
- About courses
- About services

Etc.

Through the news paper or by posters. But after developing the web site it can also use for that purpose. Internet is very good media to go through the people. By using internet NAITA can give good service to the Sri Lankan.

By using web base system applicants can apply for the courses through the internet. It can be done very efficiency and rapidly without any errors or mistakes. Also companies can publish there vacancies by using NAITA web site. For that company first send details to the NAITA by using web base system. Then if NAITA satisfy about the company and vacancies. It will publish by the NAITA by using web site.

Apprentices can use web site to get the exam result also. By entering the index number apprentices can get the exam result according to particular test.

### **1.3 Objectives:**

#### **1.3.1 Overall Objective:**

Develop a web site and web base Information management system for the National Apprentice and Industrial Training Authority.

#### **1.3.2 Specific Objectives:**

- Literature survey about the PHP and Mysql.
- Requirement analysis about details about NAITA and management process should be done.
- Develop a Database Management System for NAITA in Mysql.
- Designing and coding part of the system should be done by using HTML, PHP and JavaScript.

## Chapter 2

### Literature Review

#### 2.1 Web Database Application

An application is a program or a group of programs designed for use by an end user (for example, customers, members, circus acrobats, and so on). If the end user interacts with the application via a Web browser, the application is a Web-based or Web application. If the Web application requires the long term storage of information, using a database, it is a Web database application.

A Web database application is designed to help a user accomplish a task. It can be a simple application that displays information in a browser window (for example, it displays current job openings when the user selects a job title) or a complicated program with extended functionality.

##### 2.1.1 Web server

Web server is a net work server that manages access to files, folders and other resources over the Internet or a local Intranet via the platform-neutral HTTP. In addition, Web servers possess unique Web networking characteristic. They handle permissions, execute programs, keep track of directories and files and communicate with client computers. The web server accept HTTP requests from browsers like Internet Explorer, and Netscape Communicator and returns the appropriate HTML documents, images, etc. Microsoft Personal Web Server, Microsoft Internet Information Server and Apache are some of the popular Web Servers.

##### 2.1.2 Application Server

To process dynamic web pages, need an application server. An application server is software that helps a web server to process web pages containing server-side scripts or tags. When such a page is requested from the server, the web server hands the page off to the application server for processing before sending the page to the browser.

### **2.1.3 Internet Information Services**

Internet Information Services (IIS) makes it easy to publish information on the internet or intranet. IIS includes a broad range of administrative features for managing web sites and web server. With programmatic features like Active Server Pages (ASP), can create and deploy scalable, flexible web applications.

### **2.1.4 Dynamic Web page**

The Web pages that is created with HTML alone are static, meaning the user can't interact with the Web page. All users see the same Web page. Dynamic Web pages, on the other hand, allow the user to interact with the Web page. Different users might see different Web pages. For many years the Web survived as a domain where several simple HTML pages linked together constitute a site. But now, users expect existing pages that are updated frequently and provide a customized experience. At the same time, Web site administrator wants sites that are easier to update and maintain. For these reasons, building a site with just static HTML files is no longer acceptable. The Web is now place for dynamic, frequently database-driven, Web application.

To create dynamic Web pages, developer must use another language in addition to HTML. There are many technologies available for creating dynamic web sites.

### **2.1.5 Server Technologies for Web Applications**

The scripting or tag-based language used depends on the server technology. Here are the most popular languages for the five server technologies.

**Table 2.1** Most Popular Languages For Server Technologies

<b>Server Technology</b>	<b>Language</b>
ColdFusion	ColdFusion Markup Language (CFML)
ASP.NET	Visual Basic C#
Active Server Pages (ASP)	VBScript JavaScript
JavaServer Pages (JSP)	Java
PHP	PHP

## **2.2 PHP: Hyper Text Preprocessor (PHP).**

PHP originally stood for “Personal Home Page” as it was created in 1994 by Rasmus Lerdorf. As its usefulness and capabilities grew, it came to mean “PHP: Hyper Text Preprocessor.”

PHP is a server-side scripting language, which can be embedded in HTML. In other words, PHP pages are ordinary HTML pages that escape into PHP mode only when necessary. Or used as a standalone binary (although the former use is much more common).

PHP is mainly focused on server-side scripting, so can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies. But PHP can do much more.

There are three main areas where PHP scripts are used.

- Server-side scripting. This is the most traditional and main target field for PHP. Need three things to make this work. The PHP parser (CGI or server module), a web server and a web browser. Programmer needs to run the web server, with a

connected PHP installation. He can access the PHP program output with a web browser, viewing the PHP page through the server.

- Command line scripting. Can make a PHP script to run it without any server or browser. Only need the PHP parser to use it this way. This type of usage is ideal for scripts regularly executed using cron (on unix or Linux) or Task Scheduler (on Windows). These scripts can also be used for simple text processing tasks.
- Writing desktop applications. PHP is probably not the very best language to create a desktop application with a graphical user interface, but if programmer know PHP very well, and would like to use some advanced PHP features in his client-side applications he can also use PHP-GTK to write such programs. He also has the ability to write cross-platform applications this way. PHP-GTK is an extension to PHP, not available in the main distribution.

PHP can be used on all major operating systems, including Linux, many Unix variants (including HP-UX, Solaris and OpenBSD), Microsoft Windows, Mac OS X, RISC OS, and probably others. PHP has also support for most of the web servers today. This includes Apache, Microsoft Internet Information Server, Personal Web Server, Netscape and iPlanet servers, O'Reilly Website Pro server, Caudium, Xitami, OmniHTTPd, and many others. For the majority of the servers PHP has a module, for the others supporting the CGI standard, PHP can work as a CGI processor.

So with PHP, programmers have the freedom of choosing an operating system and a web server. Furthermore, he also has the choice of using procedural programming or object oriented programming, or a mixture of them. Although not every standard OOP feature is implemented in PHP 4, many code libraries and large applications (including the PEAR library) are written only using OOP code. PHP 5 fixes the OOP related weaknesses of PHP 4, and introduces a complete object model.

With PHP programmer is not limited to output HTML. PHP's abilities include outputting images, PDF files and even Flash movies (using libswf and Ming) generated on the fly. Programmer can also output easily any text, such as XHTML and any other XML file. PHP can auto generate these files, and save them in the file system, instead of printing it out, forming a server-side cache for your dynamic content.

One of the strongest and most significant features in PHP is its support for a wide range of databases. Writing a database-enabled web page is incredibly simple. The following databases are currently supported:

**Table 2.2** Databases Support by the PHP

Adabas D	InterBase	PostgreSQL
dBase	FrontBase	SQLite
Empress	mSQL	Solid
FilePro (read-only)	Direct MS-SQL	Sybase
Hyperwave	MySQL	Velocis
IBM DB2	ODBC	Unix dbm
Informix	Oracle (OCI7 and OCI8)	
Ingres	Ovrimos	

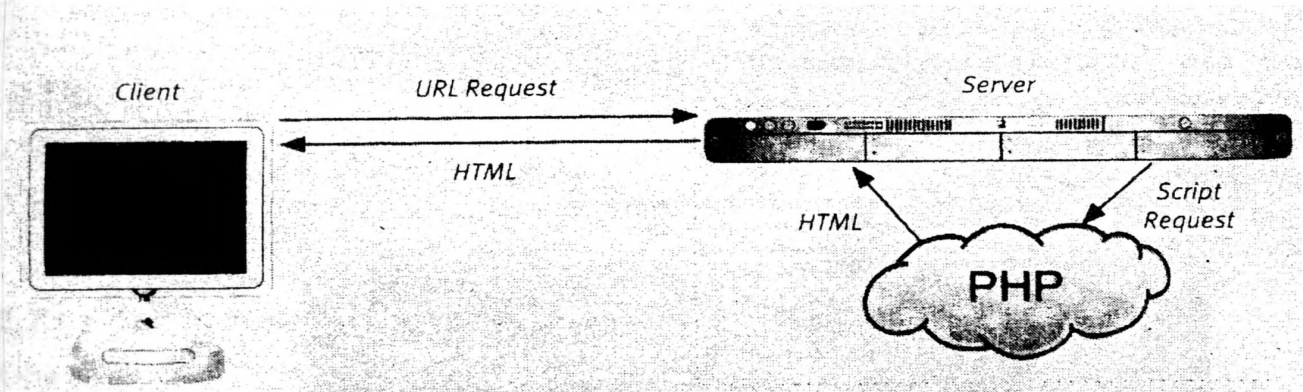
PHP also have a DBX database abstraction extension allowing programmer to transparently use any database supported by that extension. Additionally PHP supports ODBC, the Open Database Connection standard, so programmer can connect to any other database supporting this world standard.

PHP also has support for talking to other services using protocols such as LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (on Windows) and countless others. Programmer can also open raw network sockets and interact using any other protocol. PHP has support for the WDDX complex data exchange between virtually all Web programming languages. Talking about interconnection, PHP has support for instantiation of Java objects and using them transparently as PHP objects.

So when it comes developing dynamic web sites, PHP is better, faster, and easier to learn than the alternatives. What developer get with PHP is excellent performance , a tight integration with nearly every database available, stability, portability, and a nearly

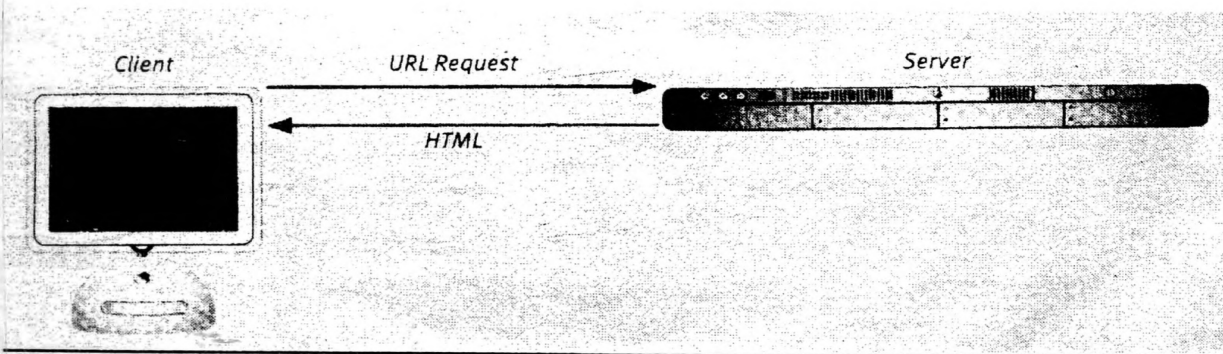
limitless feature set due to its extendibility All of this comes at no cost(PHP is open source) and with very manageable learning curve..

**How PHP works** - As previously stated, PHP is a server-side language, which means that the code write in PHP resides on a host computer that serves Web pages to the requesting visitors. When visitor goes to a Web site written in PHP, the server reads the PHP code and then processes it according to its scripted direction.



**Figure 2.1** How PHP fits into the client-server model when user request a Web page

This is differs from static HTML site where, when request is made, the server merely sends the HTML data to the Web browser and there is no server-side interpretation.



**Figure 2.2** The Process Involved With a Request For Static HTML Page



### Start and End Tags

In programming uses `<?php` as the PHP start tag and `?>` as the PHP end tag. The PHP interpreter ignores anything outside of those tags. Text before the start tag or after the end tag is printed with no interference from the PHP interpreter. (Larry Ulman, 2003)

#### Example:

```
<html>
<head> <title>PHP Test</title> </head>
<body>
<p>This is an HTML line<p>
<?php echo "This is a PHP line" ?>
</body></htm>
```

### 2.2.1 PHP in Object-Oriented Programming

Object-oriented programming is a way to group functions and data together into a prepackaged unit. This unit is known as an object.

#### 2.2.1.1 Using Objects

Typically, objects represent "real-world" or tangible entities, such as a person. Here is one version of a Person object in PHP:

```
$sunil = new Person;
$sunil->setName('sunil kumara');
print $sunil->getName( );
```

**sunil kumara**

The first line assigns a value to a \$sunil variable. This value is an object of type Person. Person is a previously specified structure containing code that describes how a "Person object" should operate. This structure is called a class.

For example, the `Person` class can look like this:

```
class Person {
    setName($name) {
        $this->name = $name;
    }

    getName() {
        return $this->name;
    }
}
```

The difference between an object and a class is that an object is an actual variable that you can manipulate. Developer can pass it to functions, delete it, copy it, and so forth. It holds a specific set of data. A class is the template that defines how the object can be used and what data it can hold.

Convert a class into an object by using the `new` keyword:

```
$sunil = new Person;
```

This command causes PHP to look for a class labeled as `Person`, create a new copy, and assign it to `$sunil`. This process is known as instantiating an object or creating

### 2.2.1.2 Autoload

When developer attempt to instantiate a class that's not defined, PHP 4 dies with a fatal error because it can't locate what developer is looking for. PHP 5 solves this problem by loading the missing code on the fly with its new autoload feature.

Extensive use of classes requires developer to either define all his classes in a single file or else place an include statement for each class he use at the top of every script. Since PHP 5 calls `__autoload ( )` when he instantiate undefined classes, he can make it include all the classes used by his script with only a little work:

```
function __autoload($class_name) {  
  
    include "$class_name.php";  
  
}
```

### **2.2.1.3 Data Encapsulation**

In PHP 4, all properties and methods are public. In PHP 5, however, developer can use the private label to restrict access to only those methods defined inside the class. When this label is applied to a method or property, it is known as private. Marking something as private signals that it may change in the future, so people shouldn't access it or they'll violate encapsulation.

This is more than a social convention. PHP 5 actually prevents people from calling a private method or reading a private property outside of the class. Therefore, from an external perspective, these methods and properties might as well not exist because there's no way to access them.

### **2.2.1.4 Constructors and Destructors**

Objects in PHP 5 also have the ability to call constructors and destructors. A constructor is a method that is called automatically when an object is instantiated. Depending upon how the constructor is implemented, developer may be able to pass it arguments.

For example, a constructor for a class that represents a database may take the address of the database developer wish to connect to, as well as the username and password necessary for authentication:

```
$db = new Database('db.mysql', 'naita', 'naitd');
```

PHP 4 has object constructors, but object destructors are new to PHP 5. Destructors are like constructors, except that they're called when the object is deleted. Even if developer doesn't delete the object yourself using unset( ), PHP 5 still calls the destructor when it determines that the object is no longer used. This may be when the script ends, but it can be much earlier.(O'Reilly, 2004)

## **2.3 Apache Web Server**

Of all the web servers on the market today, Apache is most popular because it supplies basic web server functionality right out of the box. Also Apache is currently the leading UNIX Web server. It is a high-performance httpd(HTTP daemon)server that has its root in UNIX. A daemon is a UNIX background process that implements the server side of a protocol. Also httpd is the program developer would run on a UNIX platform to launch a Web server. On other platforms, such as Microsoft Windows NT, the Web server is a background process implemented as a system service.

The Apache Web Server is a drop-in replacement for National Center for Supercomputing Applications HTTPd server. On a web server already running NCSA, one can simply compile Apache and replace the HTTPd binary from NCSA with the new apache binary. It is develop by the Apache Group and available free of charge. Apache comes in source form and can be compiled on many plat forms. Such as AIX, Linux, UNIX, and Solaris. There is also version of Apache for OS/2 and Windows NT. There is an abundant array of server- side programming tools and languages available for the Apache web server platform. Perl, PHP, Tcl and Python are some of them.

## **2.4 Databases**

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching, and replicating the data it holds. Other kinds of data stores can be used, such as files on the file system or large hash tables in memory, but when professionals talk about databases they mean a standalone application such as Oracle or SQL Server or Sleepycat.

### **2.4.1 Accessing a Database**

An application server lets work with server-side resources such as databases. For example, a dynamic page may instruct the application server to extract data from a database and insert it into the page's HTML.

The instruction to extract data from a database is called a database query. A query consists of search criteria expressed in a database language called SQL (Structured Query Language). The SQL query is written into the page's server-side scripts or tags.

An application server cannot communicate directly with a database because the database's proprietary format renders the data undecipherable in much the same way that a Word document opened in Notepad is undecipherable. The application server can communicate only through the intermediary of a database driver. A database driver is software that acts like an interpreter between the application server and the database. After the driver establishes communication, the query is executed against the database and a recordset is created. A recordset is a subset of data extracted from one or more tables in a database. The recordset is returned to the application server and the data used in the dynamic page.

#### **2.4.2 ODBC/JDBC versus native API**

There are two generic standard APIs for database access. Open Database Connectivity (ODBC), and Java Database Connectivity (JDBC). ODBC is closely associated with Microsoft, and JDBC is even more closely associated with Sun Microsystems. Nevertheless, other companies have implemented these standards in their own products, with the addition of specific drivers for each client program.

ODBC and JDBC are more or less mutually exclusive. Something called the *ODBC-JDBC bridge* is used to allow Java programs to access ODBC databases, but it is very slow. There are also proprietary drivers that do the same job more quickly.

There are also databases that clients can access through their own APIs rather than ODBC or JDBC. This is invariably faster because there are fewer layers in the stack. Most open source databases fall into this category. Some of these also have ODBC or JDBC drivers. So for instance, PHP can access MySQL with a native API, whereas a Java subsystem can use the same database via JDBC. Before you commit to any multiple-access scheme, be very sure the drivers you need are available, affordable, and maintainable.

**Table 2.3 Database Interfaces for Web Application**

Web application	Database interface	Common drivers
ColdFusion MX JSP	JDBC	Sun JDBC-ODBC driver i-net Sprinta JDBC driver for SQL Server Oracle Thin JDBC driver
ColdFusion 4 or 5	ODBC or OLE DB	ColdFusion native drivers Microsoft Access Driver Microsoft SQL Server Driver
ASP	ODBC or OLE DB	Microsoft Access Driver Microsoft SQL Server Driver Microsoft SQL Server Provider Microsoft ODBC for Oracle
ASP.NET	OLE DB	Microsoft Jet Provider Microsoft SQL Server Provider Microsoft Oracle provider
PHP	MySQL specific	MySQL driver

### 2.4.3 MySQL

MySQL is an open source application, like PHP and some variants of Unix meaning that it is free to use or even modify. MySQL is relational database management system. Like PHP MySQL offers excellent performance, portability, and reliability , with a moderate learning curve and little no cost.

Web Developer can build and interact with a MySQL database by using a few simple statements in the SQL language, which is the standard language for communicating with RDBMSs. And MySQL runs on a wide variety of operating systems Windows, Linux, Mac OS, most varieties of Unix (including Solaris, AIX, and DEC Unix), FreeBSD, OS/2, Irix, and others.

By incorporating a database in to a web application, some of the data generated by PHP can be retrieved from MySQL. This further moves the site's content from a static basis to a flexible one, flexibility being the key to a dynamic web site. (Janet Valade, 2004)

MySQL has been known to handle databases as large as 60,000 tables with more than 5 billion rows. And can work with table as large as 8 million tetra bytes on some operating system, generally a healthy 4 GB otherwise.

The MySQL, software consists of several pieces including the MySQL sever (mysqld, which runs and manage the databases), the MySQL client( mysql, which gives developer an interface to the server), and numerous utilities for maintenance and other purposes. The first logical use of SQL and MySQL will be to create database. The syntax for creating a new database is

```
CREATE DATABASE databasename;
```

Also ,

Simplest select query is

```
SELECT * FROM tablename;
```

#### **2.4.4 PHP with MySQL**

PHP's MySQL and MySQL Improved functions are easy to use, if sometimes named confusingly. Each instance of a PHP/MySQL interaction must have a connection, a database select, and a query or command that returns a result identifier. The result identifier is like an ATM receipt that reports on the success or failure of an operation.

If data is returned after a SELECT statement, one of the PHP/MySQL fetching functions must also be employed. Data pulled from a MySQL database exists in a kind of limbo until one of the fetching functions is applied to the result set. If developer wish to loop through the result set again, he can use `mysql_data_seek()` to reset the row pointer to zero.

PHP also has a large number of functions that return data about the database itself or about a particular operation. Two of the most common are `mysql_num_rows()`, which returns the number of rows in a result set; and `mysql_insert_id()`, which returns the ID of the proximate INSERT operation.

PHP handles much of the MySQL connectivity for developers without requiring specific link identifiers or result pointers. The exception comes when he need multiple database connections on the same Web page. In this case, developer can use exactly the same functions and syntax but simply pass the correct link identifier with most commands. Specific Web-server compatibility with MySQL is not required, since PHP will handle all the dirty work for developer.

There are several factors that make using PHP and MySQL together a natural choice:

**PHP and MySQL work well together** - PHP and MySQL have been developed with each other in mind, so they are easy to use together. The programming interfaces between them are logically paired up. Working together wasn't an afterthought when the developers created the PHP and MySQL interfaces.

**PHP and MySQL have open source power** - As they are both open source projects, PHP and MySQL can both be used for free. MySQL client libraries are no longer bundled with PHP. Advanced users have the ability to make changes to the source code, and therefore, change the way the language and programs work.

**PHP and MySQL have community support** - There are active communities on the Web in which developer can participate and they'll answer his questions. Developer can also purchase professional support for MySQL if he needs it.

**PHP and MySQL are fast** - Their simplicity and efficient design enables faster processing.

**PHP and MySQL don't bog developer down with unnecessary details** - developer doesn't need to know all of the low-level details of how the PHP language interfaces with the MySQL database, as there is a standard interface for calling MySQL procedures from PHP. (O'Reilly, 2006)



### 2.4.5 To connect and select a database

- Create a new PHP document in your text editor

```
<? Php mysql_connect.php
```

- Set the database host, username, password, and database name as constants.

```
define('DB_USER', 'username');
```

```
define('DB_PASSWORD', 'password');
```

```
define('DB_HOST', 'localhost');
```

```
define('DB_NAME', 'sitename');
```

- Connect to MySQL.

```
$abc = mysql_connect ( DB_HOST, DB_USER, DB_PASSWORD);
```

The `mysql_connect( )` function, if it successfully connects to MySQL, will return a resource link that corresponds to the open connection.

- Select the database to be used and close the PHP page.

```
Mysql_select_db(DB_NAME);
```

```
?>
```

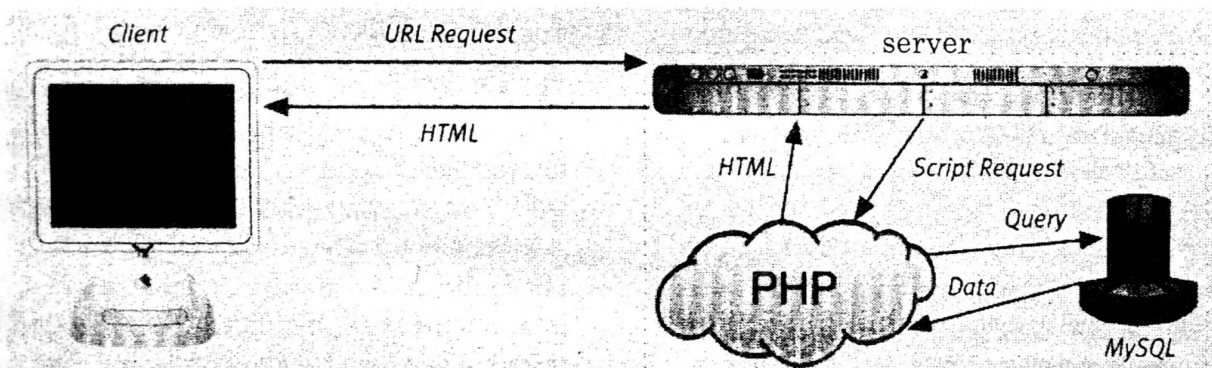


Figure 2.3 Dynamic web applications, using both PHP and MySQL

### 2.5 HTML

HTML stands for Hyper Text Markup Language, the standard for describing the contents and appearance of pages on the World Wide Web. HTML consists of pairs of opening

and closing tags, with attributes and values in between. The tags describe each element on a web page, such as a paragraph of text, a table, or an image.

## **2.6 Java script**

JavaScript is a dynamic scripting language that allows developer to build interactivity into otherwise static HTML pages. This is done by embedding blocks of JavaScript code almost anywhere in his Web page.

To make this work, blocks of JavaScript code are delineated by opening and closing Script tags:

```
<script language="JavaScript">  
JavaScript code goes here  
</script>
```

A scripting language developed by Netscape Communications and Sun Microsystems, Inc. Compared to Java, JavaScript is limited in performance because it is not compiled before execution. Basic online applications and functions can be added to web pages with JavaScript, but the number and complexity of available application programming interface functions are fewer than those available with Java. JavaScript code, which is included in a web page along with the HTML code, is generally considered easier to write than Java, especially for novice programmers. A JavaScript-compliant web browser, such as Microsoft Internet Explorer or Netscape Navigator, is required to interpret JavaScript code. (Arman Danesh, 2004)

## **2.7 Cascading Style Sheets (CSS)**

CSS is a breakthrough in Web design because it allows developers to control the style and layout of multiple Web pages all at once. Web developer can define a style for each HTML element and apply it to as many Web pages as developer wants. To make a global change, simply change the style, and all elements in the Web are updated automatically.

The CSS syntax is made up of three parts: a selector, a property and a value:

```
selector {property: value}
```

The selector is normally the HTML element/tag developer wish to define, the property is the attribute he wishes to change, and each property can take a value. The property and value are separated by a colon and surrounded by curly braces:

```
body {color: black}
```

Developer can group selectors. The class Selector and id selector

### 2.7.1 How to Insert a Style Sheet

When a browser reads a style sheet, it will format the document according to it. There are three ways of inserting a style sheet:

**External Style Sheet** - An external style sheet is ideal when the style is applied to many pages. With an external style sheet, developer can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section:

**Internal Style Sheet** - An internal style sheet should be used when a single document has a unique style. Developer defines internal styles in the head section by using the <style> tag, like.

**Inline Styles** - An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly, such as when a style is to be applied to a single occurrence of an element. To use inline styles you use the style attribute in the relevant tag. The style attribute can contain any CSS property.

## 2.8 Software Engineering

Software Engineering, this term distinguishes it on the one hand from unsystematic, amateur programming and, on the other, from computer science which is its theoretical basis. It is both disciplined and practical. Briefly, to produce business software, an analysis of the current manual system or the need for a new system can specify what needs to be done, a design can specify how this can be achieved and then programmers can implement the design to produce working software. With appropriate testing, the software can be installed and should do the job

*Software Engineering is the systematic approach to the development, operation, maintenance and retirement of software.*

### **2.8.1 Software Development Methods**

There are several methods available to develop the system. Following are the basic popular models used to this purpose.

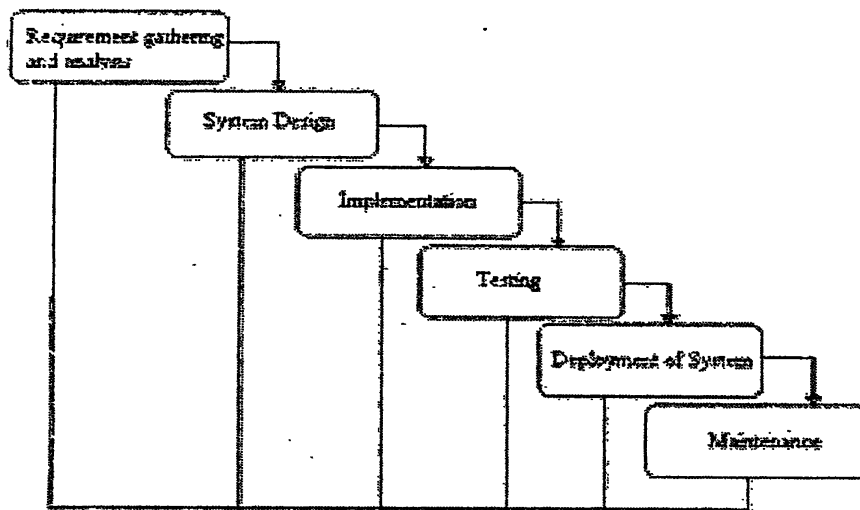
1. Linear Sequential Model (Waterfall Approach)
2. Prototyping Model
3. Rapid Application Development Model
4. Component Assembly Model
5. V-Model
6. Spiral Model

### **2.8.2. Linear Sequential Model (Waterfall Approach)**

One approach/process used in Software Development is "The Waterfall Model". Waterfall approach was first Process Model to be introduced and followed widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate process phases. The phases in Waterfall model are: Requirement Specifications phase, Software Design, Implementation and Testing & Maintenance. All these phases are cascaded to each other so that second phase is started as and when defined set of goals are achieved for first phase and it is signed off, so the name "Waterfall Model". All the methods and processes undertaken in Waterfall Model are more visible.

The Linear Sequential Model consists following phases,

1. Software Requirements Analysis
2. Systems Analysis and Design
3. Code Generation
4. Testing
5. Maintenance



**Figure 2.4** Waterfall Model

### 2.8.2.1 Software Requirements Analysis

In this phase, the development team visits the customer and studies their system. They investigate the need for possible software automation in the given system. This is also known as feasibility study. Feasibility study means assesses whether the solution is economically and technically practicable. This is an initial analysis, producing options with costs. In this phase produces a description of the problem. This involves describing the functions of the software needed, possible later extensions to it, the documentation needed, and performance requirements such as response time. It also includes the environment in which the solution must work (hardware, software, organization, users). The result of the analysis phase is a requirements specification document. It describes what is needed.

### 2.8.2.2 System Analysis and Design

In this phase produces some sort of model of a system satisfying the requirements. The required functions are decomposed into modules and their interfaces. The user interface is designed. Data structures are specified. In terms of the client/server technology, the number of tiers needed for the package architecture, the database design, the data structure designs etc. Design transforms the what of analysis into the *how* of a design

specification but they do not trespass into implementation details. The logical system of the product is developed in this phase. A software development model is created. Analysis and Design are very crucial in the whole development cycle. Any glitch in the design phase could be very expensive to solve in the later stage of the software development. Much care is taken during this phase.

### **2.8.2.3 Code Generation**

In this phase design must be translated into a machine-readable form. This phase also called Production or implementation involves creating software that works. The details of this will depend upon the construction tool being used, but there are some general principles. There may be a transition stage in which the logical design specification is transformed into a more detailed specification, such as using a high level language for processes. Production proceeds module by module. These are assembled into working software. If the design is performed in a detailed manner, code generation can be accomplished without much complication. Programming tools like Compilers, Interpreters, and Debuggers are used to generate the code. Different high-level programming languages like C, C++, VB.net, Java and different web development techniques such as Active Server Pages (ASP), Java Server Pages (JSP), PHP are used for coding. With respect to the type of application, the right programming language is chosen.

### **2.8.2.4 Testing**

Once the code is generated, the software program testing begins. Different testing methodologies are available to unravel the bugs that were committed during the previous phases. Different testing tools and methodologies are already available. Some companies build their own testing tools that are tailor made for their own development operations.

Testing has several aspects and does not happen only after production but throughout the development. Low-level testing and debugging occurs as each module is written. Tuning and optimization may be necessary once they are assembled. Verification checks that the product of coding is a correct translation of the design specification. (Just as the design

specification is verified against the requirements specification.) Validation checks that the software product is still fulfilling the user requirements.

#### **2.8.2.5 Maintenance**

This phase of "The Waterfall Model" is virtually never ending phase (Very long). Generally, problems with the system developed (which are not found during the development life cycle) come up after its practical use starts, so the issues related to the system are solved after deployment of the system. Not all the problems come in picture directly but they arise time to time and needs to be solved; hence this process is referred as Maintenance.

Software will definitely undergo change once it is delivered to the customer. There are many reasons for the change. Change could happen because of some unexpected input values into the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the post implementation period. The software may have undetected errors in it. In addition it will need adapting or improving over time. (For commercial software this can be more than half the total development effort.) (Sommerville, 1996)

The advantage of waterfall development is that it allows for departmentalization and managerial control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process like a car in a carwash, and theoretically, be delivered on time. Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order, without any overlapping or iterative steps.

There are some disadvantages of the Waterfall Model.

- As it is very important to gather all possible requirements during the Requirement Gathering and Analysis phase in order to properly design the system, not all requirements are received at once, the requirements from customer goes on getting added to the list even after the end of "Requirement Gathering and Analysis" phase, this affects the system development process

- The problems with one phase are never solved completely during that phase and in fact many problems regarding a particular phase arise after the phase is signed off, this result in badly structured system
- The project is not partitioned in phases in flexible way.
- As the requirements of the customer goes on getting added to the list, not all the requirements are fulfilled, this results in development of almost unusable system. These requirements are then met in never version of the system; this increases the cost of system development.



## **Chapter 3**

### **Methodology**

The requirements were well defined. So best software development process for the project was Linear Sequential modal (Waterfall approach). Therefore the Linear Sequential modal was used to develop the system.

Major Phases of the water fall approach

1. Requirements Analysis
2. Systems Analysis and Design
  - 2.1 Data and Database Design
  - 2.2 Interface and User Interface Design
3. Coding and Implementation
4. Testing
  - 4.1 Unit Testing
  - 4.2 System Testing

#### **3.1 Requirements Analysis**

First major phase of the development process was requirement analysis. Requirements were captured for design a web page and develop web base apprentices' management system for NAITA. Functional and non- functional requirement were analyzed during requirement analysis phase .To assesses whether the solution is economically and technically practicable feasibility study was done as an initial analysis, producing options with costs.

##### **3.1.1 Requirement Capture**

There are several organizations and offices in island wide under the NAITA. Details were acquired same of them. Several techniques were used in here. Interviewed some people used as one of those techniques.

- Interviewed people who were join with NAITA. Like Administration, Clerical Staff, Inspectors, Instructors, other staff members and apprentices.
- Interviewed people who are not join with NAITA. Like who wish to join with NAITA, Villages, School leavers, pupils and etc..

As another technique currently working documents and reports were also collected to get some details. Also requirements for the system were identified by conducting Formal meetings with the External supervisor.

Following table shows how the system was required by the NAITA with different users.

**Table 3.1** User Activities Table

Potential User	Activities performed
Normal users	<ul style="list-style-type: none"> <li>▪ To use get details about NAITA and its services</li> <li>▪ Online registration</li> <li>▪ To get exam result</li> <li>▪ Send applications for the vacancies, for the tests by using web base system.</li> </ul>
Administration	<ul style="list-style-type: none"> <li>▪ Update the web page</li> <li>▪ Apprentice management</li> <li>▪ Perform some task like get applications for the vacancies, for the tests by using web base system.</li> </ul>

### 3.1.2 Functional Requirements Analysis

Functional requirements describe what a system does or are expected to do. From the above requirement capture section; users can use the systems to perform activities that provided values to him or her to enhance the provision of such values. Those are the requirements of the project. Other wards this describes the functionalities have to provide by the system. They were statements of services the system should provide how the system should react to particular inputs and how the system should behave in particular situation.

- The user should be provided with facility to have some links to the pages with in Get details about NAITA. (History, Structure, services, etc..) Latest news about the NAITA
- The user should be provided facilities to online registration send application for the tests and for the vacancies.
- The user should be provided facilities to get exam results.
- Administrator must have permission for enter to the Administration module
  - The user should be provided with facility to enter a user ID and a password.
  - The user should be provided with facility to change user ID and a password.
- Administrator should be provided facilities to update web page.
- Administrator should be provided facilities to Apprentice management.

### **3.1.2 Non-functional requirement**

Non-functional requirements describe what qualities the system should have usability, performance, security, privacy and maintainability were considered.

#### **Usability**

The design of the user interfaces should be intuitive and simple. Web page must be more attractive.

#### **Performance**

The system should reduce the time as well as reducing the errors in entry. The system should have consistent performance at all times for all users and maintain performance.

#### **Security**

General users should not be allowed to access Administration module. Only administrator has the right to access Administration module.

## Privacy

Some of information should not be shown to the general users of the system and only authorized users should be able to view this information.

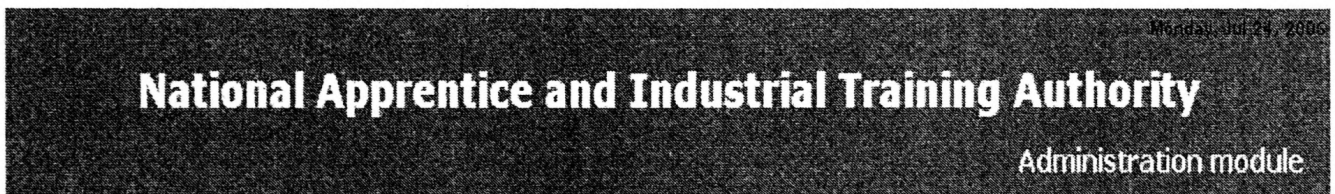
## Maintainability

Component based architecture is used in the system to simplify maintenance and allow for its expansion.

## 3.2 System Analysis and Design

The main purpose of this phase was produced some sort of model of a system satisfying the requirements. The required functions were decomposed into modules and their interfaces. The user interface was designed.

According to the user requirements Interfaces and User Interfaces were designed using relevant tools. Interfaces were built using scripting languages such as JavaScript as well as user interfaces were built using Hyper Text Markup Language (HTML) and Cascading Style Sheets (CSS).



Copyright© 2006 NAITA | Design & Developed by B. T. G. S. Kumara

**Figure 3.1** Designs for the Login Page to the Administration Module

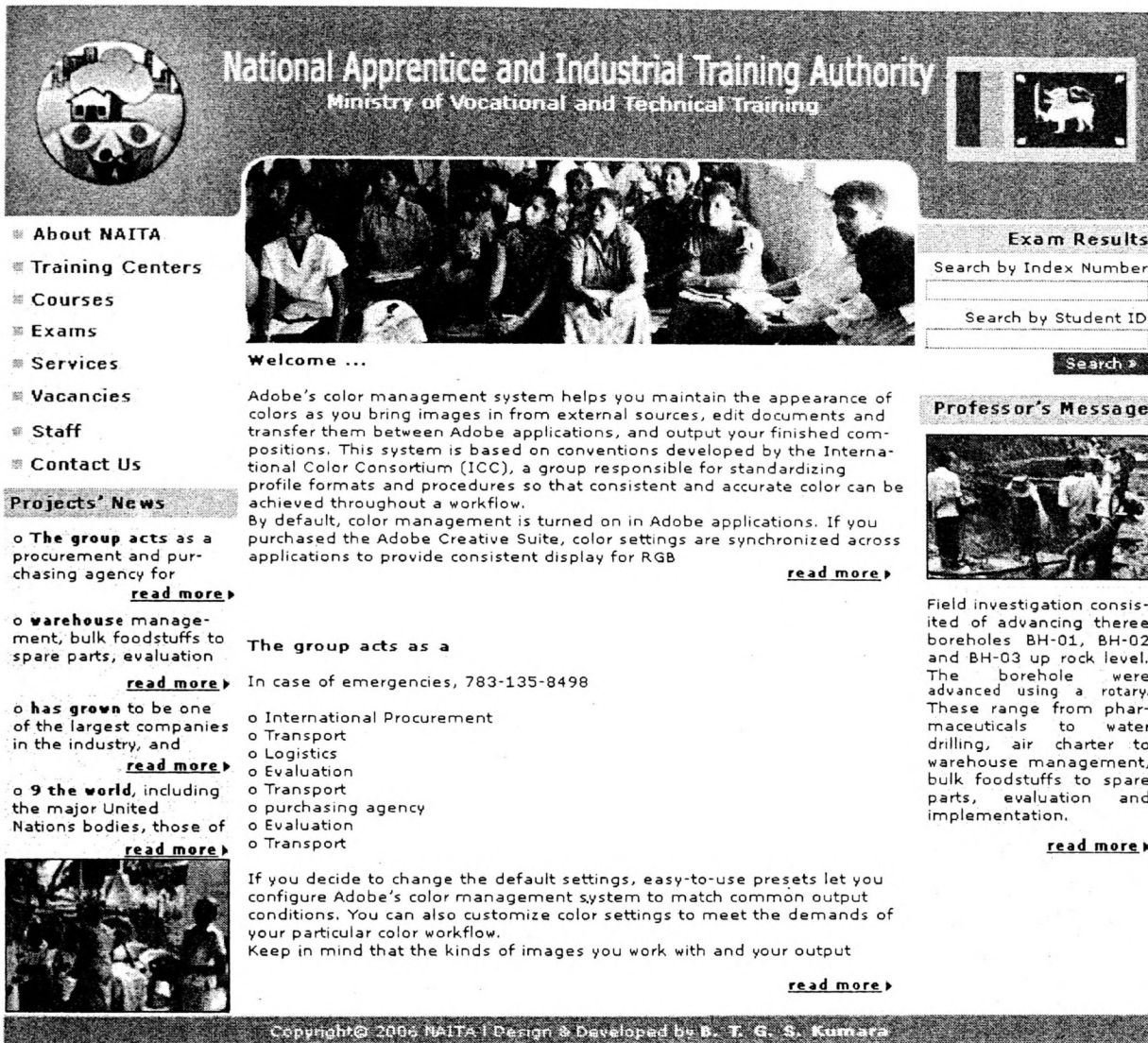


Figure 3.4 Design of the web page

After requirement analysis entities, relationships and attributes were identified. The Database and tables were created using normalization as well as keys were specified for relevant tables.

### 3.3 Coding and Implementation

Main purpose of this phase was creating software that works. There was a transition stage in which the logical design specification was transformed into a more detailed specification, such as using a high level language for processes. Production proceeded module by module. These were assembled into working software.

Coding was carried out using Hyper Text markup Language as the interface designing language and PHP was used to create dynamic pages. Structured Query Language (SQL) utilized as the tool for establishing the connection between the database and the respective interface.

### 3.3.1 Environment Used

The system was implemented using sever side scripting technology called PHP. JavaScript as the scripting language and manipulate (Feed, Retrieve, store) data using Structured Query Language (SQL)). The database was implemented using MySQL. User Interfaces, links and images were implemented using Macromedia Flash, Photoshop 7, and Dreamweaver. Pages were created as xxx.htm and xxx.php extensions. Those php and html pages were saved in C:\project\nita\. This is the place where php pages or dynamic web pages run.

When coding, PHP give freedom to select which kind of technique to be used. In here mixture of procedural programming and object oriented programming was used. Because some function were not used in more than once. That mean those function did not need to reuse. Reusability is a one of object oriented concepts. So by using object oriented programming technique for implement this function was not gives any advantage.

But sometime reusability was come forward for some functions. Like that situation object oriented programming was very helpful. In implementation, data base connection was used several times. So class mySQL was implemented for data base connection.

```
<?
class mySQL{
    var $last_error;
    var $last_erronumber;
    var $dbServer;
    var $dbLogonName;
    var $dbPassword;
    var $dbDatabase;
```

```

//constructor function
function mySQL(){
    switch($_ENV['OS']){
        case "Windows_NT":
            $this->dbServer = "localhost";
            $this->dbLogonName = "";
            $this->dbPassword = "";
            $this->dbDatabase = "nita";
            break;
        default:
            $this->dbServer = "mysql4.samagi.com";
            $this->dbLogonName = "sunil_Glob1";
            $this->dbPassword = "Glob1";
            $this->dbDatabase = "Sunil_tasteglobe";
            break;
    }
    $last_error="";
    $last_errornumber=0;
}

function getLastError(){
    return $this->last_error;
}

function getLastErrornumber(){
    return $this->last_errornumber;
}

function halt(){
    echo('<hr>');
    echo('ERROR ('. $this->last_errornumber. ');');
}

```

```

    echo(' '.$this->last_error);
    echo('<hr>');
    exit;
}

function execute($sql,&$result){
    $result = false;

    $con    =    mysql_connect($this->dbServer,$this->dbLogonName,$this-
>dbPassword);
    if(!$con){
        $this->last_error = mysql_error();
        $this->last_erronumber = mysql_errno();
        $this->halt();
    }

    $db = mysql_select_db($this->dbDatabase,$con);
    if(!$db){
        $this->last_error = mysql_error();
        $this->last_erronumber = mysql_errno();
        $this->halt();
    }

    $result = mysql_query($sql,$con);
    if(!$result){
        $this->last_error = mysql_error();
        $this->last_erronumber = mysql_errno();
        $this->halt();
    }

    mysql_close($con);
}

```



```

$this->last_error = 'Execution was sucessfull';
$this->last_erronumber = 0;
return true;
}

```

```

function execute_with_id($sql,&$result,&$insert_id){
    $result = false;

    $con    =    mysql_connect($this->dbServer,$this->dbLogonName,$this-
>dbPassword);
    if(!$con){
        $this->last_error = mysql_error();
        $this->last_erronumber = mysql_errno();
        $this->halt();
    }

    $db = mysql_select_db($this->dbDatabase,$con);
    if(!$db){
        $this->last_error = mysql_error();
        $this->last_erronumber = mysql_errno();
        $this->halt();
    }

    $result = mysql_query($sql,$con);
    if(!$result){
        $this->last_error = mysql_error();
        $this->last_erronumber = mysql_errno();
        $this->halt();
    }
}

```

```
$insert_id = mysql_insert_id($con);

mysql_close($con);

$this->last_error = 'Execution was sucessfull';
$this->last_errornumber = 0;
return true;
}

}//:mySQL
?>
```

Class PHPMailer, Class smtp and Class hft\_image are another three classes which were implemented using PHP.

MySQL was used for implement the data base. Courses, director\_message, nita\_hilights, students, student\_exam\_result , vacancies and system tables were created.

Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null	Unsigned	Zero Fill	Unique
✓	courses_id	INTEGER	11	0			✓		✓			
	trade_no	VARCHAR	20	0								
	trade	VARCHAR	20	0								
	period_training	VARCHAR	20	0								

Indices

Name	Fields	Unique	Collation	Full Text
PRIMARY	courses_id	✓	Ascending	

Options

TransactSafe	TableType	Row Format	Check Sum	Delay Key Write	Pack Keys	Temporary	Min Rows	Max Rows	Union
	MyISAM	DYNAMIC					0	0	

Definition

```
CREATE TABLE `courses` (
  `courses_id` int(11) NOT NULL auto_increment,
  `trade_no` varchar(20) default NULL,
  `trade` varchar(20) default NULL,
  `period_training` varchar(20) default NULL,
  PRIMARY KEY (`courses_id`)
) TYPE=MyISAM;
```

Figure 3.3 courses table

Following source code shows how the database connection was implemented for the login users to the administrator module.

```
<?
session_start();
header("Cache-control: private");

require_once("../lib/class.mysql.php");

$user_name = $_REQUEST['login'];
$password = $_REQUEST['password'];

$db = new mySQL;
```

```

if($user_name !=" and $password !="){
    $sql = "SELECT * FROM system WHERE user_name = '$user_name' AND
password = '$password'";
    $db->execute($sql, $result);

    if(mysql_num_rows($result) !="){
        $_SESSION['isAdmin'] = true;
        header('Location: news_list.php');
        exit;
    }else {
        header('Location: index.php?err=1');
        exit;
    }
}
?>

```

Cascading Style Sheet was played very important role in coding. It was very easy to implement system by using CSS. Following source code shows how the CSS was implemented in project.

```

body {
    background-color:#FFFFFF;
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
    background-image: url(images/bg.jpg);
}

.left_links {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 11px;

```

```
        color: #334170;
        font-weight: bold;
    }
a.left_links:link {
    color: #334170;
    text-decoration: none;
}
a.left_links:visited {
    text-decoration: none;
    color: #334170;
}
a.left_links:hover {
    text-decoration: underline;
    color: #786c54;
}
a.left_links:active {
    text-decoration: none;
    color: #786c54;
}

.body_text {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 10px;
    font-weight:normal;
    color:#000000;
}

.body_11_text_bold {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 11px;
    font-weight:bold;
}
```

```
        color:#000000;
    }
    .body_text_link {
        font-family: Verdana, Arial, Helvetica, sans-serif;
        font-size: 10px;
        font-weight: bold;
        color:#000000;
    }

    a.body_text_link:link {
        color: #000000;
        text-decoration: underline;
    }
    a.body_text_link:visited {
        text-decoration: underline;
        color: #000000;
    }
    a.body_text_link:hover {
        text-decoration: none;
        color: #000000;
    }
    a.body_text_link:active {
        text-decoration: none;
        color: #000000;
    }
    .footer {
        font-family: Verdana, Arial, Helvetica, sans-serif;
        color: #FFFFFF;
        font-size:10px;
        font-weight:normal;
    }
```

```
.form_text {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 10px;
    font-style: normal;
    color: #000000 ;
    border: #cccccc;
    border-style: solid;
    border-top-width: 1px;
    border-right-width: 1px;
    border-bottom-width: 1px;
    border-left-width: 1px
}
```

### 3.3.2 Configuration of Apache web Server

- First Apache web server was installed.
- Drive paths were changed to original path in the httpd.conf
- Virtual host was changed as follows

```
<VirtualHost 127.0.0.1>
    ServerName nita.com
    DocumentRoot c:\Projects\nita.com
</VirtualHost>
```

## 3.4 Testing

### 3.4.1 Unit Testing

At this stage each interfaces were tested individually through the Apache web server. All php and html pages were tested individually according to a test plan. In some cases there were used some dummy components for the testing process.

### 3.4.2 System Testing

Before testing the whole system Apache web server was configured and tested for a sample php pages. Then the integration of php and html pages was done according to the required manner.

- Client computer's web browser was used to check the system.
- Universal Resource Locator (URL): <http://local.nita.com> of the web server.
- After entering above URL checked whether default html page load or not. Then the system was checked according to the test plan.



## **Chapter 4**

### **RESULTS & DISCUSSION**

The out come of this project was attractive web page and user friendly organizational web based management information system to handle information related to the Operational Management of NAITA. This system meets all the user requirements and overcome all the draw backs of existing manual system. Implementation of this system would important in such away, the system would enhance the efficiency, Very reliably handled data (Existing system was a manual file system), Less man works, Handling data through the less cost of money. So that client was satisfied with the functionalities, usability, Security and Reliability of the system.

Literature survey about the PHP and MySQL was done in throughout this project. Requirement analysis of the details about NAITA and management process also done by using various techniques. By using MySQL. The database was implemented for the NAITA. According to the water fall approach which was the software development model for the development process, designing and coding part were done. So specific objectives of the project were successfully achieved.

## **Chapter 05**

### **Conclusion**

Since the major objective of developing a Web Based Management Information System had been achieved in the view of the stakeholders, the project work can be regarded as up to the customer expectation and satisfaction. However continues monitoring must be done in order to check whether the system meets its goal in the long run.

Future consideration for this system is to enhance some of the present features and add new features for the system.

## References

- Andrew Stellman and Jennifer Greene (2005). Applied Software Project Management . Cambridge, MA: O'Reilly Media. ISBN 0596009488.
- Arman Danesh.(2004) JavaScript TM in 10 Simple Steps or Less , 1<sup>st</sup> Edition, Wiley Publishing, Inc.,647p.
- DotCOM & Service Provider Solutions.(2000) Linux Web Solution with Apache, PHP, MySQL, and ht://Dig Technical Guide, 1<sup>st</sup> Edition, Compaq Computer Corporation.p.28.
- Eliens, A. (2000) Principle of Object-Oriented Software Development, 2<sup>nd</sup> Edition, Pearson Education Limited 2000.
- Janet, V. (2004) PHP and MySQL for Dummies , 2<sup>nd</sup> Edition , Wiley Publishing, Inc., 459p.
- Larry, U. (2003) PHP AND MySQL FOR DYNAMIC WEB SITES , 1<sup>st</sup> Indian Edition , Pearson Education, USA. 572p.
- Lerdorf , R. (2002)PHP Pocket Reference , 2<sup>nd</sup> Edition , O'Reilly Media, Inc. 82p.
- Michele, E. D and Jon, A. P. (2006) Learning PHP and MySQL , 1<sup>st</sup> Edition , O'Reilly Media, Inc., 812p.
- NileshParekh,(2005),The Waterfall model Explained, <http://www.buzzle.com/editorials/1-5-2005-63768.asp>.
- Powell,T.A.(2000) The Complete Reference Web Design. Tata McGraw-Hill publishing Company Limited, 831p.
- Sklar , D.(2004) Learning PHP 5 , 1<sup>st</sup> Edition, O'Reilly Media, Inc. 661p.
- Sommervill,I.(1995) Software Engineering, 5<sup>th</sup> Edition, Addison Wesley Publishers in autumn, 732p.
- Trachtenberg , A. ( 2004) Upgrading to PHP 5 , 1<sup>st</sup> Edition , O'Reilly Media, Inc. 581p.
- Tim Converse, Joyce Park and Clark Morgan.(2004) PHP5 and MySQL Bible, 4<sup>th</sup> Edition, Wiley Publishing, Inc.,1083p.
- Zend technologies, (2005) Zend PHP Certification Study Guide , 1<sup>st</sup> Edition, Sams Publishing , 264p.

# Appendix

## Student Management Interface in Administration Module

Monday, Jul 24, 2016

**National Apprentice and Industrial Training Authority**  
Administration module

\*\*\*\*\* Menu List \*\*\*\*\*

- » NAITA highlights
- » Director message
- » **Student Management**
- » Courses
- » Exams Schedule
- » Exams Result
- » Vacancies
- » System
- » Log off

Student Edit/Update

Name in Full:

Date of Birth:  -day-  -month-  -year-

Gender:  Male

National ID:

Address:

District:

Divisional Secretariat:

Samurdhi/Low income family?  Yes

I passed my G.C.E(A/L) :

G.C.E(A/L) Result :	subject passed	Grade
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

I passed my G.C.E(O/L) :

G.C.E(O/L) Result :	subject passed	Grade
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

Indicate 3 Trades In Order Of Preference:  Select a course

# Interface of the NAITA Web Page

Monday, Aug 07, 2006



## National Apprentices and Industrial Training Authority

Ministry of Vocational and Technical Training



- About NAITA
- Training Centers
- Courses
- Exams Schedule
- Services
- Vacancies
- Staff
- Contact Us
- Online Registration

### Welcome ...

The National Apprenticeship and Industrial Training Authority (NAITA) previously called the National Apprenticeship Board has perform a vital function over the years in selecting and training thousand of youth in industrial and business undertaking throughout the Sri Lanka. Considering the very small number of enterprises in Sri Lanka which have facilities for in-house training, the activities of NAITA perform a vital function in providing training of personnel for industry and business.

The scheme of training adopted by NAITA has in essence the German system of dual training whereby practical on the job experience is gained in the office and the factory and theoretical instruction is given at technical institutes.

### NAITA highlights

- my self

[read more »](#)

- Stored procedures are a new feature in MySQL version 5.0. A stored procedure is a set of SQL commands that can be stored in the server.

[read more »](#)

- Relational databases store all their data in tables. A table is a data structure consisting of an unordered set of horizontal rows.

[read more »](#)

- Table Editor is the basic SQL Manager 2005 for MySQL tool for working with tables.

[read more »](#)



The training programmes offered by NAITA could be divided in to two areas. The Formal training or 'off-the- Job-Training' such as basic training is given at certain technical institutions sponsored by NAITA like the Technician Training Institute, Katunayake and the Apprenticeship Training Institute, Moratuwa. The Informal Training or 'On-the-job-Training' such as industrial training is provided at certain selected organizations like the Colombo Dockyard(Pte0 Limited and Sri Lanka Ports Authority

[read more »](#)

### NAITA in Sabaragamuwa province

The centrally controlled apprenticeship training was decentralized in keeping with the decentralization policy of the government and this resulted in the establishment of the Sabaragamuwa Province Office of NAITA in Ratnapura On 14 th November 1989.The office exceeded the target recruitment of 500 for 900 and 1182 recruited for apprenticeship training in that year . In 1991 further 712 apprentices were recruited.

The objective of the Sabaragamuwa Province Office is to organize and manage the apprenticeship training programmes of the two district Ratnapura and Kegalle. A Apprentices are under going training in different establishment of the Sabaragamuwa province and their training is monitored by this office regularly.

The Sabaragamuwa province office of NAITA continues to carter to the manpower needs of this area by training skilled manpower. It is not an exaggeration to say that sabaragamuwa province of NAITA is a valuable asset to the youth of this area who are prepared to obtain training for gainful employment.

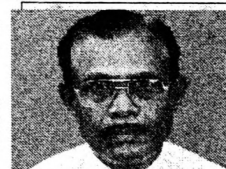
### Exam Results

Search by Index Number

Search by Reg. ID

Search »

### Provincial Director's Message



POSIX threads support. Apache can now run in a hybrid multiprocess, multithreaded mode. This improves scalability for many, but not all configurations.

[read more »](#)

## Administration Module

Monday, Jul 26, 2006

# National Apprentice and Industrial Training Authority

Administration module

\*\*\*\*\* Menu List \*\*\*\*\*

Student Management

- NAITA highlights
- Director message
- Student Management
- Courses
- Exams Schedule
- Exams Result
- Vacancies
- System
- Log off

Add a new student record >

ID	Student Name	Course	Student Data	Naita Results
1	Geethani Lakmali	Monotype Keyboard	[Edit]	[Edit]
2	Nisantha kumara	Primary Book Binder	[Edit]	[Edit]
3	Ramya Wasantha	Plumber	[Edit]	[Edit]
4	Samitha Udayanga	Compositor(Hand)	[Edit]	[Edit]
5	Sujith Susantha	Monotype Keyboard	[Edit]	[Edit]
6	Tharanga Pitigala	Diesel Pump Room Mec	[Edit]	[Edit]
7	Wasantha Kumara	Radio Mechanic	[Edit]	[Edit]

## login page to Administration Module

NAITA - Ministry of Vocational and Technical Training - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://local.nita.com/moderator/index.php

Monday, Jul 26, 2006

# National Apprentice and Industrial Training Authority

Administration module

Logon Name:

Password:

Copyright© 2006 NAITA | Design & Developed by B. T. G. S. Kumara

## Data arrangement in courses table

The screenshot shows a database management tool window titled "Table - [courses] - [nita on localhost]". The interface is divided into several sections:

- Object:** Shows the current database "nita on localhost [nita]" and the selected table "courses".
- General:** Contains options like Refresh, Table properties, Grants on table, Print, Show SQL help, Table Editor options, and Restore default size.
- Tools:** Includes a Truncate table button.
- Data Management:** Offers actions such as Commit transaction, Rollback transaction, Export data, Export as SQL Script, Import data, Load data, and Save data to file on server.
- Explorer:** Shows the current view as "Fields (4)" with the field "courses\_id [INT]".

The main data grid is displayed in "Grid View" and contains the following data:

courses_id	trade_no	trade	period_training
1	IN-01-01	Marine Filter	3
2	IN-01-02	Automobile Mechanic	4
3	IN-01-04	Heavy Duty Mechanic	4
4	IN-01-05	Diesel Engine Mechan	2.5
5	IN-01-06	Diesel Pump Room Mec	2.5
6	IN-01-07	Motor Cycle Mechanic	2
7	IN-02-01	Mason	4
8	IN-02-02	Mason(Building)	3
9	IN-02-03	Plumber	3
10	IN-03-01	Electrical Power	3
11	IN-03-02	Electrical Wireman	3
12	IN-03-03	Armature Winder	3
13	IN-03-06	Automobile Electrici	3
14	IN-03-07	Lift Mechanic	4
15	IN-03-08	Refrigeration & Air	3.5
16	IN-03-09	Air Conditioning Mec	2.5
17	IN-03-10	Refrigeration Mechan	2.5
18	IN-03-11	Refrigeration & Air	2
19	IN-03-12	Air Conditioning Ser	1
20	IN-03-13	Refrigeration Servic	1
21	IN-03-14	Vehicle Air Conditio	1
22	IN-04-01	Radio Mechanic	3
23	IN-04-01	Radio Mechanic	3
24	IN-04-02	Television Mechanic	2.5
25	IN-04-03	Electrical Equipment	1
26	IN-04-07	Telecommunication Li	2

The status bar at the bottom of the window shows "Records fetched: 61/61", "Open Time: 61 ms", and "LIMIT 0, 1000".

## Fields in students table

Table - [students] - [nita on localhost]

Object: nita on localhost [nita] > students

General

- Refresh
- Table properties
- Grants on table
- Print
- Show SQL help
- Table Editor options
- Restore default size

Tools

- Truncate table

Fields

- Add new field
- Edit selected field
- Drop selected field(s)

Explorer

Fields (12)

- student\_id [INT(11)]
- course\_id [INT(11)]
- name\_full [VARCHA
- date\_birth [DATE]
- gender [VARCHAR(
- ...

Field Name	Field Type	Size	Precision	Not Null	Default
student_id	INTEGER	11	0	<input checked="" type="checkbox"/>	Null
course_id	INTEGER	11	0	<input type="checkbox"/>	Null
name_full	VARCHAR	50	0	<input type="checkbox"/>	Null
date_birth	DATE	0	0	<input type="checkbox"/>	Null
gender	VARCHAR	10	0	<input type="checkbox"/>	Null
nic_no	VARCHAR	15	0	<input type="checkbox"/>	Null
address	TEXT	0	0	<input type="checkbox"/>	Null
disrict	VARCHAR	20	0	<input type="checkbox"/>	Null
dsd	VARCHAR	20	0	<input type="checkbox"/>	Null
low_income	CHAR	1	0	<input checked="" type="checkbox"/>	Y
is_al	CHAR	1	0	<input checked="" type="checkbox"/>	A
is_of	CHAR	1	0	<input checked="" type="checkbox"/>	0



## director\_message table

### Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null	Unsigned	Zero Fill	Unique
✓	message_id	INTEGER	1	0		0			✓			
	name	VARCHAR	255	0								
	degree	VARCHAR	20	0								
	telephone	VARCHAR	20	0								
	address	TEXT	0	0								
	message_short	VARCHAR	255	0								
	message_long	TEXT	0	0								

### Indices

Name	Fields	Unique	Collation	Full Text
PRIMARY	message_id	✓	Ascending	

### Options

TransactSafe	TableType	Row Format	Check Sum	Delay Key Write	Pack Keys	Temporary	Min Rows	Max Rows	Union
	MyISAM	DYNAMIC					0	0	

### Definition

```
CREATE TABLE `director_message` (
  `message_id` int(1) NOT NULL default '0',
  `name` varchar(255) default NULL,
  `degree` varchar(20) default NULL,
  `telephone` varchar(20) default NULL,
  `address` text,
  `message_short` varchar(255) default NULL,
  `message_long` text,
  PRIMARY KEY (`message_id`)
) TYPE=MyISAM;
```

## student\_exam\_results table

### Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null	Unsigned	Zero Fill	Unique
✓	record_id	INTEGER	11	0			✓		✓			
	student_id	INTEGER	11	0								
	subject	VARCHAR	20	0								
	grade	CHAR	1	0								
	exam_type	VARCHAR	20	0								

### Indices

Name	Fields	Unique	Collation	Full Text
PRIMARY	record_id	✓	Ascending	

### Options

TransactSafe	TableType	Row Format	Check Sum	Delay Key Write	Pack Keys	Temporary	Min Rows	Max Rows	Union
	MyISAM	DYNAMIC					0	0	

### Definition

```
CREATE TABLE `student_exam_result` (
  `record_id` int(11) NOT NULL auto_increment,
  `student_id` int(11) default NULL,
  `subject` varchar(20) default NULL,
  `grade` char(1) default NULL,
  `exam_type` varchar(20) default NULL,
  PRIMARY KEY (`record_id`)
) TYPE=MyISAM;
```

## students table

### Descriptions

There is no description for table students

### Fields

PK	Name	Data type	Size	Precision	Values	Default	Auto Increment	Binary	Not null	Unsigned	Zero Fill	Unique
✓	student_id	INTEGER	11	0			✓		✓			
	course_id	INTEGER	11	0								
	name_full	VARCHAR	50	0								
	date_birth	DATE	0	0								
	gender	VARCHAR	10	0								
	nic_no	VARCHAR	15	0								
	address	TEXT	0	0								
	disrict	VARCHAR	20	0								
	dsd	VARCHAR	20	0								
	low_income	CHAR	1	0		Y			✓			
	is_al	CHAR	1	0		A			✓			
	is_ol	CHAR	1	0		O			✓			

### Indices

Name	Fields	Unique	Collation	Full Text
PRIMARY	student_id	✓	Ascending	

### Options

TransactSafe	TableType	Row Format	Check Sum	Delay Key Write	Pack Keys	Temporary	Min Rows	Max Rows	Union
	MyISAM	DYNAMIC					0	0	

### Definition

```
CREATE TABLE `students` (
  `student_id` int(11) NOT NULL auto_increment,
  `course_id` int(11) default NULL,
  `name_full` varchar(50) default NULL,
  `date_birth` date default NULL,
  `gender` varchar(10) default NULL,
  `nic_no` varchar(15) default NULL,
  `address` text,
  `disrict` varchar(20) default NULL,
  `dsd` varchar(20) default NULL,
  `low_income` char(1) NOT NULL default 'Y',
  `is_al` char(1) NOT NULL default 'A',
  `is_ol` char(1) NOT NULL default 'O',
  PRIMARY KEY (`student_id`)
) TYPE=MyISAM;
```

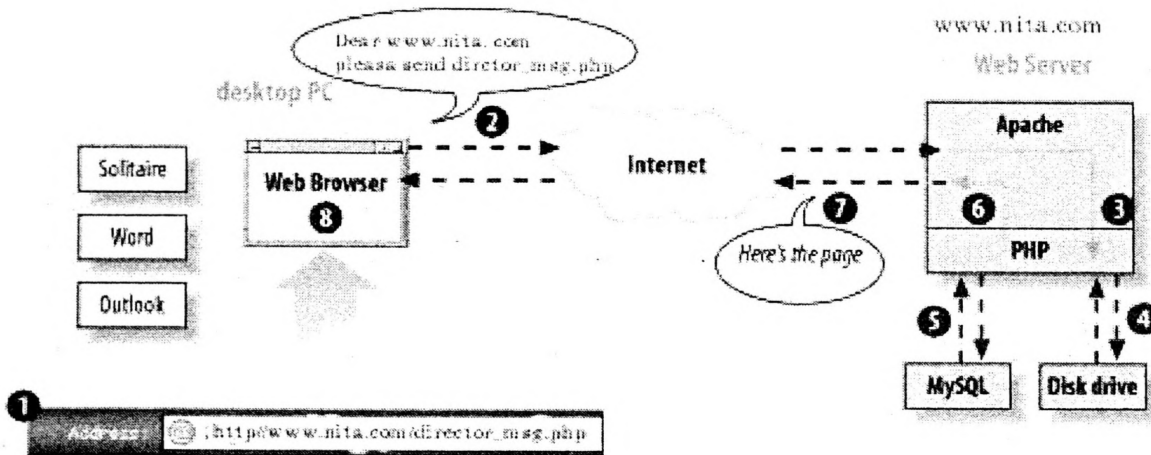
## Download PDF file for the vacancies

The screenshot shows a Microsoft Internet Explorer browser window. The address bar contains the URL: `http://local.nita.com/moderator/vacancy_reply_read.php?menu=6&vacancy_reply_id=16`. The page title is "NAITA - Ministry of Vocational and Technical Training - Microsoft Internet Explorer". The main content area displays the "National Apprenticeship and Industrial Training Authority Administration module" header. A "Menu List" is visible on the left, with "Vacancies" selected. A "File Download" dialog box is open in the foreground, displaying the following information:

- Vacancy Title: tikiri Pitigala
- File name: reply\_5\_16.pdf
- File type: System
- From: local.nita.com

The dialog box asks: "Would you like to open the file or save it to your computer?" and provides buttons for "Open", "Save", "Cancel", and "More Info". The "Always ask before opening this type of file" checkbox is checked. The browser's status bar at the bottom shows the current page URL: `http://local.nita.com/moderator/download.php?file=reply_5_16.pdf` and the "Internet" icon.

## Client and Server Communication with PHP



1. Type `www.nita.com/director_msg.php` into the location bar of Internet Explorer.
2. Internet Explorer sends a message over the Internet to the computer named `www.nita.com` asking for the `/director_msg.php` page.
3. Apache, a program running on the `www.nita.com` computer, gets the message and asks the PHP interpreter, another program running on the `www.nita.com` computer, "What does `/director_msg.php` look like?"
4. The PHP interpreter reads the file from the disk drive.
5. The PHP interpreter runs the commands in `director_msg.php`, possibly exchanging data with a database program such as MySQL.
6. The PHP interpreter takes the `director?msg.php` program output and sends it back to Apache as an answer to "What does `/director_msg.php` look like?"
7. Apache sends the page contents it got from the PHP interpreter back to client computer over the Internet in response to Internet Explorer's request.
8. Internet Explorer displays the page on the screen.

**National Digitization Project**

***National Science Foundation***

Institute : Sabaragamuwa University of Sri Lanka

1. Place of Scanning : Sabaragamuwa University of Sri Lanka, Belihuloya

2. Date Scanned : ..2017..09..25.....

3. Name of Digitizing Company : Sanje (Private) Ltd, No 435/16, Kottawa Rd,  
Hokandara North, Arangala, Hokandara

4. Scanning Officer

Name : ..B.A.C. Badarayanan.....

Signature : .......

Certification of Scanning

*I hereby certify that the scanning of this document was carried out under my supervision, according to the norms and standards of digital scanning accurately, also keeping with the originality of the original document to be accepted in a court of law.*

Certifying Officer

Designation : ..Librarian.....

Name : ..T. N. Neighsoorei.....

Signature : .......

Date : ..2017..09..25.....

**Mrs. T. N. NEIGHSOOREI**  
(MSSc, PhD, ASLA, BA)  
Librarian  
Sabaragamuwa University of Sri Lanka  
P.O. Box 02 Belihuloya, Sri Lanka  
Tele: 094 45 2280045  
Fax: 094 45 2280045

*"This document/publication was digitized under National Digitization Project of the National Science Foundation, Sri Lanka"*