**CHAPTER FOUR: SYSTEM IMPLEMENTATION**

**4.0 Introduction**

System Implementation is another phase in the software development lifecycle (SDLC) and is preceded by system design and analysis phase. System Implementation involves transforming the deliverables of the analysis and design phase into an acceptable software application using various technologies and programming languages. It involves the following

* Development environment
* Choice of programming language used
* System platform
* IDE
* Implementation Architecture

**4.1 Choice of Development Environment**

The environment used in developing this software is Sublime text 2, its’ a fast IDE and PHP editor with built-in HTML, CSS and JavaScript editor with features that helps in creating good looking websites.

**4.1.0 System Specification**

The system’s specification for Performance Tracker software are grouped into three which include

* Software Requirements
* Hardware Requirements

**Software Specification**

The software required for the Performance Tracker software includes:

* Operating System: Windows 7, Windows 8/8.1
* Server: An integrated Apache and MySQL server (XAMPP or WAMPP)
* Web browser: Chrome, Internet Explorer, Firefox, UC browser etc.

**Hardware Specification**

The hardware specification for the system:

* Processor-Pentium(R) Quad core CPU
* Secondary Memory ­­– at least 500mb free Hard Disk space is recommended.
* Ram size: 4.00GB.
* Peripherals: Keyboard, pointing devices etc.

**4.1.1 Choice of Programming Language**

The programming languages used in developing this software include:

* HTML
* CSS
* JavaScript
* MySQL
* PHP

Some of the reasons for using these programming languages are

* They communicate well with one another. PHP has a built-in feature for communicating with MySQL.
* It can run on many operating system: for example, Linux, Mac OS, Windows.
* It’s designed to support databases: PHP includes functionality designed to interact with specific databases. It relieves you the need to know the technical details required to communicate.
* HTML has an easy syntax.
* PHP is embedded inside HTML code, this makes the response time short.
* PHP is not expensive.

**4.2 Implementation Architecture**

Home Page

Contact

Account

Registration

Login

Add website

View profile

Logout

Fig 4.1: Implementation Architecture Diagram

**4.3 Software Testing**

The system is tested at every stage of its development in other to be able to detect errors and remove them immediately. The testing is in two phases

* Firstly, testing during development phases by removing bugs.
* Secondly, testing done by running the website on the local host of the developer’s computer. One can browse the website in other to know if the system meets the department’s requirement.

**4.3.0 Screenshots Of Some Dem**

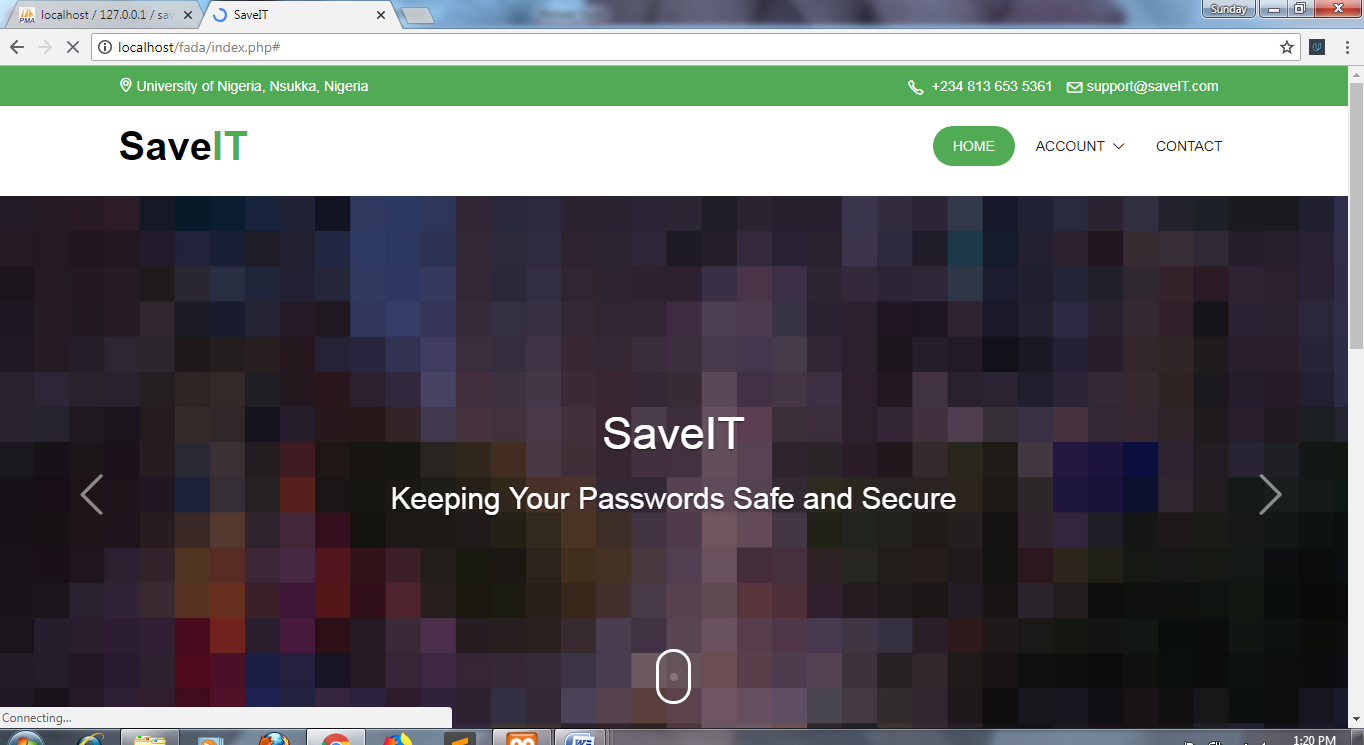
****

Figure 4.3.1 Screen shot of the index page.

Fig 4.7: Admin Update Profile

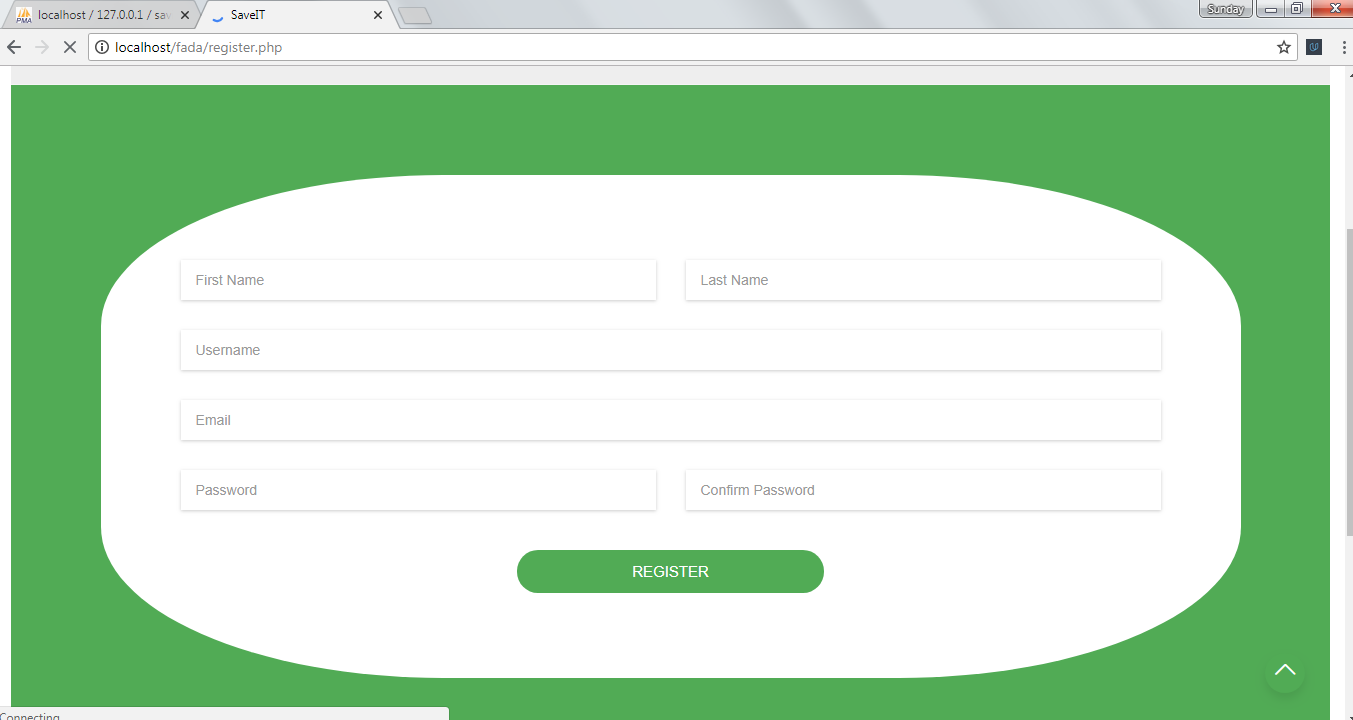
****

Figure 4.3.2 Screen shot of the system registration page.

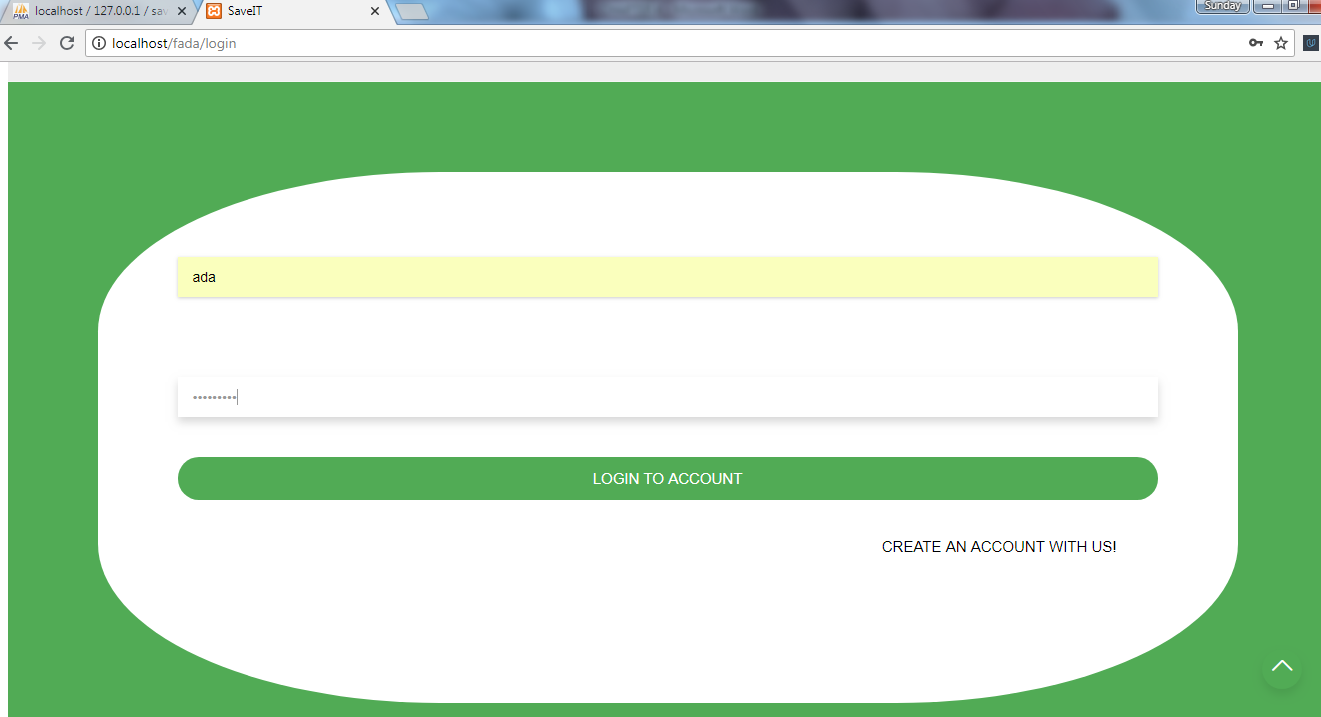
****

Figure 4.3.3 Screen shot of the login page.

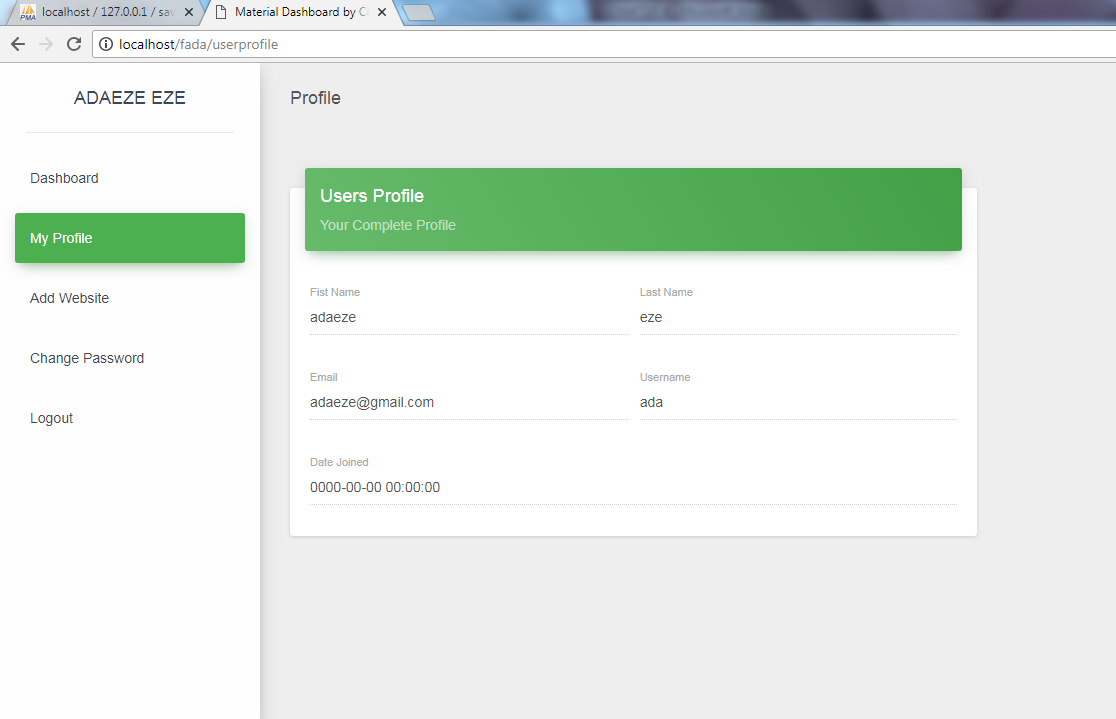


Figure 4.3.4 Screen shot of user profile page.

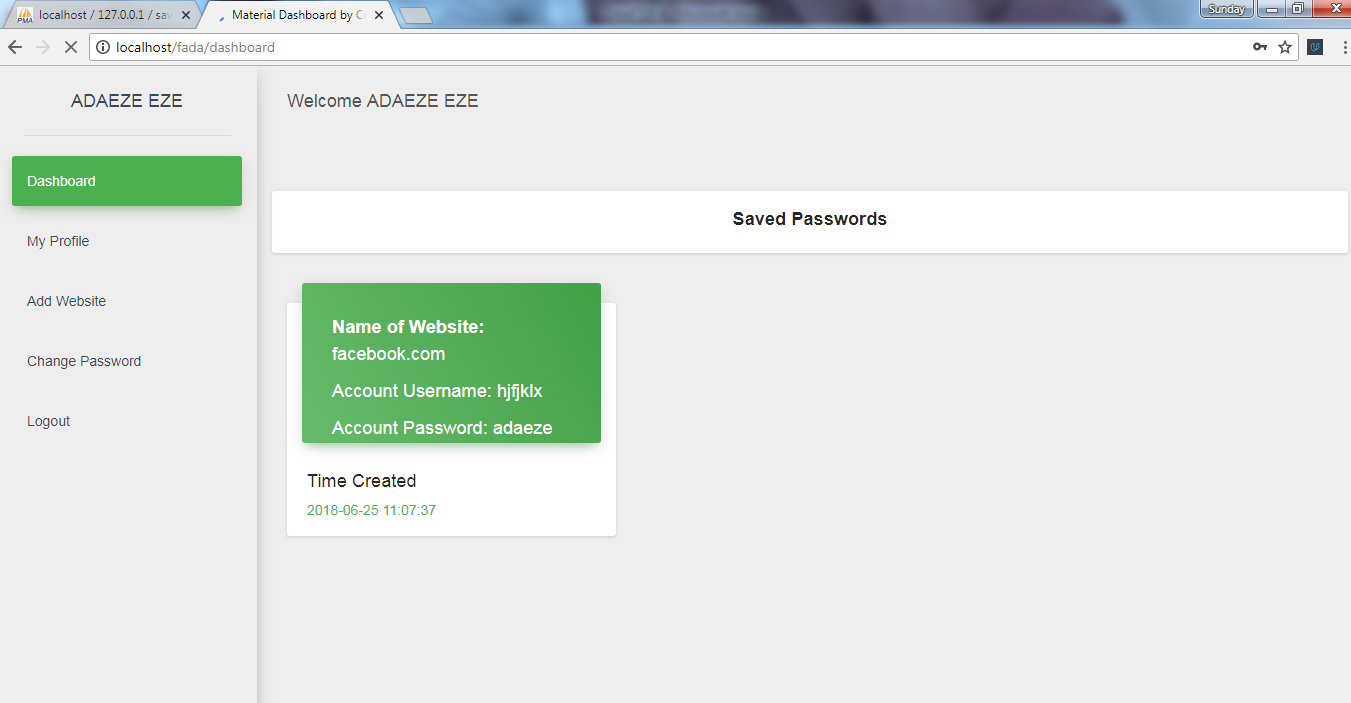
****

Figure 4.3.4 Screen shot of user dash board page

**4.4 Documentation**

The purpose of this documentation is to promote understandability of system, and to promote a user friendly experience for users. However, this documentation can be referred to both users and developers.

**4.4.1 User Manual**

This section provides the step by step method of using the software. The software is user friendly and interactive; this makes it easy for people who do not know how to use the system efficiently.

To use the software the users, have to do the following

* Boot a PC.
* Run your Local Host (Xampp)
* Open your web browser (e.g internet explorer, mozilla) and type in your localhost/index.php on the address bar and press enter.
* When the index page is displayed, user can login the dashboard.
* The user can add a web URL
* The user can upload username, password, or manage passwords.

**4.4.2 SOURCE CODE LISTING**

See Appendices