DESIGN AND IMPLEMENTATION OF A WEB BASED HUMAN RESOURCE MANAGEMENT SYSTEM FOR SABON-GARI LOCAL GOVERNMENT SECRETARIAT (saboHRM)

BY

SANNI NAFISAT

A PROJECT SUBMITTED TO THE DEPARTMENT OF MATHEMATICS
AHMADU BELLO UNIVERSITY ZARIA, NIGERIA
IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF DEGREE OF BACHELOR OF SCIENCES (B.Sc. Hons.) IN COMPUTER SCIENCE

SEPTEMBER, 2016

ABSTRACT

The goal of this project was to design and develop a human resource management system for Sabon-Gari Local Government Secretariat. This system was developed to automate the management of employees. Development work included construction of the core system framework, creation and management of employee accounts, design of employee profiles and an integrated hiring process and leave management. The tools used in the implementation of the system are PHP, MySQL, HTML, CSS, JQuery and Bootstrap.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

1.1 Background of the study

Human resource management (HRM), also called personnel management (Douglas, 1992), consists of all the activities undertaken by an organization to ensure the effective utilization of employees toward the attainment of individual, group, and organizational goals. An organization's HRM function focuses on the people side of management. It consists of practices that help the organization to deal effectively with its people during the various phases of the employment cycle, including pre-hire, staffing, and post-hire. The pre-hire phase involves planning practices. The organization must decide what types of job openings will exist in the upcoming period and determine the necessary qualifications for performing these jobs. During the hire phase, the organization selects its employees. Selection practices include recruiting applicants, assessing their qualifications, and ultimately selecting those who are deemed to be the most qualified. In the post-hire phase, the organization develops HRM practices for effectively managing people once they have come through the door. These practices are designed to maximize the performance and satisfaction levels of employees by providing them with the necessary knowledge and skills to perform their jobs.

A HRMS (Human Resource Management System) is a combination of systems and processes that connect human resource management and information technology through human resource software (Zafar, 2013). A HRMS may help to revolutionize a workplace. The automation of repetitive and time consuming tasks associated with human resources management frees up some of the organization’s most valuable employees and allows the focus to shift to culture, retention, and other highly impactful areas.

Selection of a HRMS to handle HR activities is a trademark of a modern company, there are few successful companies that do not have some sort of automation in place for HR tasks at this juncture. Web accessibility has further worked to transform the landscape of HR, putting information and task management at the fingertips of employees and managers. HRMS has helped to effectively break down bureaucracy and "flatten" many organizations.

Security is of great concern when it comes to choosing a human resources management system (Chasserio & Legault, 2009). The information stored in a HRMS is highly sensitive, including proprietary organization data and volumes of personal information about employees. It is essential for companies to choose a solution that utilizes a method of secure transmission such as SSL which encrypts the data as it transmits over the internet.

Internal security is also critical, information should be guarded by passwords that have varying levels of access in relation to what is needed for the job position. While most organizations now allow employees to access portions of HRMS solutions, employees must understand the importance of maintaining the integrity of the system and protecting the security of the information it contains (i.e. no password sharing). Safeguards should also be in place to quickly bar terminated employees' access to systems.

1.2 Statement of Problem

Even the smallest of organizations need to implement some form of digitized information system to meet up current technological improvements. Organizations generally have a choice between using a computerized or manual HRM. Manual system in organizations offers a number of potential problems which includes:

1.2.1 Administrative Overwhelm

Many organizations, especially Sabon Gari LGA, still carry out their HRM manually. A manual HRMS typically requires a great deal of paperwork. A manual process also creates an administrative burden for HR staff of the organization.

1.2.2 Organizational Issues

A related problem uncovered during requirement gathering in Sabon Gari LGA include, not keeping all important HRM records and information in an organized, effective and accurate manner. This problem is simply a matter of work flow, as keeping files and documents in a central location is not actualized in Sabon Gari LGA.

1.2.3 Financial Implication

For most organizations, a paper-based system is costly and insecure. A paper-based system for managing documents and information in Sabon Gari LGA is expensive because the organization must allocate financial resources for both the cost of the materials and the labor power of managing them, including printing, collating and filing costs. This could mean employing additional workers who require salaries and benefits.

Similarly, paper-based systems make it difficult to share documents. When an organization office is distributed nationally or internationally, the ability to send and share documents quickly is key necessity. With HRMS, this becomes much easier and faster.

1.2.4 Motivation of the Study

Over the past decade, virtually every major industry invested heavily in computerization and this has pushed organizations to change from manual to digitized technique of information management. Yet, despite these advances in our society, majority of organizations handles HRM

using paper based techniques, and very few employers of labor are able to perform recruitment without speaking to real human resource personnel.

The primary motivation of this work is the common transition from paper based information system to computerized information system by various organizations due to advances in technology.

During the information gathering phase of this work, the earlier mentioned motivation of interest turned solid as the need for specific features in Sabon Gari LGA Secretariat showed up; these features include:

i. The need for an improved data management: An HRMS can manage all employee data as well as information on benefits, such as enrollment and status changes
ii. The need for employee self- service: An HRMS enables employees to update personal data without having to involve HR for simple tasks. This frees up the time of HR professionals for more strategic functions.
iii. The need for a central storage: Because data is stored in one place, this means reporting can be more efficient. It makes access of compliance records simple. It also means there is a central location for documents such as employee handbooks, procedures and safety guidelines.
iv. The need for on-demand reporting: Customized executive and management reporting can be done on the fly. For example, a manager might want to look up employment data by location, and could do so without having to contact HR. Managers can access the information timely as it pertains to employee development, performance improvement, and wage detail (as appropriate).

v. The need for managing and tracking recruitment data: Many HRMS systems include recruitment components such as applicant and resume management.

1.3 Aim and Objective of the Study

This study is aimed at developing a web based HRMS for Sabon Gari Local Government Area Secretariat (SG-LGA Secretariat)

The objectives of this study are to

i. Identify HRMS required modules.
ii. Build system architecture incorporating selected modules in (1).
iii. Design HRMS work flow.
iv. Design a central database incorporating selected modules in (1).
v. Implement designs in 2,3,4 using software tools namely: Toad Data Modeller 5.4, PHP 5.6 & MySQL 5.6, Netbean IDE 8.2

1.4 Scope of the study

There is an excess of modules that can be added to HRMS which include payroll management, personnel recruitment, leave management, tracking absenteeism, performance evaluation etc. This work is scoped to the development of recruitment and employee personal information management (PIM) modules powered with appropriate reporting capabilities.

1.5 Research Methodology

The following are the proposed steps needed for the realization of this research work.

i. Review literatures to comprehend HRMS.
ii. Requirement gathering and analysis
iii. System Design

iv. Implementation

v. Testing

vi. Deployment

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction to HRMS

Human Resource Management System (HRMS) lays at the center of any computerized personnel department in any organization. This type of system would require a computer program that captures data of employees such as: employee number, salaries wages information, skills and qualifications etc. This would enable employee information to be readily available at all time. Employees are the backbone of any company therefore their management plays a major role in deciding the success of the company. HMRS makes it easy for the employer to keep track of records. The application provides a means of acquiring, storing, analyzing and distributing information. The HR functions consist of tracking employee data which traditionally includes personal histories, skill, accomplishments etc. To reduce the workload of the administrative activities, organizations began to electronically automate many of these processes by introducing specialized Human resource management systems.

The application may have more than one administrator based on the structure of the company using it. If there is more than one department, each department head might be administrators. The workers will have access to only information related to their department unless their departments work together with other departments.

2.2 History and Practices of HRM in Sabon Gari local government area (SLGA)

SLGA was created from Soba local government area in September 1990, during the reign of General Muhammad Babangida. Soba was also deployed from Zaria local government earlier. Sabo contains 11 political wards which include: Muchia, Agwan-gabas, Chikagi, Jushi, Zabi,

Dogarawa, Basawa, Hanwa, Samaru, Bomo and Jama'a. There are seven departments in Sabon Gari LGA secretariat which include:

i. Administration and General Service department,
ii. Finance and supply department,
iii. Works Housing Transport Lands and Survey department,
iv. Budget Planning Research and Statistics department,
v. Social development Education Information Sports and Culture department,
vi. Agricultural and Natural resources department and
vii. Primary Health Care department.

A local government secretariat is a public service institution that helps to support local government to deliver value for money services (Victor, 2013). In Sabo local government secretariat, the technique used to manage employees is done manually. Most of the records including employee records are categorized in files and stored in office cabinets. Each department has a head that controls any activity going on in the department. The Administration and General Service department is the pivot on which the other departments operate. Personnel Recruitment data submission and assessment are done manually. If information reaches an interested person, he will submit his Application letter, CV and other important credentials to the organization and wait to be called for interview.

Any request made or information brought by the community goes to the Chairman of the local government in the Administrative and General Service department. Confirmations will be made to make sure the request is genuine before being forwarded to the required department for the appropriate action to be taken towards it.

The manual system of management employed in the local government secretariat poses a lot of inefficiencies and challenges. Today the rate at which technology grows is unprecedented the management of employees using manual system has been replaced by using automated systems. Manual handling of employees poses a number of challenges. This is evident in procedures such as leave management where an employee is required to fill in a form which may take several weeks or months to be approved. The use of paperwork in handling some of these processes may lead to human error, papers might end up in the wrong hands and not forgetting the fact that it is time consuming. Keeping records in files exposes it to lack of physical security such as stealing, fire outbreak, misplacement of file, and manipulation of records. Some of these records are employee qualifications, request letters from community members, application letters etc. loss of such vital information could have great effects on the organization. Years of keeping records in cabinets leads to accumulation of too many files and records. Records become stressful to manage because the accessibility of a particular file or record becomes difficult and time consuming. The presence of all these problems in an organization facilitates low growth rate, because the workforce of the organization will be quite high but very inefficient.

This system should consist of an application program, on one hand, and a database (repository of data) on the other. The program should perform the basic operations on the database as retrieving, inserting, updating and deleting data. Any additional functionality is a goal of a further module development. It is a kind of strategy to start the development from designing and constructing the database, as this structure will determine the further structure of the application program. The following are some major features:

i. It collects and keeps all the necessary information (such as name, password, entry date, exit date, house number, etc.) of each employee.

ii. Data are much more secured using this system, since backups and passwords are applied to the application.

iii. It can easily be used and does not require technical computer or accounting knowledge.

2.3 History of the Internet

In computing, a web application or web app is a client–server software application in which the client (or user interface) runs in a web browser (Nations, 2016). In earlier computing models like client–server, the processing load for the application was shared between code on the server and code installed on each client locally. In other words, an application had its own pre-compiled client program which served as its user interface and had to be separately installed on each user's personal computer. An upgrade to the server-side code of the application would typically also require an upgrade to the client-side code installed on each user workstation, adding to the support cost and decreasing productivity. In addition, both the client and server components of the application were usually tightly bound to a particular computer architecture and operating system and porting them to others was often prohibitively expensive for all but the largest applications. Today, of course, native apps for mobile devices are also hobbled by some or all of the foregoing issues (Fraker & Mazza, 2011).

In contrast, web applications use web documents written in a standard format such as HTML and JavaScript, which are supported by a variety of web browsers. Web applications can be considered as a specific variant of client–server software where the client software is downloaded to the client machine when visiting the relevant web page, using standard procedures such as HTTP. Client web software updates may happen each time the web page is visited. During the session, the web browser interprets and displays the pages, and acts as the universal client for any web application.

In the early days of the Web each individual web page was delivered to the client as a static document, but the sequence of pages could still provide an interactive experience, as user input was returned through web form elements embedded in the page markup. However, every significant change to the web page required a round trip back to the server to refresh the entire page (Nations, 2016).

In 1995 Netscape introduced a client-side scripting language called JavaScript allowing programmers to add some dynamic elements to the user interface that ran on the client side. So instead of sending data to the server in order to generate an entire web page, the embedded scripts of the downloaded page can perform various tasks such as input validation or showing/hiding parts of the page (Swartz, 2016).

In 1996, Macromedia introduced Flash, a vector animation player that could be added to browsers as a plug-in to embed animations on the web pages. It allowed the use of a scripting language to program interactions on the client side with no need to communicate with the server (Aune, 2010).

In 2005, the term Ajax was coined, and applications like Gmail started to make their client sides more and more interactive. A web page script is able to contact the server for storing/retrieving data without downloading an entire web page (Swartz, 2016).

In 2011, HTML5 was finalized, which provides graphic and multimedia capabilities without the need of client side plug-ins. HTML5 also enriched the semantic content of documents. The APIs and document object model (DOM) are no longer afterthoughts, but are fundamental parts of the HTML5 specification. WebGL API paved the way for advanced 3D graphics based on HTML5 canvas and JavaScript language. These have significant importance in creating truly platform and browser independent rich web applications (Brown, 2016).

2.4 History of WWW

The terms Internet and World Wide Web are sometimes used interchangeably, but there are important differences between the two. The Internet makes it possible to communicate via computers around the world. The Web makes that communication a straightforward and enjoyable activity. The Web is essentially a distributed information service and is based on a set of software applications. Although it is used effectively with the Internet, it is not inherently bound to it.

Distinct from the Internet, the World Wide Web refers to hypermedia using Hypertext Markup Language or HTML. This unique language allows information to be linked so when a person selects or clicks on one part of a link their browser automatically finds the designated information. Thus, the Web's unique characteristic is that it empowers the user to "click" on a word and be transported to a related web location. The development of this innovation is attributed to Tim Berners-Lee, a researcher at the CERN Institute of Geneva, Switzerland who is credited for the creation of the first links on the World Wide Web. In 1990 Sir Tim Berners Lee developed the first web browser and that was the only way to view the web at that time (Williams, 2015).

The web became more than just an interesting experiment in 1993 with the development of a graphical browser. Up until this point, even the World Wide Web was terminal-based, meaning that the user depended on the use of a keyboard. Marc Andreesen, who was working with a team at the National Center for Supercomputing Applications (NCSA) created the first graphical browser, one that was not terminal based, called Mosaic.

In 1994, during a strong period of innovation, the first cyber-age robbery occurred in Russia. The theft of millions of dollars from Citibank showed the world the ramifications of the misuse of this

new technology. It did not take long for the world to realize that crimes could be perpetrated on such as money- making scams and computer hacking.

By 1995, an estimated 50 million users were connected to the Internet worldwide ("Internet Live Stats", 2016). With the telecommunications act of 1996, the E-rate program came into existence with the goal of connecting millions of schools. Net-coalition was founded in 1999 to set standards for e-commerce, which was already blossoming. The business and media worlds were rocked in 2000 when Time- Warner and AOL announced their merger, making the marriage of the media industry and cyber space a reality. Today more than 3.4 billion users are connected to the Internet worldwide and the number is still on the rise ("Internet Live Stats", 2016).

2.5 Web Application

Web applications use web documents written in a standard format such as HTML and JavaScript, which are supported by a variety of web browsers. Web applications can be considered as a specific variant of client–server software where the client software is downloaded to the client machine when visiting the relevant web page, using standard procedures such as HTTP. Client web software updates may happen each time the web page is visited. During the session, the web browser interprets and displays the pages, and acts as the universal client for any web application.

In the early days of the Web each individual web page was delivered to the client as a static document, but the sequence of pages could still provide an interactive experience, as user input was returned through web form elements embedded in the page markup. However, every significant change to the web page required a round trip back to the server to refresh the entire page.

2.6 Client Side Technologies

Client side operations refer to operations that are performed by the client in a client server relationship in a computer network. Typically a client is a computer application such as a web browser, which runs on a user's local computer or workstation and connects to server as necessary.

2.7 Server Side Technologies

It is a technique used in web development which involves employing scripts on a web server which produce a response customized for each client request to the website. Scripts can be written in any number of server side scripting languages that are available. Server side scripting is often used to provide customized interface for the user, it also enables the source code the website owner to hide source codes that generate the interface.

2.8 Database System

A database is an organized collection of data, it is the collection of schemas, tables, queries, reports, views and other objects. A database management system is a computer software application that interacts with the user, other applications, and the database itself to capture and analyze data. A general purpose database management system is designed to allow the definition, creation, querying, update and administration of databases.

2.9 Review of related work

Syed *et al.* (2013) developed a human resource management system which was used to reduce the effort of Administrator to keep the daily events of workers in an organization, such as attendance, projects, works, appointments, etc.

Soumya (2007) designed and implemented a human resource management system for a fictitious consulting firm that participates in the placement of contractors in different organizations. The system was implemented using a 3-tier approach, with a backend database (MySQL database), a

middle tier of Microsoft Internet Information Services (IIS) and ASP.NET, and a front end web browser (client). Software consulting and recruiting agencies match the requirements of the client firms with the skills of employees and set up interview between their employees and the client firm. Interviews are then conducted, and the candidates selected in the interview are recruited as consultants in the client's firm for the duration of the project.

Kevin and Fiona (2009) designed a human resource management system for recruitment selection, personnel management and career guidance. This was used for the selection process used to select applicants as employees of an organization. It also keeps the records and data of every other employee working in the organization, thereby showing the progress of their career including promotions and demotions.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 Fact Finding Techniques

This involves a formal way of gathering important information and opinions about the organization. It enables the developer to learn about the terminologies, problems, opportunities, constraints, requirements and priorities of the organization. Some if these techniques include:

i. **Interview:** it is the most commonly used and usually the most useful fact finding technique. It enables collection of information from individuals face-to-face. There are two types the open-ended questions which allow the interviewee to respond in any way that seems appropriate and the Close-ended question which restrict answers to specific choices.

ii. **Observing the Organization in Operation:** is an effective way to understand a system, observation is made by watching or participating in activities. It is usually adopted when the complexity of an aspect of the system prevents a clear explanation by the end-users, it therefore enables the observer to see exactly what is being done.

iii. **Research:** information is gathered using computer trade journals, reference books, and the internet. It provides information on how others have solved similar problems whether software packages exist to solve or partially solve the problems.

3.2 System Design

All users are provided with the same login interface, the user must login with a valid username and password combination. Only the User Administrator has access to all information on the application, the interfaces that appears to a user could be different based on the category he falls into. The User Admin can add a new user by entering his/her basic information, a unique username

and password is assigned to the new user. The employee applies by making a request and forwarding it to the HR manager.

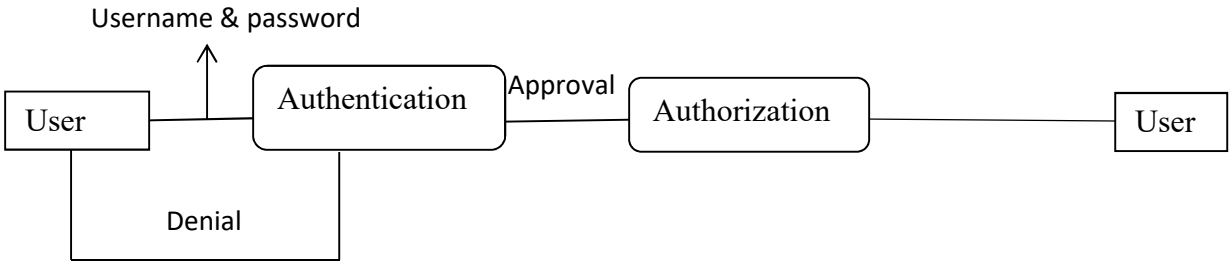

Figure 3.1: Authentication and Authorization Data Flow diagram

3.3 Database Design

Database design is the process of producing a detailed data model of database. This data contains all the logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database

The core of a Web database application is the database, which is the long term memory that stores information for the application. A database is an electronic file cabinet that stores information in an organized manner so that you can find it when you need it. After all, storing information is pointless if you can't find it. A database can be small, with a simple structure for example, a database containing the titles and authors' names of all the books that you own. Or a database can be huge, with an extremely complex structure such as the database that Amazon .com must have to hold all its information.

Over the years, as the demands for information have grown, so too have the database systems that have attempted to meet these demands. However, along with this evolution, we have seen an increase in the costs associated with storing data as well as an increase in the demand for products that can run on multiple platforms and can be optimized based on the needs of specific types of organizations. In response to this changing climate, MySQL has emerged as the most popular

open-source database management system (DBMS) in the world. Therefore, the database will be designed using MySQL. Consequently, organizations everywhere are jumping on the MySQL bandwagon, increasing the demand for those who know how to use MySQL to manage data and those who know how to create applications that can access data in MySQL databases. The following diagram shows processes that take place with a database system.

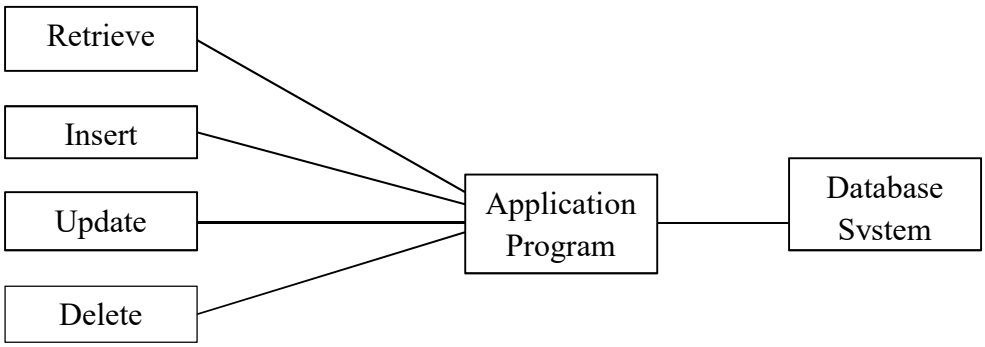

Figure 3.2: Database Workflow

3.3.1 Database Tables

A database table is another name for database relation, though the difference is that a table is usually a multi-set of rows where a relation is a set and does not allow duplicates. Besides the actual data rows, tables generally have associated with them some metadata, such as constraints on the table or on the values within particular columns. Below are the tables of each component of the system.

Name	Variable type	Constraint
Employee id	int	NN
Staff id	Varchar	NN
Firstname	Varchar	NN
Surname	Varchar	NN
DoB	Date	NN
Date of Employment	Date	
Address	Varchar	
Gender	Enum	NN
Phone no	Varchar	
Email	Varchar	
Nationality	Enum	NN
Title	Enum	NN

Figure 3.3: Employee Detail Table

Name	Variable type	Constraint
Recruitment id	int	NN
Firstname	Varchar	NN
Surname	Varchar	NN
Gender	Enum	NN
Mobile no	Varchar	NN
Email	Varchar	
CV	Varchar	
App Status	Enum	

Figure 3.4: Recruitment Detail Table

Name	Variable type	Constraint
Leave id	int	NN
Startdate	Date	NN
Enddate	Date	NN
Purpose of Leave	Varchar	NN
App date	Date	NN
Status	Enum	NN

Figure 3.5: Leave Management Table

3.4 ER Diagram (Entity Relationship Diagram)

An entity relationship diagram is a graphical representation of an information system that shows the relationship between people, objects, places, concepts, or events within that system. The ER diagram of the saboHRM is shown below.

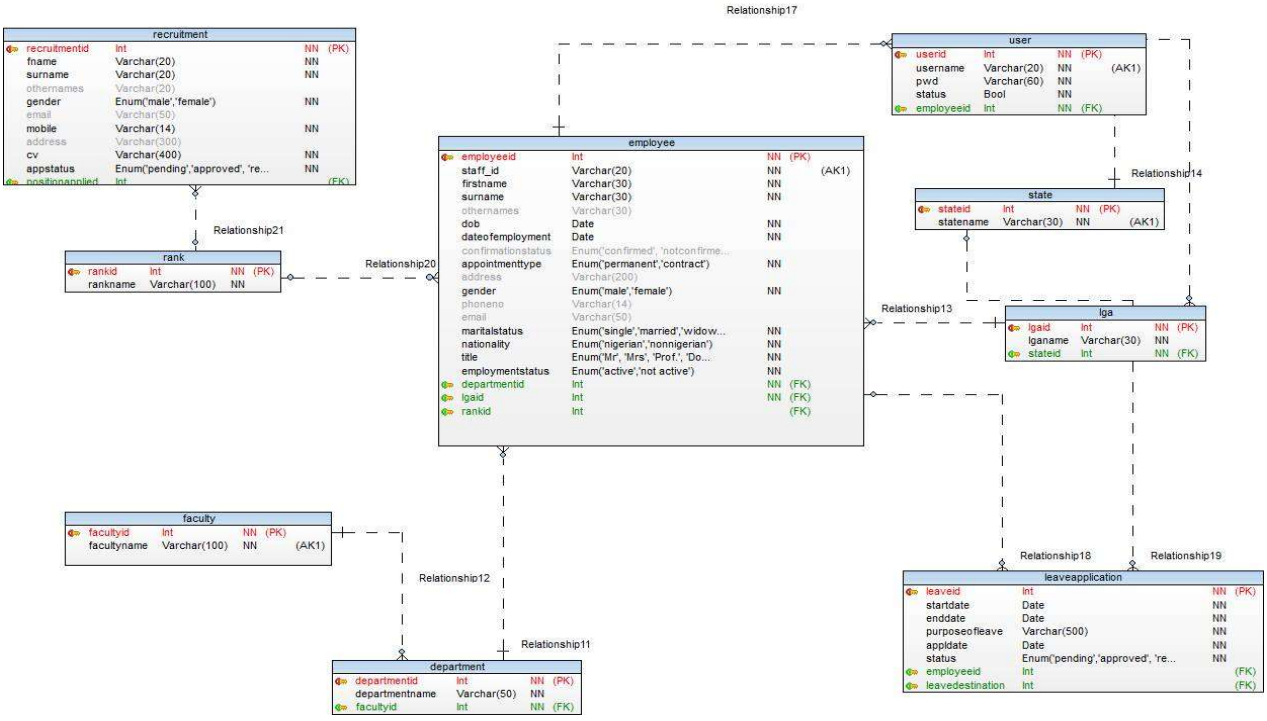

Figure 3.6: ER Diagram Of Sabohrm Schema

3.5 Development Tools Used

We explain the repertoire of tools used in the development of this web application. We categorize these tools as follows:

i. Programming Language

ii. Integrated Development environment

iii. Infrastructural Software

iv. Framework and Libraries

3.5.1 Programming Languages

i. Javascript

JavaScript is a high-level, dynamic and interpreted programming language. It has been standardized by ECMAScript language specification. It is a dynamic computer programming language and most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side programming, game development and the creation of desktop and mobile applications. JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the self and Scheme programming languages. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. The application of JavaScript to use outside of web pages for example, in PDF documents, site-specific browsers, and desktop widgets is also significant.

ii. PHP Language

PHP is a server-side scripting language designed for Web development. It runs on the server and generally cannot run on its own unless in response to an event. It is designed for use with HTML such that when invoked, it returns HTML to the browser. It provides more flexibility than HTML alone. It is platform independent, powerful, robust and scalable, web development specific. It can be object oriented, it has great documentation in different languages, and it has large active developer community.

Advantages of PHP

i. It is free and open source.

ii. It integrates well with Apache web server and MySQL RDBMS

iii. It comes with a lot of support.

iv. It is pretty fast

v. There is a lot of libraries available for use in PHP

vi. HTML

HyperText Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS), and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a webserver or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

vii. CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually

engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

3.5.2 Integrated Development Environments

i. Netbeans

NetBeans is a software development platform written in java. It allows applications to be developed from a set of modular software components called Modules. Applications based on the NetBeans platform, including the NetBeans Integrated Development Environment (IDE), can be extended by third party developers. The NetBeans IDE is primarily intended for development in java, but also supports other languages in particular PHP, C/C++ and HTML.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:

i. User Interface Management (e.g menus and bars)
ii. User settings management
iii. Storage management (saving and loading any kind of data)
iv. Window Management
v. Wizard Framework

vi. NetBeans Visual Library

ii. Notepad++

Notepad++ is a text editor and source code editor for use with Microsoft Windows. Unlike Notepad, the built-in Windows text editor, it supports tabbed editing, which allows working with multiple open files in a single window. Notepad++ is one of the most popular source code editors in the world, and supports syntax highlighting and code folding for over 50 programming, scripting, and markup languages.

iii. Toad Data Modeler

Toad Data Modeler is a database design tool allowing users to visually create, maintain and document new or existing database systems. Toad Data Modeler allows users to create logical, physical and universal entity relationship diagrams (ERD) - models. This tool supports various database specific items and therefore it is necessary to select a target database system for a physical ERD. Key features include: SQL/DDL code generation, Reverse engineering, Model documentation, Model verification, Model synchronization.

3.5.3 Infrastructural Software

i. WAMP:

WAMP is an archetypal model of web service solution stacks, named as an acronym of the names of its original four open-source components: the Windows operating system, the Apache HTTP Server, the MySQL relational database management system (RDBMS), and the PHP programming language.

ii. MySQL

MySQL is an open source database that is platform independent and can easily interface with other scripting language, but it works best with PHP. Deploying a MySQL database is cheap and easy, it doesn't require special hardware or software requirements. It can work on any web server.

Advantages of MySQL

i. It runs on many operating systems.
ii. It supports several development interfaces including scripting (PHP).
iii. It manages memory very well and prevents memory leak.
iv. It is easy to use.
v. It is fast
vi. It includes data security layers that protect sensitive data from intruders, therefore secure to use.

3.5.4 Frameworks and Library

i. Twitter Bootstrap

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Bootstrap provides a set of style sheets that provide basic style definitions for all key HTML components. These provide a uniform, modern appearance for formatting text, tables and form elements. Bootstrap comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

ii. *Jquery Core Library*

jQuery, at its core, is a DOM (Document Object Model) manipulation library. The DOM is a tree-structure representation of all the elements of a Web page and jQuery simplifies the syntax for finding, selecting, and manipulating these DOM elements. For example, jQuery can be used for finding an element in the document with a certain property (e.g. all elements with an h1 tag), changing one or more of its attributes (e.g. color, visibility), or making it respond to an event (e.g. a mouse click).

jQuery also provides a paradigm for event handling that goes beyond basic DOM element selection and manipulation. The event assignment and the event callback function definition are done in a single step in a single location in the code. jQuery also aims to incorporate other highly used JavaScript functionality (e.g. fade ins and fade outs when hiding elements, animations by manipulating CSS properties).

iii. *AdminLTE UI Template*

It is a popular open source Web application template for admin dashboards and control panels. It is a responsive HTML template that is based on the CSS framework Bootstrap. It utilizes all of the Bootstrap components in its design and re-styles many commonly used plugins to create a consistent design that can be used as a user interface for backend applications. AdminLTE is based on a modular design, which allows it to be easily customized and built upon. This documentation will guide you through installing the template and exploring the various components that are bundled with the template.

3.6 Software and Hardware requirements

SaboHRM is a web application that runs both on client and server. As a result the application will benefit from the advantages of web application in general. Below is the specification for both the client and server perspectives.

3.6.1 Client System Requirements

The application will run on any client machine as long as there is browser installed in it and availability of network connection. Base on the client side technology used for development, the application will run best on modern browsers like: Opera, Google Chrome, Microsoft Edge, Firefox and Safari.

3.6.2 Server-side System Requirements

The technology used for the development of the server side components includes PHP, MySQL and Apache Web server. As a result, the server machine must be installed with PHP 5.6 interpreter, MySQL 5.0 Database server and Apache web server 2.0. The application will require a server machine with at least 1GB RAM, 1.5GHz processor and sufficient storage space on the hard drive.

CHAPTER FOUR

SYSTEM IMPLEMENTATION AND TESTING

4.1 System Implementation

System implementation is the process of defining how an information system should be built ensuring that the information system is operational and the information system meets quality standard. This application requires the following tool for its implementation, PHP, MySQL, Netbeans, HTML etc. This tools are used together to generate the codes that is run on the WAMP which results in the application system. The application is tested if certain functionalities are not well implemented, it will be required to check the codes and make corrections or changes to it.

saboHRM is a web based application that feature both admin and user end. The application is made up of four modules: Personnel Information Management (PIM), Leave, Recruitment, and Setting. Below is the result of our implementation of saboHRM.

4.2 Interface Design

Every page in the application follows a uniform layout. Every page contains a fixed header, and a fixed sidebar to the left that can be collapsed explicitly (or automatically in mobile view). The header contains the name and logo of the application, name of the currently logged in user and control for collapsing and expanding the side navigation bar. The side navigation bar contains links to all the modules in the application.

Since the application is designed with bootstrap UI framework, the interface is mobile friendly allowing for better mobile user experience. The figure below shows a simulated mobile view of

the web application on iPhone 6.

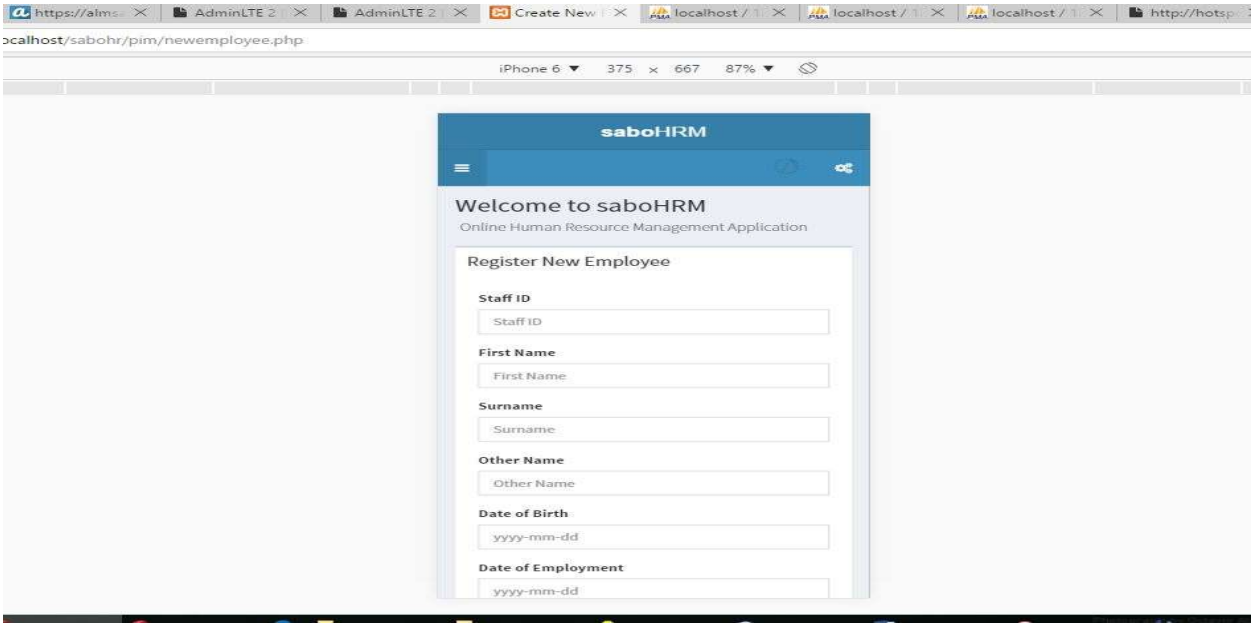

Figure 4.1: A View of The Application Simulated on IPhone 6 Mobile Device.

4.2.1 Login Procedure

Users of the application are expected to have an account the application (Fig 4.2). Upon login, the user is provided with an interface to supply a unique username and password. Once a user successfully login, access to other modules becomes possible.

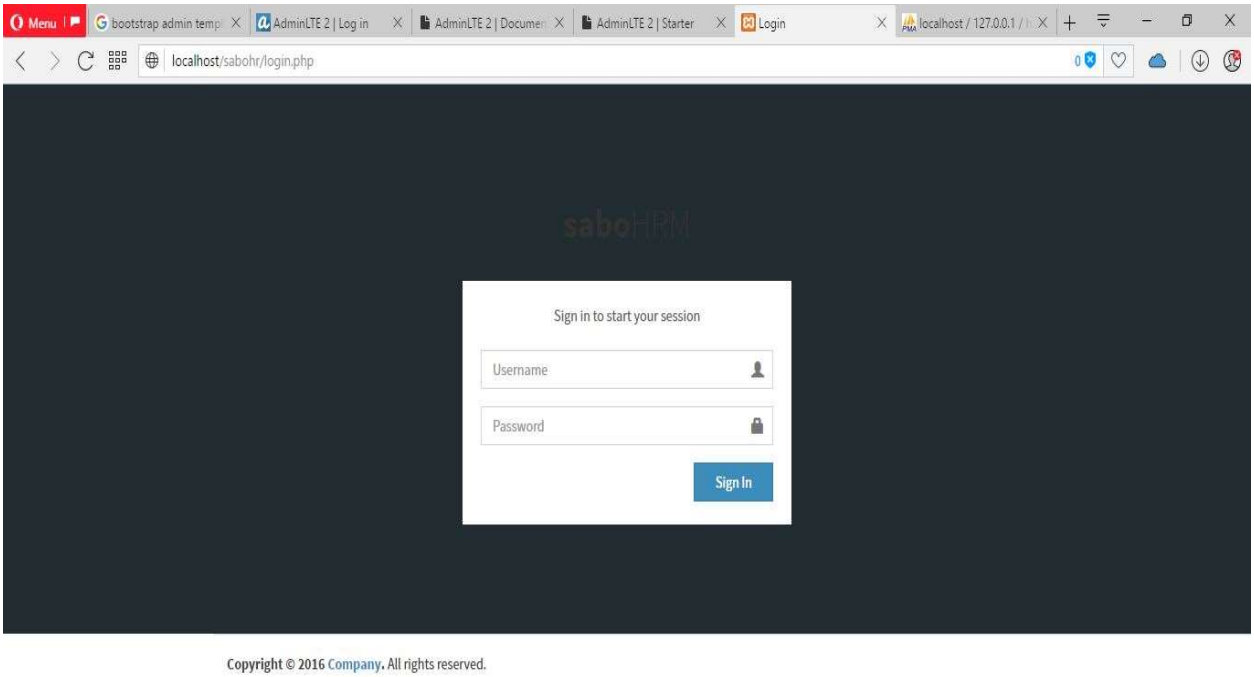

Figure 4.2: Login Page Of Sabohrm

4.2.2 Personnel Information Management (PIM) module

This module provides functionalities and interface for managing employee data. New employee can be added or modified. A list of employee can also be displayed.

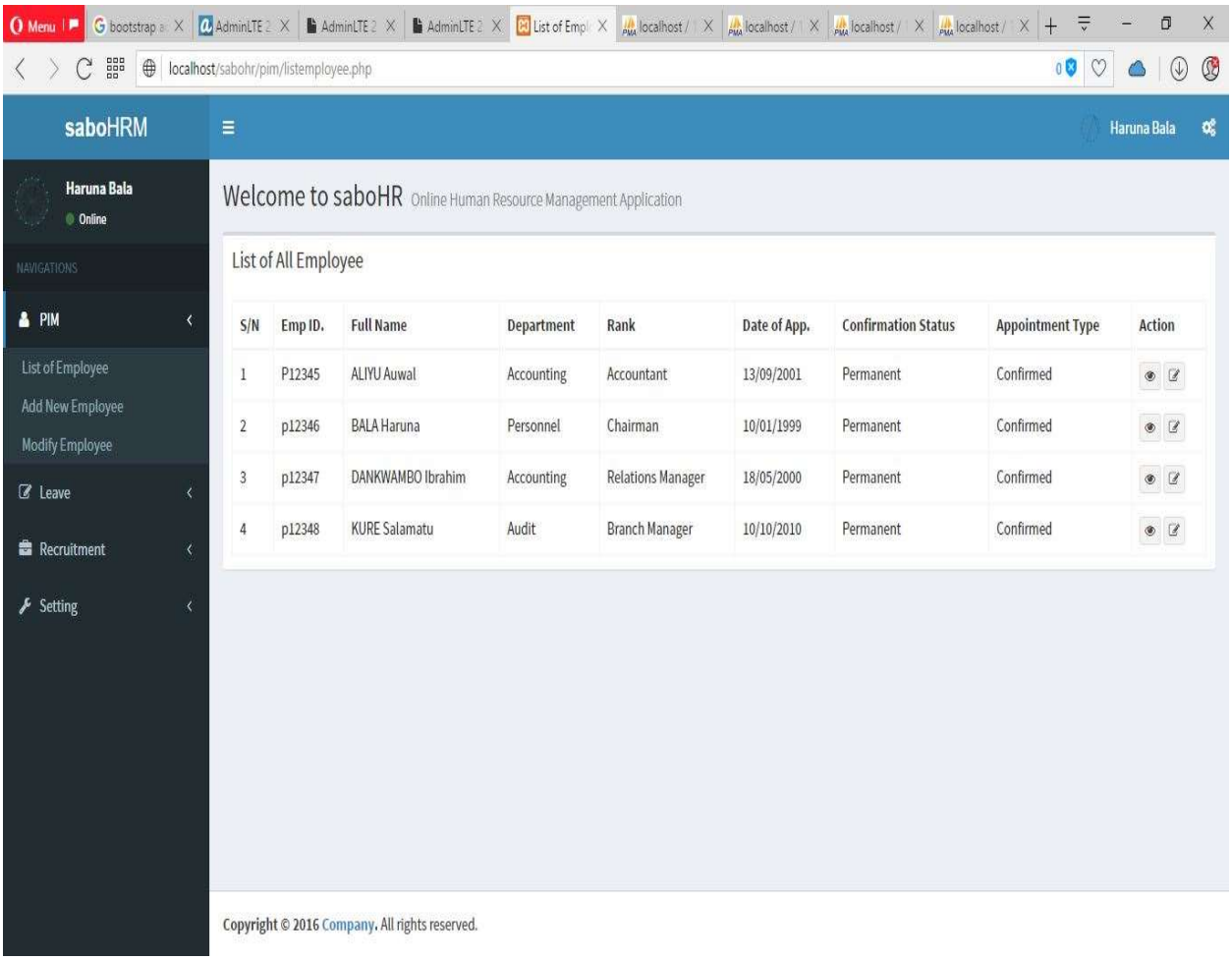

Figure 4.3: Interface Showing the List of Employee Registered With Sabohrm

Users can click on the view or edit action perform an operation on each employee in the list as shown in Fig: 4.4 and 4.5.

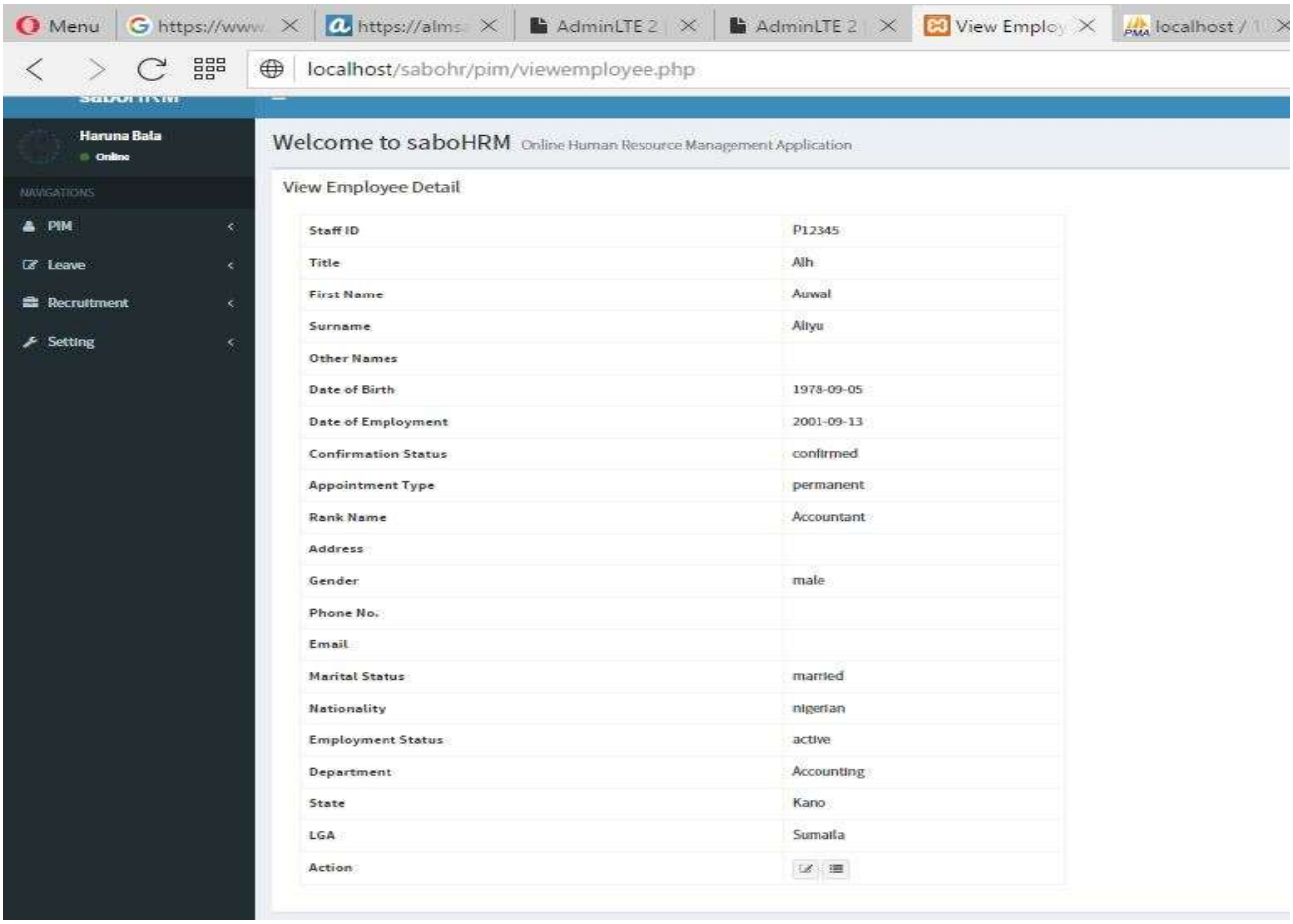

Figure 4.4: Interface Showing a Complete View of a Selected Employee When the View Icon is clicked from the List of Employee

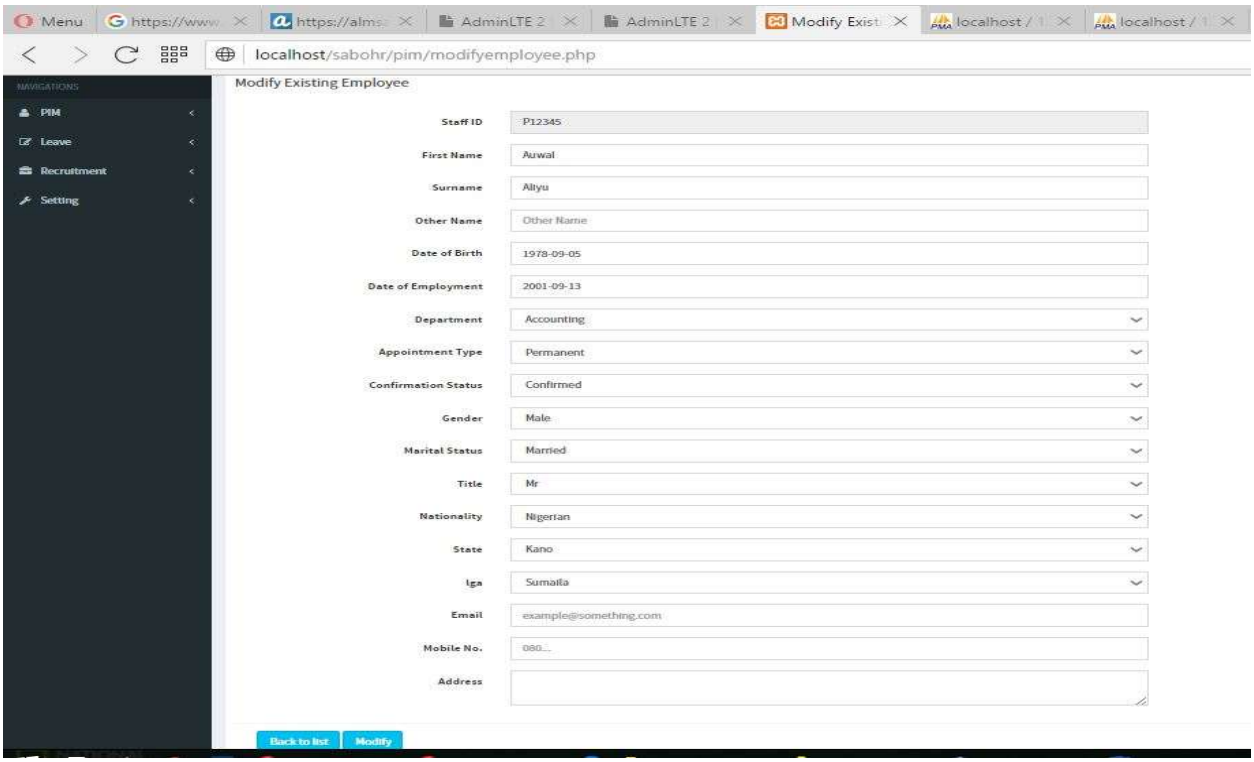

Figure 4.5: Interface Showing a Form for Editing an Employee Detail.

To register new employee users can click on the “Add New Employee” link from the side navigation panel. Below is a figure of a form for registering new employee.

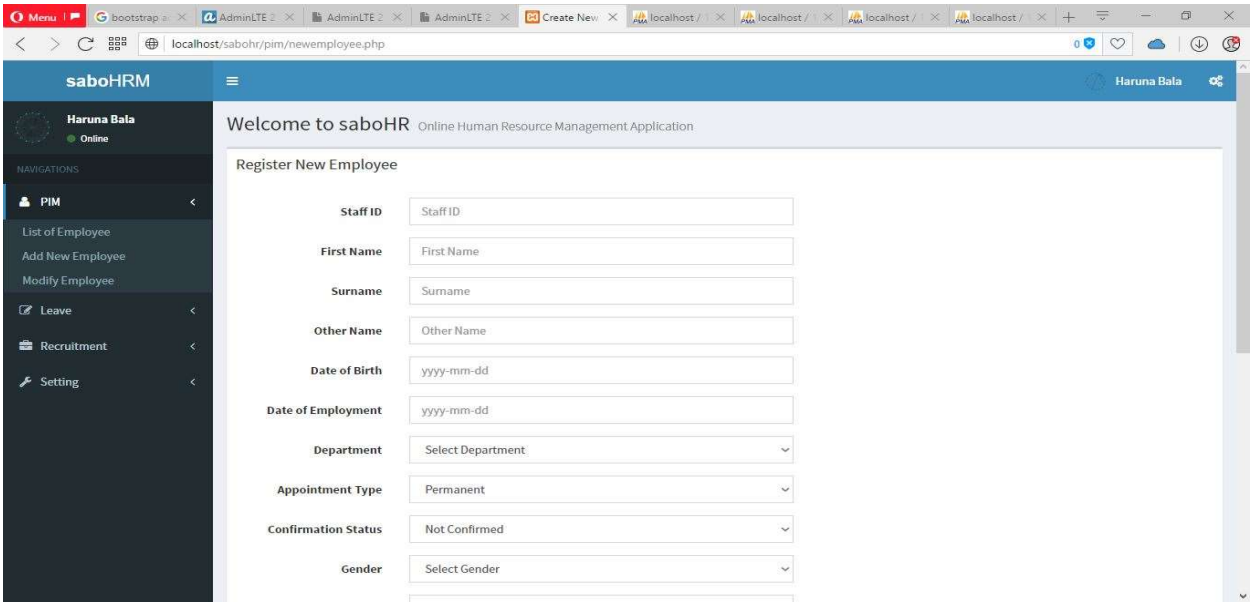

Figure 4.6: A Form for Registering New Employee.

4.2.3 Leave Module

The application provides an interface for employee to apply for leave. From the admin end, a privileged admin can view all leave applications and take action on it (approve or reject). A filter control is provided for the admin to easily filter the list of leave applications by start date, end date and status (Fig 4.7).

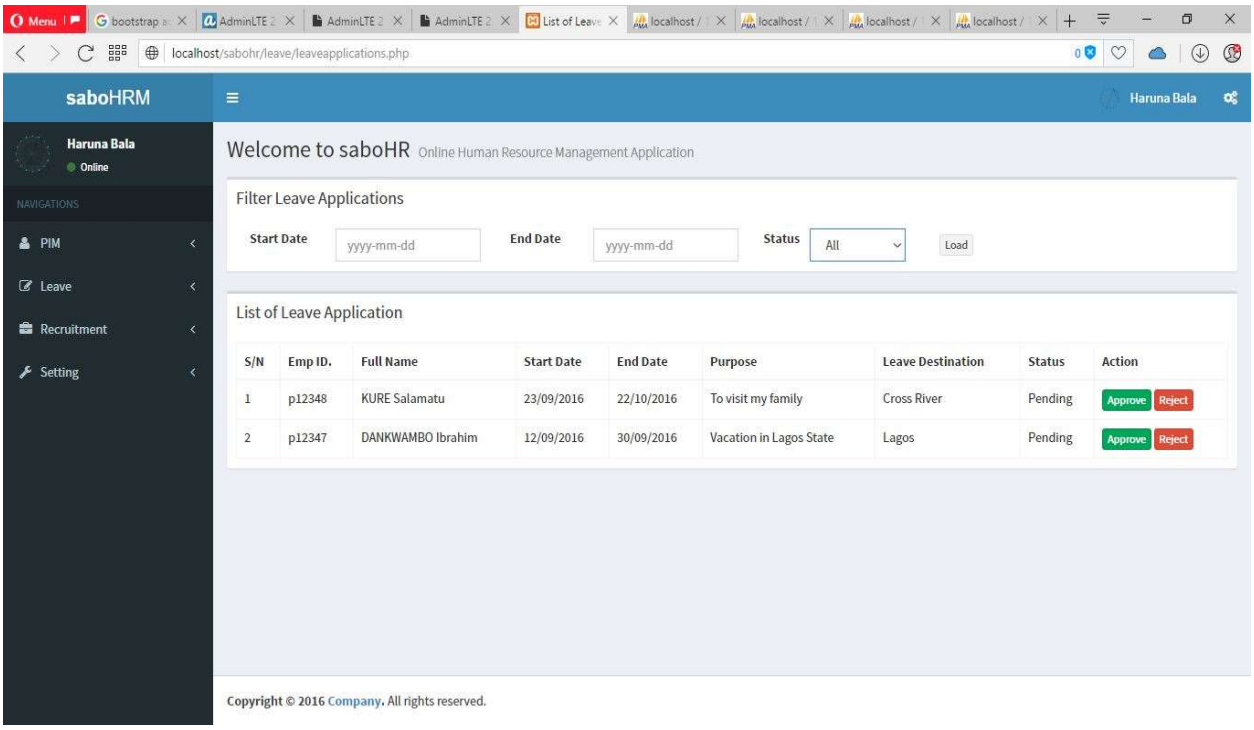

Figure 4.7: An interface for viewing leave application by the admin. Notice the filter controls above the list of leave application

4.2.4 Recruitment module

Prospective employees are also provided with a public interface where job applications can be done consideration by the organization. During the process of job application the applicants will be allowed to upload their curriculum vitae (CV). The admin can view job applications by clicking on the "Manage Job Applications" from the side navigation (Fig 4.8). The interface also provides a filter control for filtering base on position applied and status of the application.

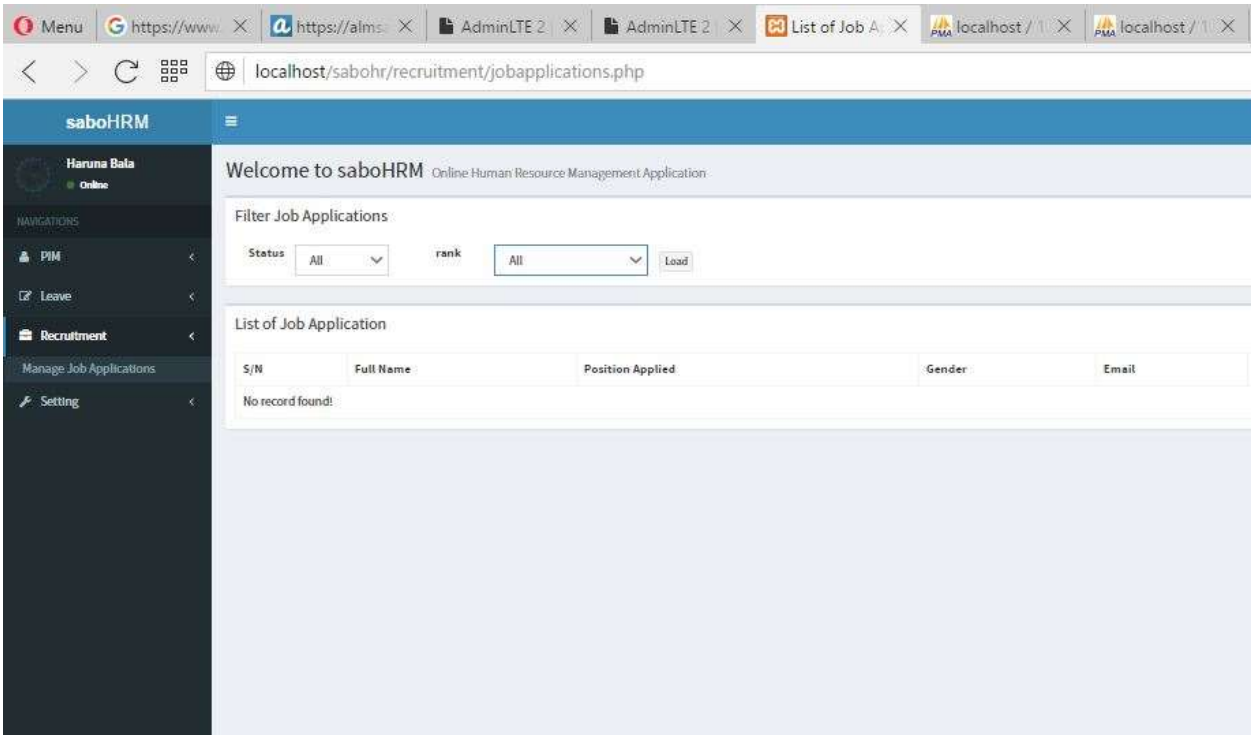

Figure 4.8: Interface for Viewing List of Job Applications.

4.3 Limitations

A typical HRMS application is large providing various modules necessary for complete Human resource management. This application only provides four modules: PIM, Leave, Recruitment and Setting. Because of time constraint and limited technical knowhow, we are able to implement only four. Other modules that could be implemented include Promotion and Payroll among others. Also due to time constraint feature for generating various reports is not included in the application.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Summary

This project covered necessary details involved in designing an online HRMS. The system captures all the information required from workers (using online form) and keeping them in a database. The stakeholders in this application include different individuals ranging from applicants, casual, junior cadre and senior cadre staff of Sabon Gari LGA. Adopted method of information generation in this work include related literature and surfing the Internet for current trends and best practices, while the method of implementation was the use of WAMP (Window Apache MySQL and PHP) software.

The basic requirement of the user of the saboHRM is given preference during the interface design. Therefore, this project if embraced, would address the problems of the existing system outlined earlier.

5.2 Conclusion

Today's advanced technology is noticeable in the application of computers to solve our present challenges, this could serve as inspiration for the automation of routine processes in HRM with a view to enhancing productivity among organizations.

This project has focused on the design and implementation of a web based HRMS for addressing challenges faced by the management of Sabon Gari LGA. Various design and implementation tools have been employed for the successful completion of this work.

The design and implementation tools were selected based on 1) review of similar work successfully implemented by organizations like the Orange HRM group. 2) A careful selection of

open source and state of the art design/implementation tools like Toad Data Modeler (for ER diagram), Wireframe (for user interface mockups) and WAMP (for database development and software coding).

This project has demonstrated a fulfillment of the vital needs of the Sabon Gari LGA HRM by developing a web based application to replace the manual system used by the organization. The system when fully adopted will enhance the process of data collection, data query, data manipulation and data storage in Sabon Gari LGA.

5.3 Recommendations

In view of the aforementioned conclusion, I recommend the following for future work:

i. The saboHRM is open for further research, as some modules (promotion, retirement, salary, and housing etc.) are not exploited and implemented due to limited data and timeframe.

ii. Having designed a HRMS to be used by Sabon Gari LGA, there is need to address the issue of training users of the web based application.

REFERENCES

Aune, S. (2010). *A Brief History Of Flash*. *TechnoBuffalo*. [Online] Available: http://www.technobuffalo.com/2010/04/05/a-brief-history-of-flash (9 August, 2016).

Brown, T. (2016). *Jump Start HTML5: Basics on SitePoint*. *SitePoint Premium*. [Online] Available: http://www.sitepoint.com/premium/books/jump-start-html5-basics/online/ch01 (13 August, 2016).

Chasserio, S. & Legault, M. (2009). Strategic human resources management is irrelevant when it comes to highly skilled professionals in the Canadian new economy. *The International Journal of Human Resource Management*, 20(5), 1113-1131.

Douglas, V. L. (1992). Wellness a Bargain for Life, A Resource for Higher Education. The Complete Guide to Creating an Effective Wellness Program in Higher Educational Institutions. Washington, D.C.: Distributed by ERIC Clearinghouse.

Fraker, M. & Mazza, A. (2011). *Direct-to-consumer genetic testing: Summary of a workshop*. Washington, D.C.: National Academies Press. (Chasserio, 2009)

Kevin, C. & Fiona, B. (2009). Human resource management system. *United States Patent Application Publication*. 12(7), pp.123-140.

Nations, D. (2016). *What is a Web Application?*. *About.com Tech*. [Online] Available: http://webtrends.about.com/od/webapplications/a/web_application.htm (6 August, 2016).

Number of Internet Users (2016) - Internet Live Stats. (2016). *Internetlivestats.com*., [Online] Available: http://www.internetlivestats.com/internet-users/#trend (11 August, 2016).

Soumya, R. L. (2007). The design and implementation of human resource management website. *A masters project submitted to department of Computer and Information Sciences*, Indiana University, South Bend.

Swartz, A. (2016). *A Brief History of Ajax (Aaron Swartz's Raw Thought)*. *Aaronsw.com*. [Online] Available: http://www.aaronsw.com/weblog/ajaxhistory (8 August, 2016).

Syed, A. S. N., Syed, A. S.F., Prabhadevi, C., Sangeetha, V., Gopalakrishnan, S. (2013). Human Resource Management System. *IOSR Journal of Computer Engineering (IOSR-JCE),* 8(4), pp. 62-71.

Victor, M. (2013). *The Local Government System in Nigeria*. [Online] Available: http://infinityinfotainment.blogspot.com.ng/2013/04/the-local-government-system-in-nigeria.html (6 July, 2016).

Williams, R. (2015). *Web browsers: a brief history*. *Telegraph.co.uk*. [Online] Available: http://www.telegraph.co.uk/technology/microsoft/11577364/Web-browser-history.html (18 July 2016).

Zafar, H. (2013). Human resource information systems: Information security concerns for organizations. *Human Resource Management Review*, 23(1), pp.105-113.

APPENDIX

login.php

```
<?php
require_once ("global.php");
require_once ("utility/connection.php");
$pagetitle = "Login";
$page = "login";
?>
<!DOCTYPE html>
<html>
    <head>
        <?php
        require_once("partial/asset.php");
        ?>
    </head>
    <body class="hold-transition skin-blue sidebar-mini">
        <div class="wrapper">
            <div class="login-box">
                <div class="login-logo">
                    <b>sabo</b>HRM
                </div>
                <!-- /.login-logo -->
                <div class="login-box-body">
                    <p class="login-box-msg">Sign in to start your session</p>

                    <form action="login_exec.php" method="post">
                        <div class="form-group has-feedback">
                            <input type="text" class="form-control"
placeholder="Username">
                            <span class="glyphicon glyphicon-user form-control-
feedback"></span>
                        </div>
                        <div class="form-group has-feedback">
                            <input type="password" class="form-control"
placeholder="Password">
                            <span class="glyphicon glyphicon-lock form-control-
feedback"></span>
                        </div>
                        <div class="row">
                            <div class="col-xs-8">
                            </div>
                            <!-- /.col -->
                            <div class="col-xs-4">
                                <button type="submit" class="btn btn-primary btn-block
btn-flat">Sign In</button>
                            </div>
                            <!-- /.col -->
                        </div>
                    </form>
                </div>
                <!-- /.login-box-body -->
            </div>
        </div>
        <!-- /.content-wrapper -->

        <?php
        require_once("partial/footer.php");
        ?>
    </body></html>
```

index.php

```
<?php
require_once ("../global.php");
$pagetitle = "saboHR";
$page = "index";
?>

<!DOCTYPE html>
<html>
    <head>
        <?php
        require_once("../partial/asset.php");
        ?>
    </head>
    <body class="hold-transition skin-blue sidebar-mini">
        <div class="wrapper">

            <!-- Main Header -->
            <?php
            require_once("../partial/header.php");
            ?>
            <!-- Left side column. contains the logo and sidebar -->
            <?php
            require_once("../partial/sidenav.php");
            ?>

            <!-- Content Wrapper. Contains page content -->
            <div class="content-wrapper">
                <!-- Content Header (Page header) -->
                <section class="content-header">
                    <h1>
                        Welcome to saboHR
                        <small>Online Human Resource Management Application</small>
                    </h1>
                </section>
                <!-- Main content -->
                <section class="content">


                </section>
                <!-- /.content -->
            </div>
            <!-- /.content-wrapper -->

            <?php
            require_once("../partial/footer.php");
            ?>
    </body>
</html>
```

listemployee.php

```
<?php
require_once ("../global.php");
require_once ("../utility/connection.php");
require_once (rootAddress("classes/Pim.php"));
$pagetitle = "List of Employee";
$page = "listemployee";
?>

<!DOCTYPE html>

<html>
    <head>
        <?php
        require_once("../partial/asset.php");
        ?>
    </head>

    <body class="hold-transition skin-blue sidebar-mini">
        <div class="wrapper">

            <!-- Main Header -->
            <?php
            require_once("../partial/header.php");
            ?>
            <!-- Left side column. contains the logo and sidebar -->
            <?php
            require_once("../partial/sidenav.php");
            ?>

            <!-- Content Wrapper. Contains page content -->
            <div class="content-wrapper">
                <!-- Content Header (Page header) -->
                <section class="content-header">
                    <h1>
                        Welcome to saboHRM
                        <small>Online Human Resource Management Application</small>
                    </h1>
                </section>
                <!-- Main content -->
                <section class="content">
                    <div class="box">
                        <div class="box-header">
                            <h3 class="box-title">List of All Employee</h3>
                        </div>
                        <!-- /.box-header -->
                        <div class="box-body">
                            <table class="table table-bordered table-hover dataTable"
role="grid" aria-describedby="example2_info">
                                <thead>
                                    <tr role="row"><th>S/N</th><th>Emp
ID.</th><th>Full Name</th><th>Department</th><th>Rank</th><th>Date of
App.</th><th>Confirmation Status</th><th>Appointment Type</th><th>Action</th></tr>
                                </thead>
                                <tbody>

                                    <?php
                                    $sn = 1;
                                    $rows = Pim::getAllEmployee();
                                    if (count($rows) > 0) {
                                        echo"";
                                    }
                                    foreach ($rows as $row) {
```

```
                                        ?>
                                        <tr role="row">
                                            <td><?php echo $sn; ?></td>
                                            <td><?php echo $row['staff_id']; ?></td>
                                            <td><?php echo strtoupper($row['surname'])
." ".  ucfirst($row['firstname']). " ". ucfirst($row['othernames']) ; ?></td>
                                            <td><?php echo $row['departmentname'] ;
?></td>
                                            <td><?php echo $row['rankname'] ; ?></td>
                                            <td><?php $dt = new
DateTime($row['dateofemployment']); echo $dt->format("d/m/Y"); ?></td>
                                            <td><?php echo
ucfirst($row['appointmenttype']); ?></td>
                                            <td><?php echo
ucwords($row['confirmationstatus']); ?></td>
                                            <td>
                                                <a class="btn btn-default btn-xs"
title="View Details" href='viewemployee.php?empid=<?php echo $row['employeeid'];
?>'><i class='fa fa-eye'></i></a>
                                                <a class="btn btn-default btn-xs"
title="Modify" href='modifyemployee.php?empid=<?php echo $row['employeeid']; ?>'><i
class='fa fa-edit'></i></a></td>
                                        </tr>

                                        <?php
                                        $sn++;
                                    }
                                    ?>
                                </tbody>
                            </table>
                        </div>
                    </div>
                </section>
                <!-- /.content -->
            </div>
            <!-- /.content-wrapper -->

            <?php
            require_once("../partial/footer.php");
            ?>
    </body>
</html>
```

```
<?php
require_once ("../global.php");
require_once ("../utility/connection.php");
require_once (rootAddress("classes/Recruitment.php"));
$pagetitle = "List of Job Applications";
$page = "jobapplications";
?>
<!DOCTYPE html>
<html>
    <head>
        <?php
        require_once("../partial/asset.php");
        ?>
    </head>
    <body class="hold-transition skin-blue sidebar-mini">
        <div class="wrapper">
            <!-- Main Header -->
            <?php
            require_once("../partial/header.php");
            ?>
            <!-- Left side column. contains the logo and sidebar -->
            <?php
            require_once("../partial/sidenav.php");
            ?>
            <!-- Content Wrapper. Contains page content -->
            <div class="content-wrapper">
                <!-- Content Header (Page header) -->
                <section class="content-header">
                    <h1>
                        Welcome to saboHRM
                        <small>Online Human Resource Management Application</small>
                    </h1>
                </section>
                <!-- Main content -->
                <section class="content">
                    <div class="box">
                        <div class="box-header">
                            <h3 class="box-title">Filter Job Applications</h3>
                        </div>
                        <!-- /.box-header -->
                        <div class="box-body">
                            <form class="form-inline" id="filterfrm">
                                <div class="form-group">
                                    <label for="status" class="col-sm-3 control-
label">Status</label>
                                    <div class="col-sm-5">
                                        <select class="form-control" id="appstatus"
name="appstatus">
                                            <option value="">All</option>
                                            <option value="pending">Pending</option>
                                            <option value="approved">Approved</option>
                                            <option value="rejected">Rejected</option>
                                        </select>
                                    </div>
                                </div>
                                <div class="form-group">
                                    <label for="rank" class="col-sm-3 control-
label">rank</label>
                                    <div class="col-sm-5">
                                        <select class="form-control" id="rank"
name="rank">
                                            <option value="">All</option>
                                            <?php
```

```
                                                $sql = "select * from rank";
                                                $objPDO= openConnection();
                                                $stmt = $objPDO->prepare($sql);
                                                $stmt->execute();
                                                while($row = $stmt-
>fetch(PDO::FETCH_ASSOC)){
                                                    echo "<option
value='".$row['rankid']."'>".$row['rankname']."</option>";
                                                }
                                            ?>
                                        </select>
                                    </div>
                                </div>
                                <input class="btn btn-default btn-xs" type="submit"
name="load" id="load" value="Load" />
                            </form>
                        </div>
                    </div>
                    <div class="box">
                        <div class="box-header">
                            <h3 class="box-title">List of Job Application</h3>
                        </div>
                        <!-- /.box-header -->
                        <div class="box-body">
                            <table class="table table-bordered table-hover dataTable"
role="grid" aria-describedby="example2_info">
                                <thead>
                                    <tr role="row"><th>S/N</th><th>Full
Name</th><th>Position Applied</th><th>Gender</th><th>Email</th><th>Mobile</th><th>App.
Status</th><th>Action</th></tr>
                                </thead>
                                <tbody id="filtercontent">


                                </tbody>
                            </table>
                        </div>
                    </div>
                </section>
                <!-- /.content -->
            </div>
            <!-- /.content-wrapper -->

            <?php
            require_once("../partial/footer.php");
            ?>
            <script type="text/javascript">

                $(document).on("submit", "#filterfrm", function (event) {
                    $("#filtercontent").html("<i>Loading</i>");
                    $.ajax({
                        type: 'POST',
                        url: 'ajax/getjobapplications.php',
                        data: $("#filterfrm").serialize()
                    }).done(function (data) {
                        $("#filtercontent").html(data);
                    });
                    return false;
                });
            </script>
    </body>
</html>
```

leaveapplication.php

```
<?php
require_once ("../global.php");
require_once ("../utility/connection.php");
require_once (rootAddress("classes/Leave.php"));
$pagetitle = "List of Leave Applications";
$page = "leaveapplications";
?>

<!DOCTYPE html>
<html>
    <head>
        <?php
        require_once("../partial/asset.php");
        ?>
    </head>
    <body class="hold-transition skin-blue sidebar-mini">
        <div class="wrapper">

            <!-- Main Header -->
            <?php
            require_once("../partial/header.php");
            ?>
            <!-- Left side column. contains the logo and sidebar -->
            <?php
            require_once("../partial/sidenav.php");
            ?>

            <!-- Content Wrapper. Contains page content -->
            <div class="content-wrapper">
                <!-- Content Header (Page header) -->
                <section class="content-header">
                    <h1>
                        Welcome to saboHR
                        <small>Online Human Resource Management Application</small>
                    </h1>
                </section>
                <!-- Main content -->
                <section class="content">
                    <div class="box">
                        <div class="box-header">
                            <h3 class="box-title">Filter Leave Applications</h3>
                        </div>
                        <!-- /.box-header -->
                        <div class="box-body">
                            <form class="form-inline" id="filterfrm">
                                <div class="form-group">
                                    <label for="start" class="col-sm-4 control-
label">Start Date</label>

                                    <div class="col-sm-5">
                                        <input type="text" class="form-control"
id="start" name="start" placeholder="yyyy-mm-dd">
                                    </div>
                                </div>
                                <div class="form-group">
                                    <label for="end" class="col-sm-4 control-
label">End Date</label>

                                    <div class="col-sm-5">
                                        <input type="text" class="form-control"
id="end" name="end" placeholder="yyyy-mm-dd">
                                    </div>
                                </div>
```

```
                                <div class="form-group">
                                    <label for="status" class="col-sm-3 control-
label">Status</label>

                                    <div class="col-sm-5">
                                        <select class="form-control" id="status"
name="status">
                                            <option value="">All</option>
                                            <option value="pending">Pending</option>
                                            <option value="approved">Approved</option>
                                            <option value="rejected">Rejected</option>
                                        </select>
                                    </div>
                                </div>
                                <input class="btn btn-default btn-xs" type="submit"
name="load" id="load" value="Load" />
                            </form>
                        </div>
                    </div>
                    <div class="box">
                        <div class="box-header">
                            <h3 class="box-title">List of Leave Application</h3>
                        </div>
                        <!-- /.box-header -->
                        <div class="box-body">
                            <table class="table table-bordered table-hover dataTable"
role="grid" aria-describedby="example2_info">
                                <thead>
                                    <tr role="row"><th>S/N</th><th>Emp
ID.</th><th>Full Name</th><th>Start Date</th><th>End
Date</th><th>Purpose</th><th>Leave Destination</th><th>Status</th><th>Action</th></tr>
                                </thead>
                                <tbody id="filtercontent">

                                </tbody>
                            </table>
                        </div>
                    </div>
                </section>
                <!-- /.content -->
            </div>
            <!-- /.content-wrapper -->

            <?php
            require_once("../partial/footer.php");
            ?>
            <script type="text/javascript">

                $(document).on("submit", "#filterfrm", function (event) {
                    $("#filtercontent").html("<i>Loading</i>");
                    $.ajax({
                        type: 'POST',
                        url: 'ajax/getleaveapplications.php',
                        data: $("#filterfrm").serialize()
                    }).done(function (data) {
                        $("#filtercontent").html(data);
                    });
                    return false;
                });
            </script>
    </body></html>
```