Chapter One: **INTRODUCTION**

**1.0 Introduction:**

Passwords are means of authentication and security to online users as there is need to keep their data secured. These passwords on daily basis have been continuously targeted at by various cracking and harvesting attacks. Basically password security heavily depends on creating strong passwords that are sufficiently long and random, and protecting them from being guessed or stolen [1]. However researchers have demonstrated that strong passwords that are sufficiently long, random, and hard to crack by attackers are often difficult to remember by users. But no matter how you go about making a password strong, online passwords are still vulnerable to harvesting attacks such as phishing. These hard problems have been further aggravated by the fact that web users have more online accounts than ever before, and they are forced to create and remember more and more username and passwords probably using insecure practices such as sharing the same password across different websites and writing passwords down [2]. Even with strong security policies bad security behavior like using weak passwords, reusing passwords and writing them down are still a big part of the everyday process [3]. According to one study, users are generally unaware what constitutes a good password and a bad one [4]. It is also important to have in mind that strict policies regarding passwords can also backfire, causing even more security issues than before the implementation [1].

A new class of software grew popular: password managers. Password manager is one of the most popular solutions that can potentially well address the aforementioned password security and management problems[5].

In general, password manager’s work by saving users online passwords and later auto filling the login forms behalf of the user. Therefore a remarkable benefit brought by this is that user do not have to remember many passwords, most of this password managers are browser based.

In this work, a cloud based password management system that is not browser based is introduced, where user can upload there authentication details and save them on the cloud not on the users local machine and user need not to remember this passwords or the one used to login to these platform.

* 1. **Statement of the Problem.**

Online users forget passwords easily and there is a need for a system to help the user or authority in accessing, protecting and storing passwords.

Meanwhile browser based password management is attracted using phishing and java scripts. In this research / project a design and web security for cloud-based password management would be considered.

* 1. **Aim**

The aim of this project is to design a web security for cloud based password management.

* 1. **Objectives**

The following are the objectives of this system:

1. To develop a website where online users can store password.
2. To enable user register and save URL and login details.
3. To encrypt the user password so that no one can have access to it.
4. To make it easy for the user not to forget the login details to this system.
   1. **Significance of The Study**

The significance of this project is to provide a reliable way of handling online password security reasons. This project work will aid the user to relie on the system to remember their password and save them the trouble of forgettin

* 1. **Scope of Study**

This study is aimed at helping user secure their login details in the cloud by encryption. This also makes it possible for user to login from any device and don’t have to worry remembering passwords again.

Chapter Two**: LITERATURE REVIEW**

**2.0 Introduction**

The technologies embraced in the development of this project are discussed in this chapter. Related works are also reviewed and analyzed in the later section of this chapter.

**2.1 Theoretical Background**

In this chapter, we will look into the technologies related to password management and introduce the technologies used in development process. We also elaborate more on the literature review of other research done in this field and also compare it with current ones to effectively see similarities, contrasts and improvements so far.

The major technologies used in this project are web related technologies such as PHP and relational database technology such as MySQL. With PHP as the frontend and MySQL as the backend database. PHP (Hypertext Preprocessor) is a widely used open source general purpose scripting language that is especially suited for web development and can be embedded into HTML (Hypertext Markup Language) SQL (Structural programming Language) server provides database to store information on remote location. The system can be accessed using a username and password by the user and the saved details cannot be seen or used by the manager of the system.

The basic functionalities of this system are that user can be able to create, save, and delete username and password on this platform. This security system has also a way to make user not to forget the login details of this system by using OTP (0ne time password) for logging in. password managers always face a challenge, which is phishing that happens through extensions built on other browser password management system but this system is developed without an extension, it serves as a password bank.

**2.2 Literature Review**

This section explores the different web password management systems previously developed by companies or proposed in publications.

Morris and Thompson pointed out long ago in 1979 that weak passwords

Suffer from brute-force and dictionary attacks [6], this passwords are called weak in the sense that the user uses simple and easy to remember words such as user first name, last name or words in the dictionary as password which the attacker can easily get by running very large set of words to generate potential passwords. . Later, Feldmeier and Karn Further emphasized that increasing password entropy is critical to improving Password security [7].

However, strong passwords that is sufficiently long, Random, and hard to crack by attackers are often difficult to remember by Users due to human memory limitations. Adams and Sasse discussed pass- word memorability and other usability issues and emphasized the importance

of user-centered design in security mechanisms [8].

Web Wallet [9] is an anti-phishing solution and is also a password manager that can help users fill login forms using stored information; however, as pointed out by

the authors, users have a strong tendency to use traditional Web forms for

typing sensitive information instead of using a special browser sidebar user

interface.

In addition, Web Wallet is not cloud-based, which means the user information are saved in the local machine. In [10] the authors have proposed user-centric trust Identity service with an aim to create trust among Cloud Service Providers (CSP). Their model has Authentication, Authorization, and Provisioning and Audit modules along with the Trust agent. The federated environment will allow users to login to various Cloud Service Providers depending on the application access. When the user moves to different CSP the user credentials follow in the federated environment. The Trust Agent in the Identity Management sends the Trust Token along with the user attribute which creates a trust between CSPs.

In [11] the authors have propose an Identity and Access Management architecture in cloud to achieve security requirement like Strong Authentication, Data Loss Prevention, Security as a Service. The various systems components for addressing the above security requirements are Cloud Resource provider, Identity Management, Policy Management, Resource Engine and Access Decision-making. The various advantages of their approach were Comprehensive identity management, standardized architecture, and scalable design.

Information security can be defined as protecting information with regard to three main requirements: confidentiality, integrity, and availability. Confidentiality meaning that unauthorized parties are not allowed to intercept the information. Integrity meaning that the information has not been corrupted or changed during storage or transmission by unauthorized parties. Availability meaning that information is available to the authorized parties when trying to access it. This model is often referred to as the CIA triad. Modern business needs have also increased the need for non-repudiation which means that an action, e.g. a business transaction, cannot be denied afterwards [12]. Some has criticized the model of not including enough principles to ensure the information security of an organization. Dhillon and Backhouse proposed already in 2000 that the principles of responsibility, integrity (integrity of a person and not only information), trust and

2ethicality could be added to the model [13]. Another study proposes that authentication, access control, and non-repudiation should be added to the model [14]. Even though the CIA triad may not present the whole picture when it comes to information security there is consensus in the scientific community that confidentiality, integrity, and availability are key concepts in ensuring the security of information [13],[10] [17].

Chapter Three**: SYSTEM ANALYSIS AND DESIGN**

**3.0 Introduction**

With the many ways, the use the Internet, it’s easy to consider some passwords less important than others. However, all passwords are important because wrongdoers can piece together the information you store online and use it for their benefit. They can even use information you share on social media networks. And commercial websites give customers the ability to store billing and shipping addresses along with credit card information.

**3.1 Description of the Existing System**

Secure password practices result in numerous cryptic passwords which are very difficult to keep track of. It is impossible for most people to consistently remember more than just a few of them. Any time we discuss password managers, the ensuing commentary can sometimes get a little heated. People really love their password managers and we love to hear that too. One of the biggest, if not THE biggest, point of contention, however, is the cloud. Specifically, the cloud as a place to store your password vault, the cache of your credentials that your password manager absolutely needs to keep safe at all costs. This is a clear dividing line for many password manager aficionados. The following are the features found in the existing system.

1. The existing system requires a user must create his/her account and download the system extension on user device.
2. It is a browser based
3. User details are stored both on the local machine and on remote cloud storage server.

**3.2 Analysis of the Proposed System**

The complex methods that attackers can use to gain access to your personal information are becoming more easily accessible to wrongdoers and are increasingly effective. It is important to avoid the common mistakes that give these individuals the opportunity to exploit your personal data. The system to be built has the following functionalities and features.

1. Ability for a user to create a profile.
2. Ability for a user to store files in the system.
3. The user can update his/her files or profile.
4. The system uses a master password to authenticate a user during login process.
5. The system encrypts the user password to avoid being hacked by others.
6. The system stores and secures the user’s profile from others.
7. The system uses a web based platform to function.

**3.2.1 Non-Functional Requirements**

These are the tools that will aid the functional requirements of the proposed system. In addition to the functional requirements are other requirements that don't actually do anything, but that are critical nevertheless. In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. These nonfunctional requirements, also called "quality attributes," specify system characteristics that are required for acceptance of the system by the end user. It is therefore imperative that you document such requirements in the functional specification. The following are the non-functional requirements of the store management system;

1. **Reliability:** Users of this system will be confident of its functionalities and lose fear of failure of the system.
2. **Availability:** The system will be functional at any point in time the user is in need of it.
3. **Security:** The system will protect the user’s details of transaction and will not disclose the activities performed in the system to any end user. All users of the system must be uniquely identified.
4. **Performance:** The system will be able to satisfy the end user for the reason why it was developed.
5. **Usability:** The system will be developed in a way that the reason for its development will be met, i.e. users will cope and make use of the system appropriately. The system’s user interface intuitive, easy to use and provide an overall positive user experience.
6. **Integrity:** The online store management system will be able to protect and preserve transactions made in the system.

vii. **Maintainability:** The system will be developed in a way that changes can be made easily i.e. for bug fixes or adding new functionality in the future. The system will be made easy for the user to maintain with just reading through the manual or undergoing a few hours training. All source code shall adhere to an agreed upon and well-defined set of coding standards for each development language used.

**3.2.2 Functional Requirements**

[Password manager](https://en.wikipedia.org/wiki/Password_manager) software is used by individuals to organize and encrypt many personal passwords using a single login. This often involves the use of an [encryption key](https://en.wikipedia.org/wiki/Encryption_key) as well. Some of the functional requirements for this system are as follows:

**Logout:** A logout process must be provided. An automatic logout during session inactivity must occur.

**Password security:** Passwords are stored on the database as a secure hash and cannot be decoded. All passwords during logon are transmitted from client to server as a salted hash, an encrypted hash, or an encrypted salted hash.

**Login:** A login process must be provided.

**Password Reset:** The password reset page must utilize the same requirements as the account logon page.

**Account Creation:** The password entry page portion of account creation must be separate from the parts of the account creation page that contain personal data including a secret question.

**3.3 Design of the Proposed System**

The system modeling tool used is object oriented analysis and design and it employs the use of unified modeling language (UML). UML is a method of visualizing and documenting information

UML diagram can be of three types:

1. Use case diagram
2. Class Diagram
3. Activity Diagram

**3.3.1 Use Case**

UML Use Case Diagrams can be used to describe the functionality of a system in a horizontal way. That is, rather than merely representing the details of individual features of your system, Use case diagrams can be used to show all of its available functionality. It is important to note, though, that Use case diagrams are fundamentally different from sequence diagrams or flow charts because they do not make any attempt to represent the order or number of times that the systems actions and sub-actions should be executed.

Fig. 3.3.1 Use case diagram

**3.3.2 Class Diagram**

The class diagram helps in specifying the structural relationship between parts of the system in an object oriented manner. Each noun or “thing,” describe in the requirement document is qualified to be a class in the system, while each verb or “action,” in the requirement document is a candidate to be a function/method in the system.

USER

Website

1 0…\*

+Id

+UserId

+websiteURL

+webSitePassword

+webSiteUsername

+Id

+email: String

+password: String

+register(email, password):void

+getPasswordFromFile():String

+loginPassword(Password): void

+savePassword(Password): void

+storeUserDetail

Fig 3.3.2 class of Diagram of the system

**3.3.3 Activity Diagram**

The activity diagram defines or explains the work flow (sequence of action) of the system during program execution. It models the actions the objects performs and specify the order in which it performs them, it is model in flow chart and the activity display in the flow chart can be branched, sequential or concurrent.

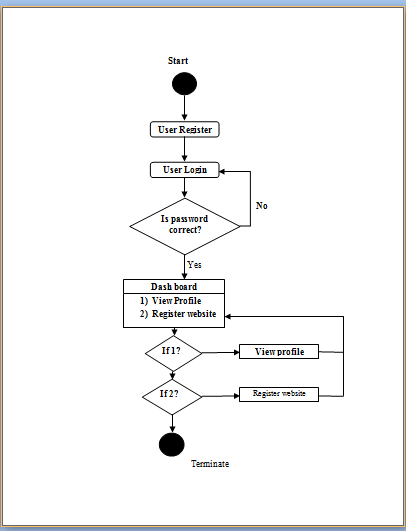


Fig 3.3.3 Class Diagram of the proposed system

**3.3.4 Input Design**

The input design is concerned with data capture and data entry. The user registers and login into the system to interact with the system with the help of the HTML form field. It allows the user to fill in the fields and click on the submit button for processing by PHP script on the server.

Email………………………………………….

Master Password………………………………

Login

Fig 3.3.4 Input design of the proposed system

**3.3.5 Output Design**

The output is the result that display after a user has submitted an input query to the system.

HOME

Add New +

Dash board

<https://unn.edu.ng>

username: Kalueze

password: Sunday30$eze

<https://facebook.com>

username: fadausk2

password: adaeze20$h

Profile

Manager Password

logout

Fig 3.3.5 Output design of the proposed system

**3.3.6 Database Design**

This design shows us how the database is been structured and how data are stored in there.

1. **User Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| FIELD | DATA TYPE | SIZE | NULL | KEY | DEFAULT | EXTRA |
| Id | Int | 11 |  | Primary Key |  |  |
| Username | Varchar | 30 | Null |  |  |  |
| Password | Varchar | 30 | Null |  |  |  |
| Phone number | Varchar | 30 | Null |  |  |  |

Table 3.3.6 user table

1. **website Table:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Int | 25 | Null | Primary Key |  |  |
| Username | Varchar | 25 | Null |  |  |  |
| Password | Varchar | 30 | Null |  |  |  |
| Email | Varchar | 30 | Null |  |  |  |
| Phone No | Varchar | 11 | Null |  |  |  |

Table 3.3.7 website table

* + 1. **System architecture**

The architecture of the system design is a 3-tier application. The tiers are the presentation tier, middle tier and data tier. The presentation tier is the user interface and it is designed using HTML, CSS and BOOSTRAP framework for the design of the graphical user interface. The middle tier which is also known as the business logic connects the presentation tier and data tier together. The middle tier of the system is design using PHP programming language, while the data tier which together with the middle tier is known as the server side or backend of a system is the part of the system that is responsible for storing the data in a database. The database management system use for the design of this system is MYSQL server. The system architecture is shown below.

Server

Middle Tier Data Tier

SQL Database

PHP

HTML5, CSS3

Presentation Tier

Figure 3.3.6 shows the system architecture

**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

**4.1.1 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.

**4.1.2 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.

**4.2 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.

**4.3 Implementation Architecture**

Home Page

Contact

Account

Registration

Login

Manage passwords

Add website

Profile

Logout

Fig 4.1: Implementation Architecture Diagram

**4.4 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.

**4.4.0 Screenshots of Some Dem**

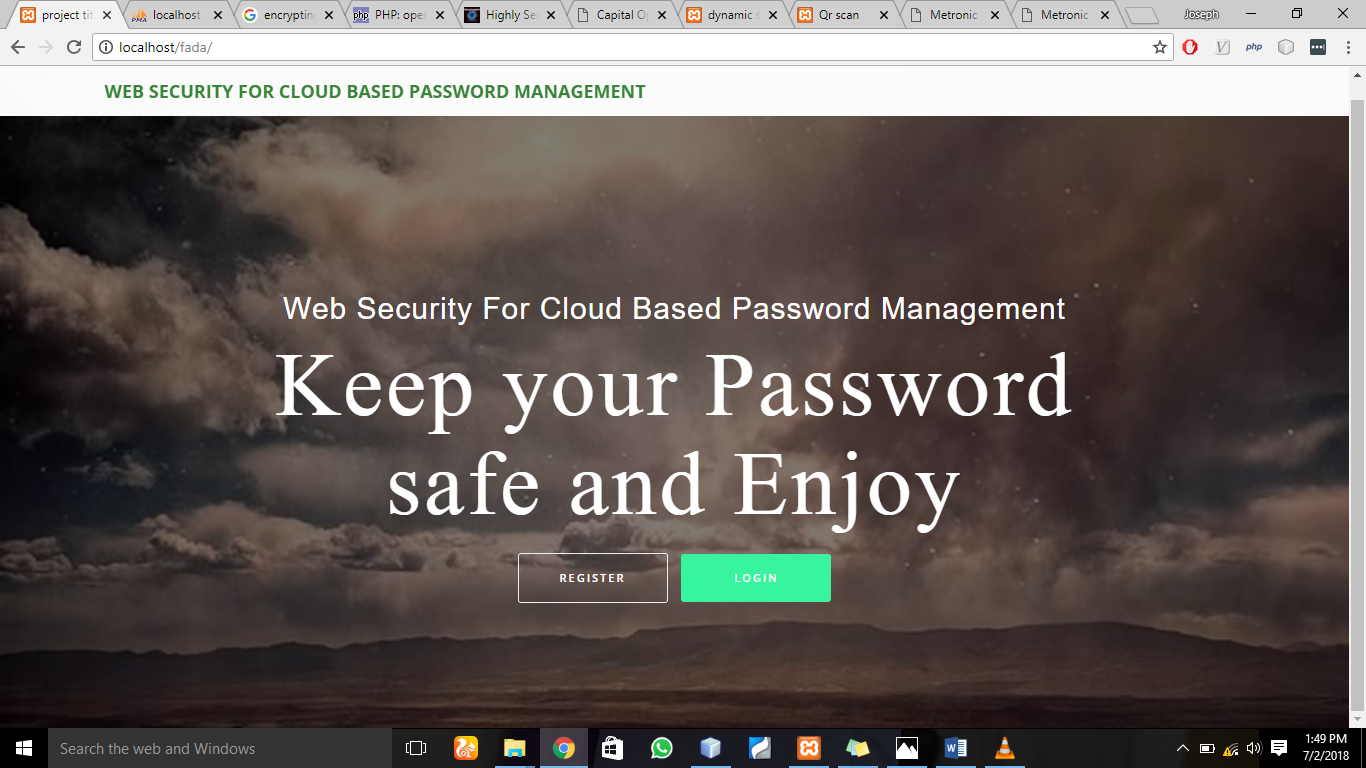
****

Figure 4.3 Screen shot of the index page.

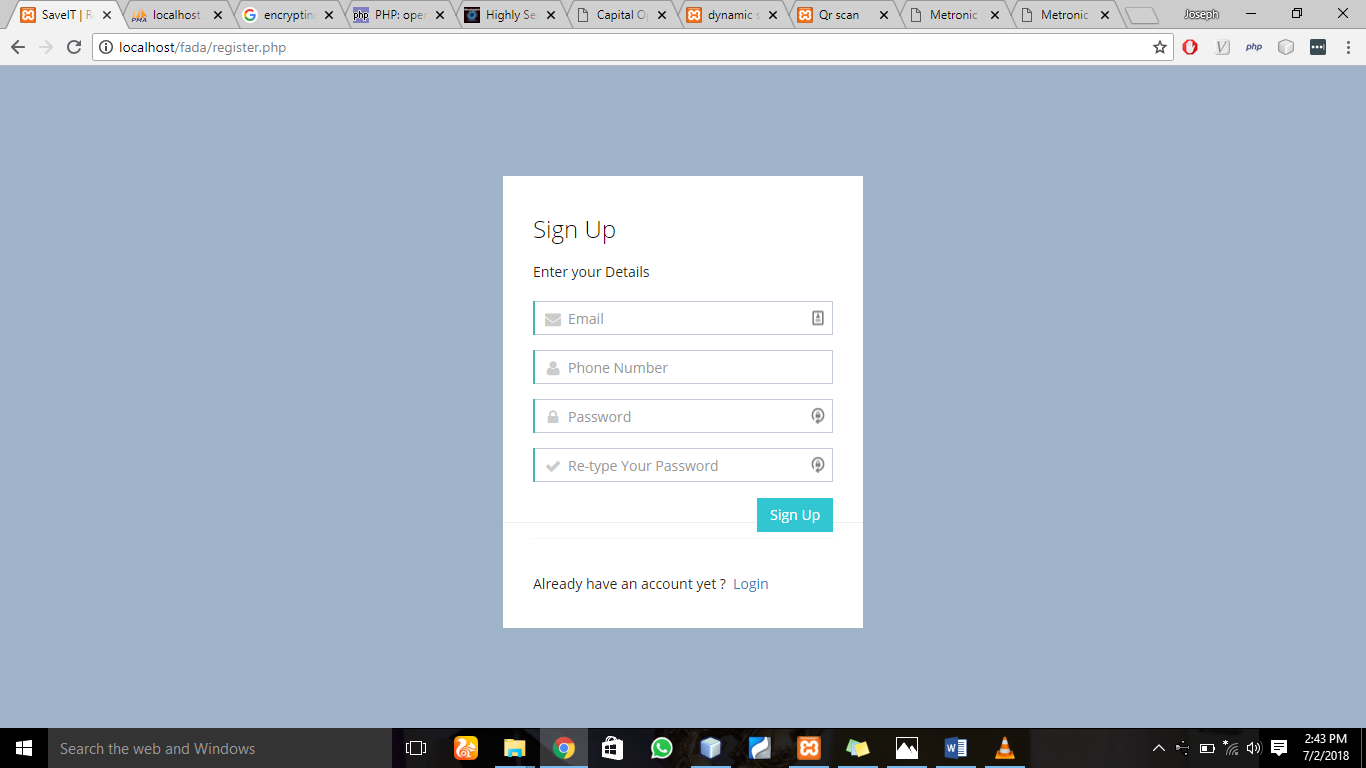
****

Figure 4.4 Screen shot of the system registration page.

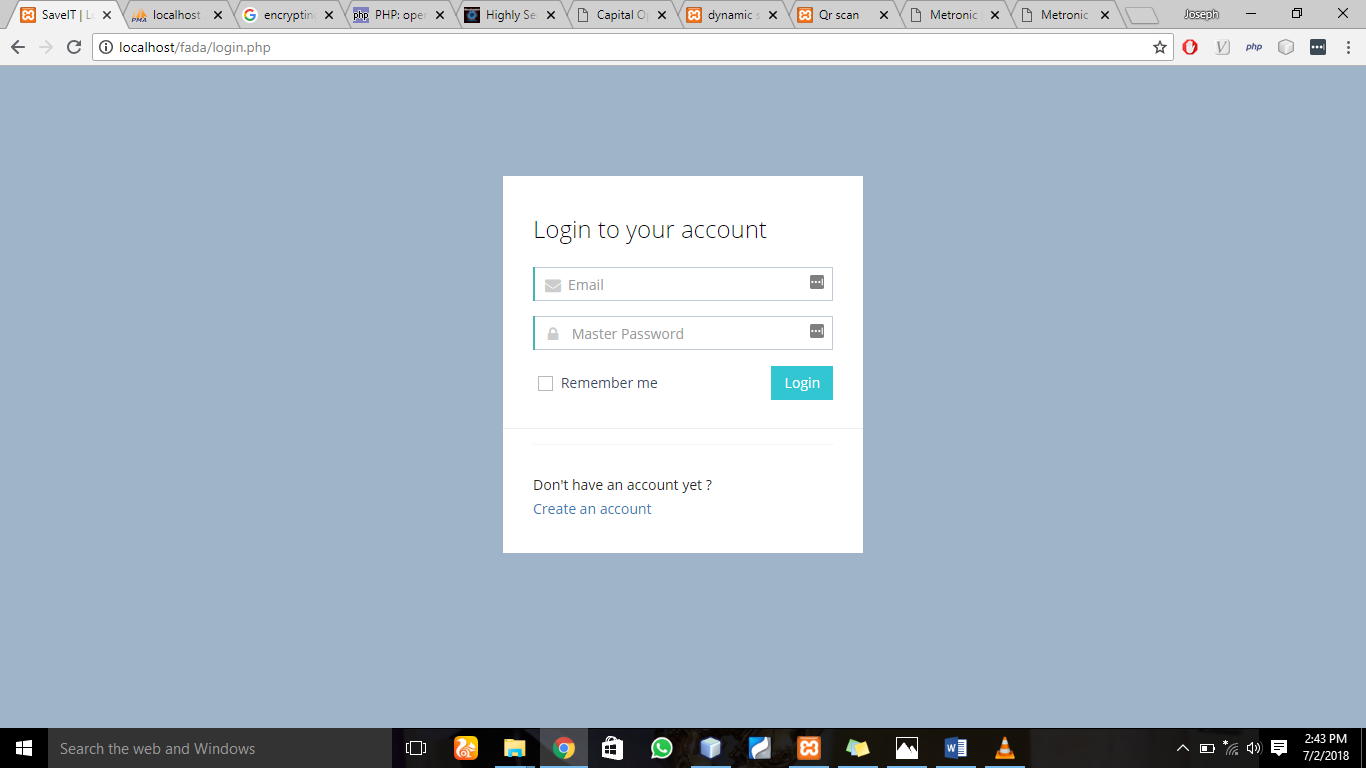
****

Figure 4.5 Screen shot of the login page.

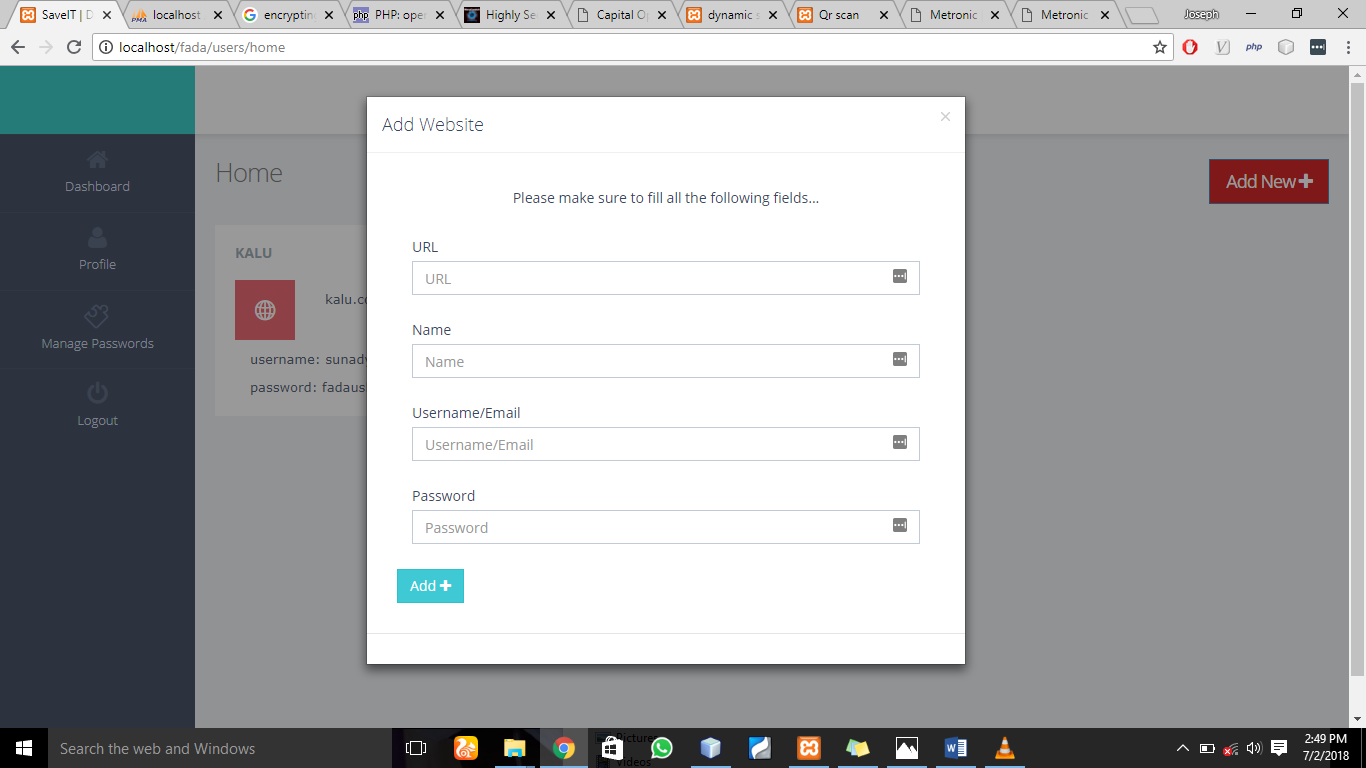
****

Figure 4.6 Screen shot of the add website page.

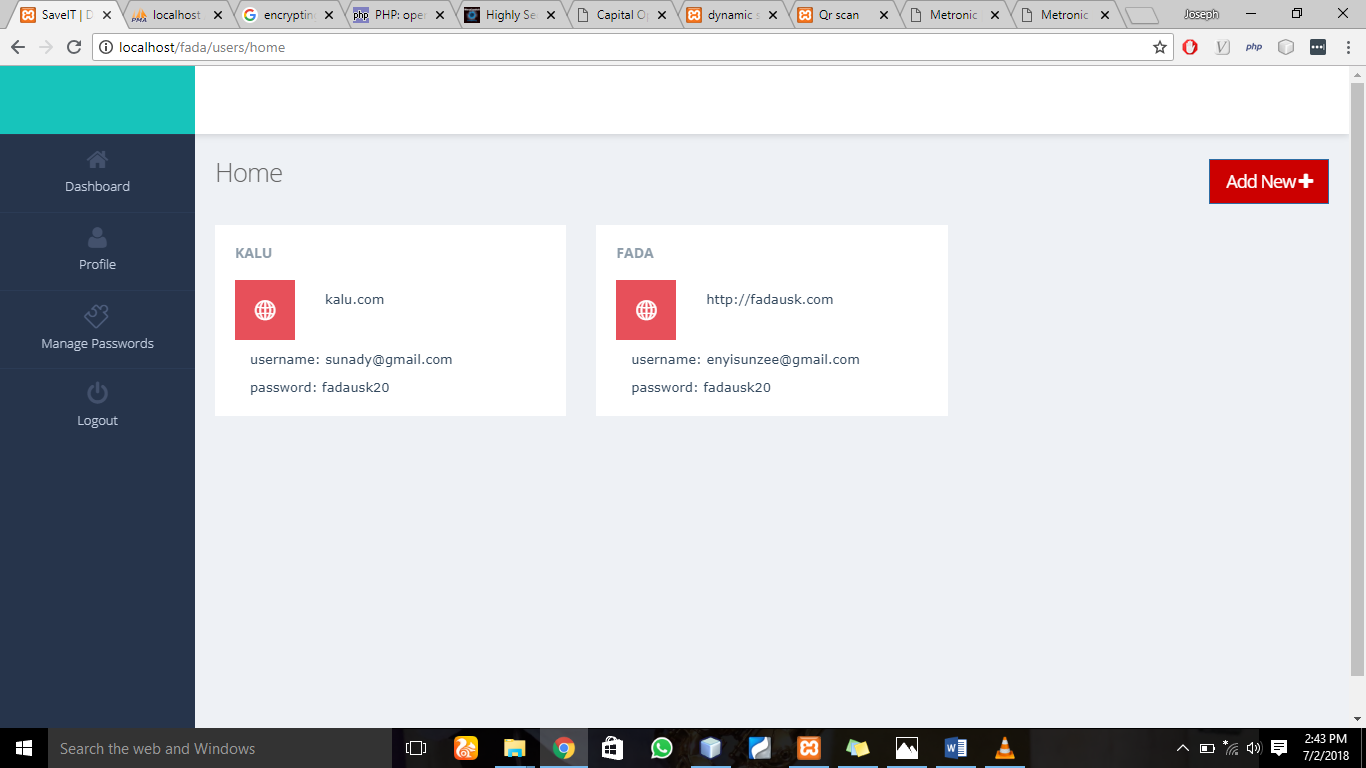
****

Figure 4.7 Screen shot of the dashboard 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

CHAPTER FIVE: **SUMMARY AND CONCLUSION**

**5.0 Summary**

This project work starts by introducing the need for password security and the problems of wrong practice of storing password on papers, human memory. It goes further to delve into the review of some related technologies currently present in the system and how their comparison with the SaveIT password security system. Furthermore, it discusses and shows the analysis of the entire system using OOAD methodology. It Unified Modelling System to represent the various diagrams to describe the activities in the system.

Conclusively, it discusses the implementation of the proposed system, and explains how the analysis and design is converted to an actual program.

**5.1 Conclusion**

This project work has been able to successfully demonstrate a web security for cloud based password that stores user’s login details and encrypt the password before sending it to the cloud database for the user. This system has also achieved increase in the password management system, and privacy of passwords.

**5.2 Recommendations**

One recommendation for this system is that functionalities like finger print and face recognition can be added to it to increase security, as it is limited in this aspect. This will go a longer way in increasing security of password.

Reference

[1] Adams, A., & Sasse, M. A. (1999). Users are not the enemy. Communications of the ACM, 42(12), 40-46.

[2] S. Komanduri, R. Shay, P.G. Kelley, M.L. Mazurek, L. Bauer, N. Christin, L.F Cranor and S. Egelman, “of passwords and people: Measuring the effect of password- composition policies , in Proc of CH1, 2011.

[3] Zhang, J., Luo, X., Akkaladevi, S., & Ziegelmayer, J. (2009). Improving multiplepassword recall: an empirical study. European Journal of Information Systems, 18, 165-176.

[4] Grawemeyer, B., & Hilary, J. (2011). Using and managing multiple passwords: A week to a view. Interacting with Computers, 23, 256-267.

[5] Rui Zhao, Chuan Yue, Kun Sun. Vulnerability and Risk Analysis of Two Commercial Browser and

Cloud Based Password Managers

[6] R. Morris, K. Thompson, Password security: a case history, Commun.

ACM 22 (11) (1979) 594–597.

[7] D. C. Feldmeier, P. R. Karn, Unix password security – ten years later,

in: Proceedings of the Annual International Cryptology Conference

(CRYPTO), 1989, pp. 44–63.

[8] A. Adams, M. A. Sasse, Users are not the enemy, Commun. ACM 42 (12)

(1999) 40–46.

[9] M. Wu, R. C. Miller, G. Little, Web wallet: preventing phishing at-

tacks by revealing user intentions, in: Proceedings of the Symposium on

Usable Privacy and Security (SOUPS), 2006, pp. 102–113.

[10]. Samlinson, E.; Usha, M., "User-centric trust based identity as a service for federated cloud environment," in Computing, Communications and

Networking Technologies (ICCCNT),2013 Fourth International Conference on , vol., no., pp.1-5, 4-6 July 2013

[11]. Yan Yang; Xingyuan Chen; Guangxia Wang; Lifeng Cao, "An Identity and Access Management Architecture in Cloud," in Computational

Intelligence and Design (ISCID), 2014 Seventh International Symposium on, vol.2, no., pp.200-203, 13-14 Dec. 2014

[12] Rui Zhao and Chuan Yue, Toward A Secure and Usable Cloud-based Password

Manager for Web Browsers, 2014, pp11-15

[13] Siponen, M. T., & Oinas-Kukkonen, H. (2007). A Review of Information Security Issues and Respective Research Contributions. The Data Base for Advances in Information Systems, 38(1), 60-80.

[14] Dhillon, G., & Backhouse, J. (2000). Information System Security Management in the New Millenium. Communications of the ACM, 43(7), 125-128.

[15] Simmonds, A., Sandilands, P., & Ekert, L. (2004). An Ontology for Network Security Attacks. Proceedings of the Second Asian Applied Computing Conference (AACC), (pp. 317-323). Kathmandu, Nepal.

[16] Siponen, M. T., & Oinas-Kukkonen, H. (2007). A Review of Information Security Issues and Respective Research Contributions. The Data Base for Advances in Information Systems, 38(1), 60-80.

[17] Hilton, J., & Cherdantseva, Y. (2013). A Reference Model of Information Assurance & Security. Eighth International Conference on Availability, Reliability and Security (ARES) (pp. 546-555). Regensburg: IEEE.