**CHAPTER ONE**

**INTRODUCTION**

**1.0 BACKGROUND OF THE STUDY**

Protecting data is at the heart of many secure systems, and many users rely on a database management system to manage the protection. Databases are essential to many business and government organizations, holding data that reengineered to make them more effective and more tunes with new and revised goals. Database security is a difficult operation that any organization should enhance in order to run its activities smoothly. The various threats pose a challenge to the organization in terms of integrity of the data and access. The threats can result from either by an outside illegal program action or by an outside force such as fire or a power failure. Most of the database contains sensitive data for users which can be vulnerable to hacking and misuse. Therefore, firms have greater control and check on their database to maintain the integrity of the information and ensure that their systems are monitored closely to avoid deliberate violations by intruders. (Sukhdev, 2013).

In the physical world, security is a fairly simple concept. If the locks on your house’s doors and windows are so strong that a thief cannot break in to steal your belongings, the house is secure. For further protection against intruders breaking through the locks, you might have security alarms. Similarly, if someone tries to fraudulently withdraw money from your bank account but the teller asks for identification and does not trust the thief’s story, your money is secure. When you sign a contract with another person, the signatures are the legal driving force that impels both parties to honor their word. (Saurabh , 2012).

In the digital world, security works in a similar way. One concept is privacy, meaning that no one can break into files to read your sensitive data (such as medical records) or steal money (by, for example, obtaining credit card numbers or online brokerage accounts information). Privacy is the lock on the door. Another concept, data integrity, refers to a mechanism that tells us when something has been altered. That’s the alarm. By applying the practice of authentication, we can verify identities. That’s comparable to the ID required to withdraw money from a bank account (or conduct a transaction with an online broker). And finally, non repudiation is a legal driving force that impels people to honor their word. (Saurabh , 2012).

**1.1 STATEMENT OF PROBLEMS**

The problem is security. The password method used in almost all commercial operating systems is probably not very strong against a sophisticated or unsophisticated attacker. The choice of data security comes next in the minds of those that want reduction of unauthorized access on confidential files or data.

Security provided by the computer operating systems come with a preset super user account and password. The super user may have a password to control network functionality, another to conduct or access nightly backups, create accounts, and so on. For a cracker, logging on to a system as the super user is possibly the best way to collect data or do damage. If the super user has not changed an operating system’s preprogrammed passwords, the network is vulnerable to attack. Most crackers know these passwords, and their first attempt to break into a network is simply to try them. If an attacker cannot log on as the super user, the next best thing might be to figure out the user name and password of a regular user. It is used to be standard practice in most Universities and colleges, and in some commercial companies, to assign every student or employee an account with user name and initial password – the password being the user name. Everyone was instructed to log on and change the password, but often, hackers and crackers logged on before legitimate users had a chance. (Shelly Rohilla, 2013).

**1.2 AIM AND OBJECTIVES**

This project is aimed at developing a secured database management system.

The specific objective of this project is to:

1. developing a secure database system.
2. implement the above objectives using PHP as programming language and MYSQL as database

**1.3 RESEARCH METHODOLOGY**

Computer Security has been defined as the art of protecting computer system and information from harm and unauthorized use. One of the most efficient methods of securing data in any database system is the use of OTP (One Time Password). OTP is a form of user validation that auto generates a random number and sends it as a secret key to the user via any of the user’s available contacts e.g email or phone number, before the user can gain access to take any action on any record in the database. This secret key sent is only known to the user. The scope of this study covers the data security, data integrity, user authentication and user validation.

The model used in this project is a waterfall model, also to implement this project a PHP programming language and MYSQL is used as developing tools.

**1.4 SCOPE OF STUDY**

Development of secured database management system was carry out in five Chapters. Chapter one contains introduction, objective and significant of the study, while chapter two give the literature review. Chapter three contains the design method on materials used Requirement for the design was also includes, while chapter four

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.0 INTRODUCTION**

All organizations depend on computerized information systems for carrying out their daily activities. All this is done with the help of a database. A database can be defined as a collection of related data, where data are facts that have an implicit meaning. For example, student roll number, name and other details can be recorded in the database. It is built to store logically interrelated data representing some aspects of the real world, which must be collected, processed, and made accessible to a given user population. The collection of software programs that provide the functionalities for defining, maintaining, and accessing data stored in a database is called a database management system (DBMS). There are three level of DBMS abstraction external level, conceptual level and internal level. DBMS should also provide the security besides access and manipulation of data.

**2.1 THREATS OF DATABASE SECURITY**

Tejashri and Raut gave their own opinion about the threats of database security. They expressed in their book that database security issues have been more complex due to widespread use. Database are a firm main resource and therefore, policies and procedure must be put into place to safeguard its security and the integrity of the data it by contains. Besides, access to the database has been become more rampant due to the internet and intranets therefore, increasing the risks of unauthorized access.

The objective of database security is to protect database from accident or intentional los. These threats pose a risk on the integrity of the data and its reliability. Database security allows or refuses users from performing actions on the database.

*Fig 2.1*

There are different threats to the database systems. Such as Excessive Privilege Abuse When users are granted database access privileges that exceed the requirements of their job function, these privileges may be abused for malicious purpose. Another threat is a weak audit trial. This is due to weakness in organizational internal system. This is due to weak deterrence mechanism. Denial of service is another problem in database security. Weak database audit policy represents a serious organizational risk on many levels. Another threat to the problem of database insecurity is weak system and procedures for performing authentication. Weak authentication schemes allow attackers to assume the identity of legitimate database users by stealing or otherwise obtaining login credentials. Strong authentication is therefore required to address these challenges.

**2.1.1 SECURITY ATTACKS**

Sukhdev Singh Ghuman stated in his book that there are different types of attack that can be carried out by an attacker after breaching through all layers of security which are:

1. **Direct Attacks:** Directly hitting the target data is known as direct attack. These attacks are accessible and successful only if the database does not accommodate any protection system.
2. **Indirect Attacks:** As its name implies indirect attacks are not directly executed on the target but data from or about the target can be collected through other transitional objects. For purpose to cheat the security system, some of the combinations of different queries are used. These kinds of attacks are difficult to track.
3. **Passive Attacks:** In this attack, attacker only inspects data present in the database and do not perform any alteration. Passive attack can be carried out in following ways:
   * 1. Static leakage: In this attack, information about database plaintext values can be attained by examining the snapshot of database at a particular time.
     2. Linkage leakage: in this information about plain text values can be achieve by linking the database values to position of those values in index.
     3. Dynamic leakage: changes performed in database over a period of time can be observed and analyzed and information about plain text values can be obtained.
4. **Active Attacks:** In active attack, actual database values are modified. These are more problematic than passive attacks because they can misguide a user. For example a user capturing wrong information by the result of a query. There are various ways of performing such kind of attack which are mentioned below:
   * 1. Spoofing – In this attack, cipher text value is replaced by a generated value.
     2. Splicing – in this, a cipher text value is replaced by different cipher text value.
     3. Replay – It is a kind of attack where cipher text value is replaced with old version previously updated or deleted.

**2.2 REVIEW OF EXIXTING WORK ON DATABASE SECURITY**

Tejashri and Raut (2014) talked about the security of database systems, as an example of how application security can be designed and implemented for specific task. The basic problems access control, exclusion of spurious data, authentication of users, and reliability.

Ashour and Mostafa (2016), According to him, in recent years, hardware capability and capacity of volumes in addition, huge uses of world wide web platform and information systems have led to adopt the relational database system as infrastructure to the data repository.

This paper addresses the relational database threats and security techniques considerations in relation to situations: threats, countermeasures (computer-based controls) and database security methods. This method deploy system or application that have functions, services and tools for data maintenance and management packed into the so called relational database management system (RDBMS). Such function contains services plus privileges for authorization to keep authorized access to the database. The system result is to address relational database threats in security techniques. The database last part is about the benefits and drawback of using either encryption inside RDBMS or outer.

Abdul and Abdulrahman (2012) explained that it is deliberate effort to protect an organization data against threaths such as international or accidental loss destruction or misuse. The objectives focuses on security issues that are associated with the database system that are often used by many firms in their operations. It has enabled business to enhance their effectiveness and efficiency in operations. The method is a backup method which ensures that the information is stored elsewhere and recovered in case of failure or attacks. He also described the requirements that are set for a robust database management system. The result is that data integrity refers to reliability and accuracy of the data that is stored and used in the system. The accuracy of the data that is stored and used in the system. This system promotes its integrity and reduces chances of compromising the data accuracy.

Saurabh and Siddhaling (2012) concluded that databases is the backbone for any type of applications. Database contains very important and confidential information so there is a chance of attacks. Various attacks on databases are discussed in this paper. Review of some important database security techniques like access control, techniques against SQLIA, encryption and data scrambling are discussed. Even some future research areas in the field of database security are also discussed in this paper.

Sohail and Irfan (2009) discussed the different security issues and its models for different types of database management system. Several proposals for discretionary and mandatory security models for the protection of conventional databases and object-oriented database systems are presented. Still, there is not a standard for designing these security models. The work presented in this paper gives a collected picture of different security issues of database; it can be extended to define, design and implement an effective security policy on a database environment and provides a consolidated view of database security.

Shelly and Pradeep (2013) explained that databases are a favorite target for attackers because of their confidential and important data. There are many ways in which a database can be compromised. There are various types of attacks and threats from which a database should be protected. In this paper, solutions of most of the threats mentioned, although some solutions are good while some are only temporary. Different types of threats are discussed in this paper.

Shivnandan and Rakesh (2014) expressed that databases form the backbone of many applications today. They are the primary form of storage for many organizations. So the attacks on databases are also increasing as they are very dangerous form of attack. They reveal key or important data to the attacker. Various attacks on databases are discussed in this paper. Review of some important database security techniques like access control, techniques against SQLIA, encryption and data scrambling are discussed. Even some future research areas in the field of database security are also discussed in this paper. This research will lead to more concrete solution for database security issue.

**2.3 TECHNIQUES FOR DATABASE SECURITY**

Ashour and Mostafa opined in his book about the techniques for database security that **Encryption** is an encoding process of sensitive data to become unreadable. Most of relational database management systems support this purpose to secure its data.

The encryption concept has four main factors that are defined as:

* An encryption key to encrypt the data (plaintext).
* An encryption algorithm with the encryption key transforms the plaintext to cipher text.
* A decryption key to decrypt the cipher text.
* A decryption algorithm with the decryption key transforms the cipher text back into the plaintext.

There are two forms of encryption techniques that are called **symmetric** and **asymmetric**. The symmetric one depends on the safe channel while exchanging the key, in addition, the key of encryption is similar to the key of decryption that is being utilized, for instance, IDEA (international data encryption algorithm). Symmetric algorithm is much faster than the asymmetric algorithm that uses two different keys (private and public keys) such as RSA (the name is derived from Ron Rivest, Adi Shamir, and Leonard Adleman). Generally, they are often used together, in which public key (asymmetric) encrypts a randomly generated encryption key, and the random key encrypts the actual message (using a symmetric algorithm). The database scheme of encryption should enhance sharing of data within the database without losing data privacy.

To improve the performance, the data should be divided into sensitive data and insensitive data. The insensitive data can be retrieved rapidly, and the sensitive data is encrypted/ decrypted using Encryption algorithms.

**Web-based database security**: the transmitted data from a server to a client must be in a secured way. The client should be authenticated such as **Host Identity Protocol (HIP).** It sets up a trusted relationship between hosts on the Internet by passing to the web server. The HIP and Web server help in authentication process.

**Log file** is an important file to monitor the processes and operations occurred online. It periodically tracks the status of operations to indicate the modification may occur when the system fails. It also integrates with the audit module to track the log file of the users to guarantee the web database security.

**Negative Database:** this process depends on adding false data to the original to make the malicious users to be confused, and only valid to legal users. It has four modules: database cache, database encryption algorithm, virtual database, and negative database conversion. The first three generates the data for the conversion to generate false data.

*Fig 2.2*

**2.4 REQUIREMENTS FOR DATABASE SECURITY**

Abdulrahman and Abdulrahman (2012) cited in their book that User authentication and identification is normally required before the user can access the database. Authentication methods are passwords, biometric readers or signature analysis devices. These are required for better management of users. The second requirements involves authorization and access controls. These are the rules that govern what access to what information. These policies govern how information is disclosed and then modified. When you look at the access controls, these are the polices that govern the authorizations. There has to be integrity and consistency in the database operations. There has to be a correct set of rules in operation which protects the database from malicious destructions. Auditing is another requirement in database. This demands that a record of actions pertaining to operations. This is necessary in order to review and exams the efficiency of the controls system and recommend for better actions (Coronel et al, 2012).

**2.5 PRINCIPLES FOR INTEGRITY AND RELIABILITY IN DATABASE SECURITY**

Databases amalgamate data from many sources, and users expect a DBMS to provide access to the data in a reliable way. When software engineers say that software has reliability, they mean that the software runs for very long periods of time without failing. Users certainly expect a DBMS to be reliable, since the data usually are key to business or organizational needs. Moreover, users entrust their data to a DBMS and rightly expect it to protect the data from loss or damage.

Data integrity refers to reliability and accuracy of the data that is stored and used in business. Data should assist a firm to make the right decision and avoid inconsistencies. Element integrity concern that the value of a specific data element is written or changed only by authorized users. Proper access controls protect a database from corruption by unauthorized users. Users trust the DBMS to maintain their data correctly, so integrity issues are very important to database security.

**CHAPTER THREE**

**RESEARCH METHODOLOGY AND SYSTEM ANALYSIS**

**3.0 SYSTEM INVESTIGATIONS AND ANALYSIS**

System investigation is concerned with the in-depth and comprehensive study carried out upon the old system so as to find out relevant facts that will be of help in designing the new system.

**3.1 SECURED DATABASE SYSTEM**

The secured database system is designed mainly for the general students of Rufus Giwa Polytechnic, Owo. All students will now have a record in the school's database through a registration portal. This will definately be part of the improvement in the institutions to ensure they have a secured record in the school's database. Only a sole administrator can gain access to view, a list of all the records in the database.

**3.2 REGISTRATION FORM FOR DATA COLLECTION**

Forms will be created online to allow students to be registered to the system. All student ranging from all programs and all levels. Students can register by providing the data that includes: full name, surname, other name, Gender, Date of Birth, State of Origin, Local Government, email, Phone number, Faculty, department, programs and level. After registration, the registered student's record will be sent and stored in the database.

**3.3 PASSPORT CAPTURE**

This is the uploading of a passport photograph as a part of the record item to be stored in the database. Each student's photograph needs to be captured in a digital file size that shouldn't exceed 200KB. The file too must be in a jpeg, jpg or png format.

**3.4 SOLE ADMINISTRATOR SECURED LOGIN PAGE**

The sole administrator is the only user that has the access to take actions on any record in the database. Before the sole administrator can gain access, he/she must first pass through a login page that will require him/her to input his/her username and password that will authenticate him/her to successfully pass through the first stage of the security system.

**3.5 SECURED OTP PAGE**

After by passing the first stage of authentication, the otp page will automatically send a secret key of 6 randomly generated digits to the registered email and phone number linked/associated with username authenticated from the secured login page. The sole administrator will be required to input the secret key sent to his/her email/phone number via sms into the provided space in the OTP page correctly (The secret key is only known to the receiver only and has a time expiration to be five minutes). Once the sole admin successfully passes through this stage, she will automatically be redirected to the dashboard of the database showing the statistics of the records in the database.

**3.6 DATABASE PAGE**

This is where a list all the records in the database is displayed. Now the sole administrator has the power to view individual records, edit individual records and as well delete individual records. The sole administrator can also send a private email to student's email containing individual's personal records like student's registration number.

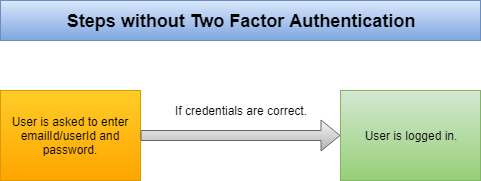
**3.7 OBJECTIVES OF THE NEW SYSTEM**

A lot of benefits are derived from the proposed system, they include:-

* Confidentiality of data.
* Data integrity.
* Message authentication.
* User authentication and validation with OTP.
* Preventing secret key from unauthorized person.

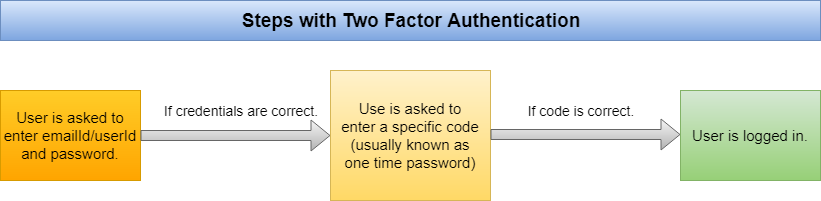
**3.8 ALGORITHM**

Two-factor authentication (or multi factor authentication) is just an extra layer of security for a user’s account. That means that, after enabling two factor authentication, the user has to go through one more step to log in successfully. For example, the usual steps for logging in to an account are:



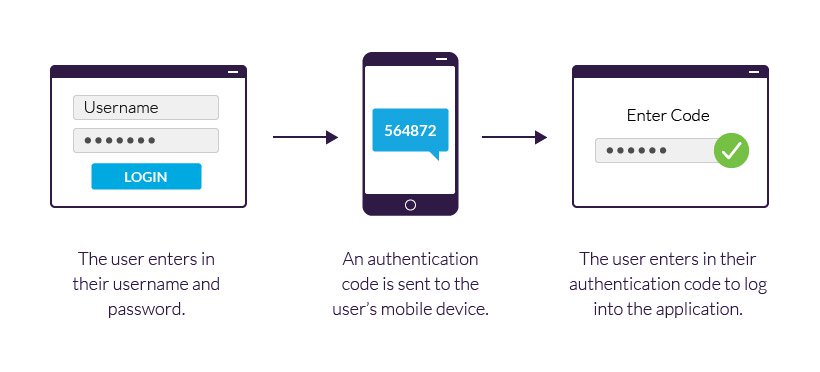
*fig 3.1*

But after enabling 2-factor authentication, the steps look something like this:



*fig 3.2*

So this adds one more step to the login process. This method is more secure, because a criminal cannot access the user’s account unless they have access to both the user’s regular password and one time password.



*Fig. 3.3*

**CHAPTER FOUR**

**IMPLEMENTATION AND RESULT**

**4.0 CHOICE OF PROGRAMMING LANGUAGE**

The programming language used in the development of this project is PHP Hypertext Program (PHP). The PHP programming language is one of those intermediate languages that Programmers use to create web applications and has a .php file extension.

**PHP IS:**

**Simple:** It is very simple and easy to use, compared to another scripting language it is very simple and easy, this is widely used all over the world.

**Interpreted:** It is an interpreted language, i.e. there is no need for compilation.

**Faster:** It is faster than other scripting languages e.g. asp and jsp.

**Open Source:** Open source means you no need to pay for using PHP, you can free download and use.

**Platform Independent:** PHP code will be run on every platform, Linux, Unix, Mac OS X, Windows.

**Case Sensitive:** PHP is case sensitive scripting language at the time of variable declaration. In PHP, all keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are NOT case-sensitive.

**Speed Comparision of ASP PHP JSP:** PHP is faster than other scripting languages e.g. asp and jsp.

**4.1 PROGRAM DEVELOPMENT**

**PROGRAM DESIGN**

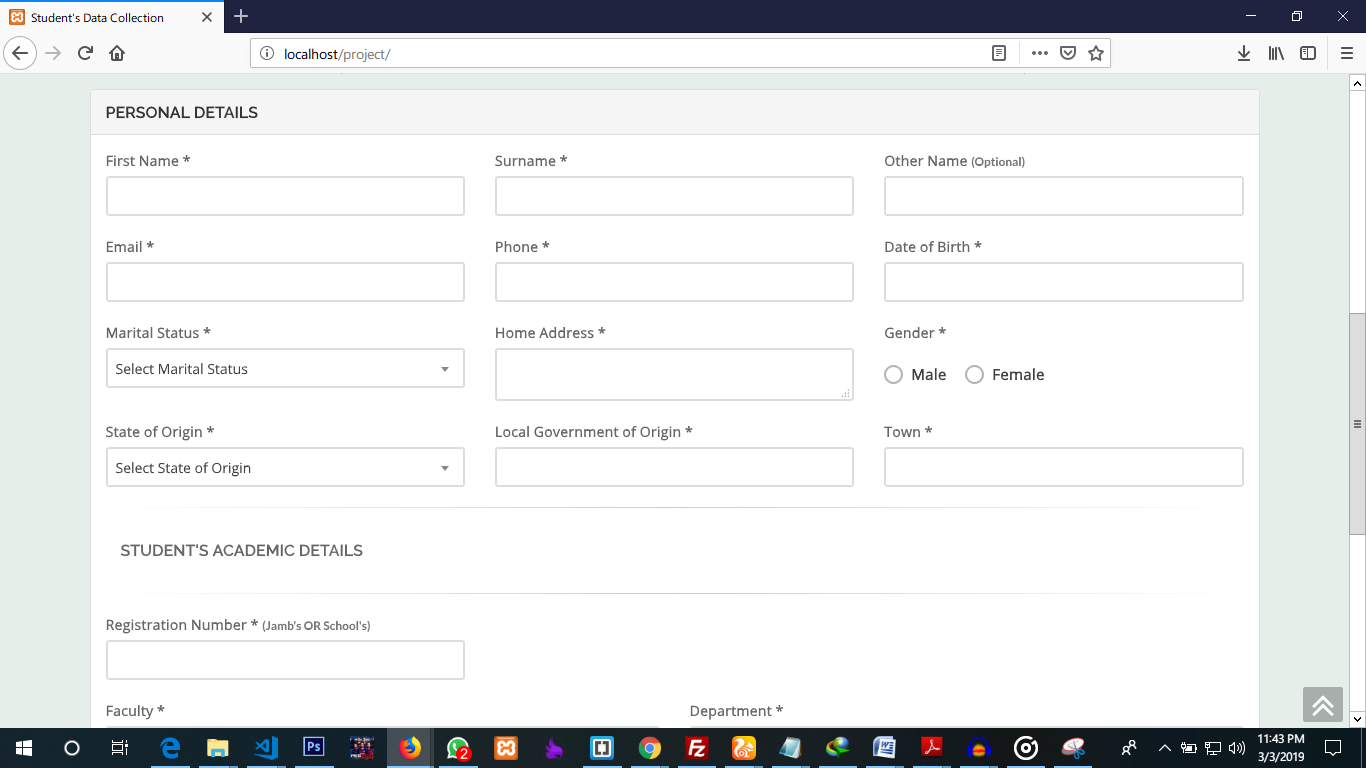
The program for the new system was coded using PHP. One of the most intermediate languages that programmers use to create web applications. A php file carries the extension .php. It is flexible, powerful, easier to use, object oriented, internet friendly and secure. The application have been designed and stored on a web server. The program code can be seen in the appendix. To view the designed application, logon to [www.rugipodatacollection.tk/admin](http://www.rugipodatacollection.tk/admin) and login with the below details.

A database system was developed for students of Rufus Giwa polytechnic, Owo to help keep students records secured. The database used is MySql.

**INPUT DESIGN**

The design displays the e-registration form that will input collected data into the database. Registration form for inputing student’s data is shown in fig.

*Fig. 4.1*



*Fig. 4.1.1*

Start

Username

& Password

Correct

Username

& Password?

OTP

Correct

OTP ?

Dashboard

Database

1

1

NO

YES

NO

YES

*Fig. 4.2*

1

1

Edit

View

Delete

Email

Stop

*Fig. 4.2.1*

**4.2 SYSTEM IMPLEMENTATION**

1

1

The purpose of implementation phase is to interpret or translate the software into source code. Each component of the design is implemented as a program module. The end-product of this phase is a set of program module that has been individually tested. Each module is unit tested to determine the correct working of all the individual modules. It involved testing each module in isolation as this is the most efficient way to debug the error identified at this stage.

**4.2.1 ACCESSING THE SOFTWARE FOR PROPER CONFIGURATION**

In this phase, the system design depicted in form of diagram and model in chapter three is converted into tangible software in form of coding. All program codes are implemented using PHP programming language. System component shown in the system architecture in chapter three earlier are implemented one after the other in an object oriented fashion using web technology tools such as PHP as the integrated development environment and MySQL as the Database Management System.

**4.3 HARDWARE AND SOFTWARE REQUIREMENT**

**The hardware requirements are:**

1. 320GB Hard Disk Drive or higher
2. 2GB RAM or higher
3. Internet Network Tool such as Modem, Router, Mobile Hotspot etc
4. Monitor
5. Keyboard and Mouse
6. Uninterrupted Power Supply & Stabilizer
7. Printer

**The software requirements are:**

1. Operating system and utilities: Window 7, 8 or higher
2. Web server software (XAMPP or WAMP)
3. Web browsers such as Mozilla Firefox, Netscape, Opera Mini etc.
4. Anti-virus and Anti-malware

**4.4 SYSTEM TEST RUNNING**

The system test running was done to ensure the system integrity, efficiency and effectiveness are achieved. Testing for errors in the program is necessary in order to find errors that may occur in the language errors, logic errors and error analysis program. Tests performed on the system include the entire menu of programs, processes applications, and preparing reports. The system test running was successful and the units upon which the testing was done include the following:

1. Sole admin secure Login testing
2. Secure OTP testing with email
3. Input, delete and edit student’s data testing
4. Send individual details/ Records to students
5. Testing of data searching for student record
6. Logout testing

**4.5 SYSTEM INTERFACES**

Interfaces provide interaction. The system interface allows the system users to interact effectively with the system. The main interfaces are categorized as follow:

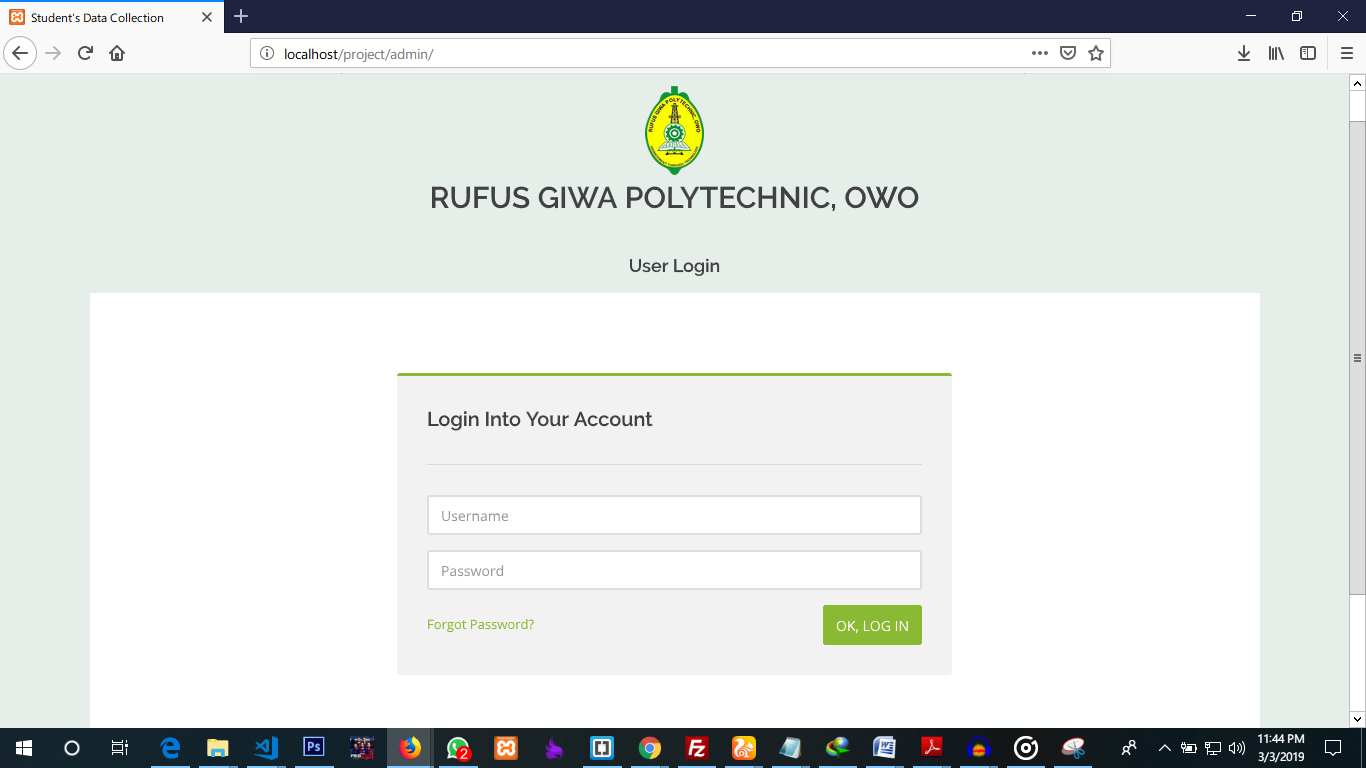
1. Student’s Rgistration Interface
2. Sole Administrator’s Secured Login Interface

**4.6 STUDENT’S REGISTRATION INTERFACE**

The students’ registration interfacae is where every student will input his/her required details to be inputed into the database. Details required includes personal and academic details.

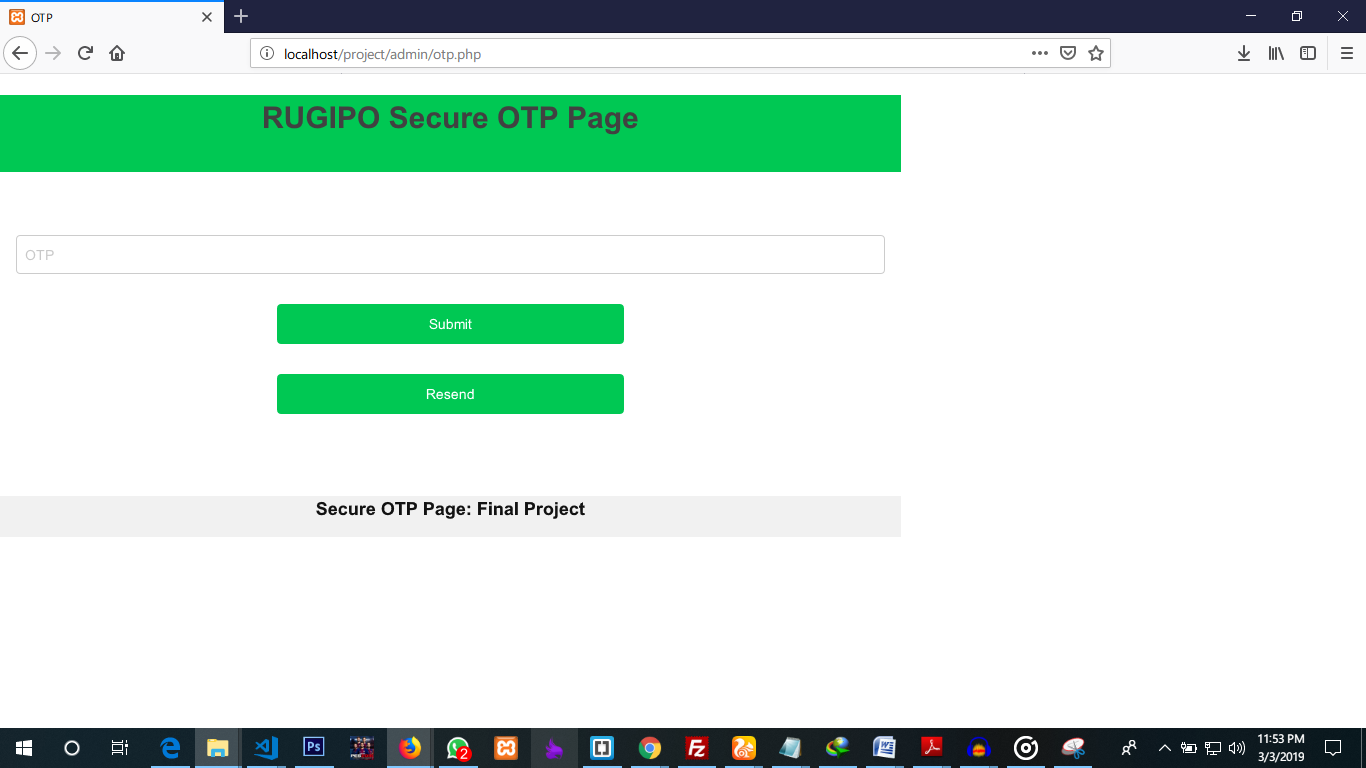
**4.6.1 HOMEPAGE INTERFACE**

This interface a secure login page that will require the sole administrator to input his / her username and password before he can login successfully login. The secure login page is shown in figure 4.1 below:



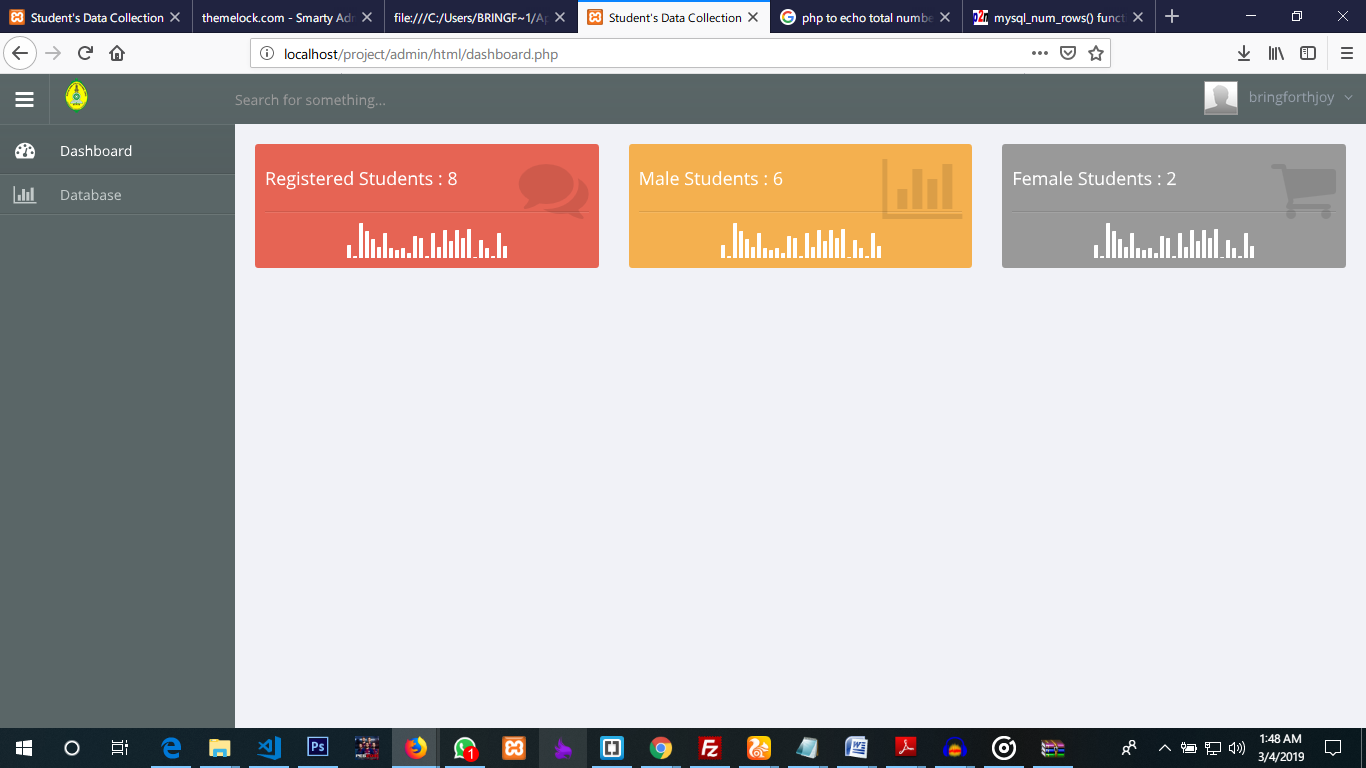
*Fig. 4.3 (Login page)*

The OTP authentication page for admin:

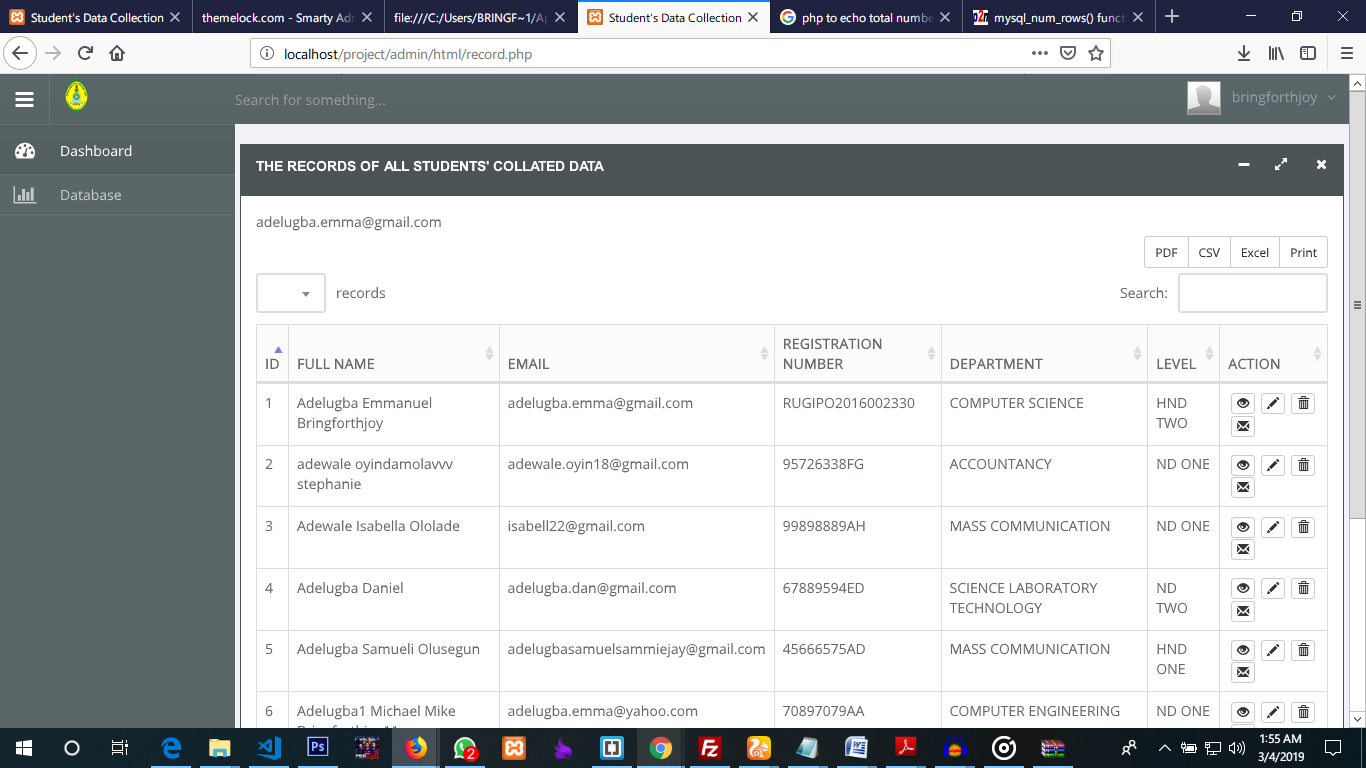


*Fig. 4.4 (OTP Authentication Page)*

After the admin successfully input the correct OTP, he will be redirected to the admin dashboard: where he can see the statistics of all the record he has in the database



*Fig. 4.5 (Admin Dashboard)*

The ‘database’ menu: this will open the page were all the records in the database are listed.

*Fig. 4.6 (Database Page)*

**CHAPTER FIVE**

**SUMMARY, CONCLUSION AND RECOMMENDATION.**

**5.0 SUMMARY**

In the course of the research of my project, I have discovered the use of OTP to verify Users before they have access to a database has made the database secure, hence, preventing unauthorized users from gaining access to the database. The system was developed in such a way that two phases of authentication will be required before the admin/user will gain access to the database system. Whenever an admin/user successfully passes through the first phase of the login where he/she is required to input his/her login details, the system automatically sends a 6 digits password (OTP) that is automatically and randomly generated to the email/ phone number linked with the particular user logged in from the first login phase. After this point, the user is then required to enter/input his OTP to successfully gain access to the dashboard and the database.

**5.1 CONCLUSION**

The use of a two way authentication factor increased the level of security for the database. Before any unauthorized user can gain access to the database, he/she must have both a user’s password and the OTP as well which will almost be impossible to get.

Therefore, it is evident that this security algorithm is very strong and difficult to break into. It was used in the development of this project and was tested in various ways and of which it was almost impossible for unauthorized users/hackers to break into.

**5.2 RECOMMENDATION**

Having presented all that is needed for the successful implementation of this project, The following recommendations are suggested by the researcher aim at improving / correcting some lapses.

1. Developers and engineers need to understand the use of OTP in order to effectively build it into their products.
2. Sales and marketing people need to understand the use of OTP in order to prove the products details in their database that are to be sold are secure.
3. Learning institutions need to secure their database against unauthorized users from gaining access and manipulating students’ data.
4. IT professionals need to understand the use of OTP in order to deploy it properly in their systems.
5. Those in the banking/Accounts sector can implement the use of OTP to ensure all their transactions are secure.

**REFERENCE**

Abdulrahman & Abdulrahman (2012) *Security in Database Systems, global journal of computer science and technology network, web & security vol 2 issue 17 version 1.0,2012 pg 1-7*

Ashour A. (2016), Security of Relational Database Management System: Threats and Security Techniques. Infrastructure University Kuala Lumpur, Malaysia. Pg. 1-6

Burtescu, E. (2009). Database Security-Attacks and Control Methods. *Journal of Applied Quantitative Methods*, *vol.4, no.4 pg 449-454*

Kayarkar, H. (2012). Classification of Various Security Techniques in Databases and their Comparative Analysis. *vol.8, pg 15-25*

Sukhdev S. (2013), Database Security. SBDSM Khalsa College Domeli (Kapurthala) IJCSMC Vol. 2, Feburary 2013 Issue 2, Pg. 102-105

Mr. Saurabh Kulkarni, Dr. Siddhaling Urolagin (2012) Review of Attacks on Databases and Database Security Techniques. *International Journal of emerging technology and advanced Engineering, ISSN,Volume 2, Issue 11. 242, Pg 2250-2459*

Mubina Malik and Trisha Patel (2016) Database Security - Attacks And Control Methods. Cmpica, Charotar University of Science & Technology (CHARUSAT), Changa.

Sohail IMRAN, Dr Irfan Hyder (2009) Security Issues in Database. Second International Conference on Future Information Technology and Management Engineering.

Shelly R. (2013) Database Security: Threats and Challenges, *International journal (Advanced research in computer science and software Engineering,Volume.5(3),2014*

Shivnandan S. (2014) A Review Report on Security Threats on Database

Tejashri R. Gaikwad1, A. B. Raut (2014) A Review on Database Security. Amravati, Maharashtra*, India, International journal of science and research pg 1-3*

**APPENDIX**

dbase.php

<?php

//db details

$dbHost = 'localhost';

$dbUsername = 'root';

$dbPassword = '';

$dbName = 'project';

//conneting to the database and the table

$con = mysqli\_connect("$dbHost","$dbUsername","$dbPassword","$dbName") or die('Could not connect:'.mysql\_error());

?>

otp\_process.php

<?php

session\_start();

$rndno=rand(100000, 999999);//OTP generate

$message = urlencode("otp number.".$rndno);

$to=$\_POST['email'];

$subject = "OTP";

$txt = "OTP: ".$rndno."";

$headers = "From: admin@rugipo.costandcapitalconsult.com" . "\r\n" .

"CC: adelugba.emma@yahoo.com";

mail($to,$subject,$txt,$headers);

if(isset($\_POST['btn-save']))

{

//$\_SESSION['name']=$\_POST['name'];

$\_SESSION['email']=$\_POST['email'];

//$\_SESSION['phone']=$\_POST['phone'];

$\_SESSION['otp']=$rndno;

header( "Location: otp.php" );

} ?>

otp.php

<?php

session\_start();

//include 'session.php';

if(isset($\_POST['save']))

{

$rno=$\_SESSION['otp'];

$urno=$\_POST['otpvalue'];

if(!strcmp($rno,$urno))

{

echo "<p>Thank you for show our Demo.</p>";

header ("location: html/dashboard.php");

//For admin if he want to know who is register

}

else{

echo "<p>Invalid OTP</p>";

}

}

//resend OTP

if(isset($\_POST['resend']))

{

$message="<p class='w3-text-green'>Sucessfully send OTP to your mail.</p>";

$rno=$\_SESSION['otp'];

$to=$email;

$subject = "OTP";

$txt = "OTP: ".$rno."";

$headers = "From: adelugba.emma@gmail.com" . "\r\n" .

"CC: adelugba.emma@yahoo.com";

//mail($to,$subject,$txt,$headers);

$message="<p class='w3-text-green w3-center'><b>Sucessfully resend OTP to your mail.</b></p>";

}

?>

<!DOCTYPE html>

<html>

<header>

<title>OTP</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="http://studentstutorial.com/div/d.css">

<!-- CORE CSS -->

<link href="assets/plugins/bootstrap/css/bootstrap.min.css" rel="stylesheet" type="text/css" />

<!-- THEME CSS -->

<link href="assets/css/essentials.css" rel="stylesheet" type="text/css" />

<link href="assets/css/layout.css" rel="stylesheet" type="text/css" />

<!-- PAGE LEVEL SCRIPTS -->

<link href="assets/css/header-1.css" rel="stylesheet" type="text/css" />

<link href="assets/css/color\_scheme/green.css" rel="stylesheet" type="text/css" id="color\_scheme" />

<style>

a{

text-decoration:none;

}

</style>

<header>

<body>

<br>

<div class="row">

<div class="col-md-8">

<div class="w3-container w3-center w3-green">

<h2>RUGIPO Secure OTP Page</h2>

</div>

<br>

<form class="w3-container" method="post" action="">

<br>

<br>

<p><input class="w3-input w3-border w3-round" type="password" placeholder="OTP" name="otpvalue"></p>

<p class="w3-center"><button class="w3-btn w3-green w3-round" style="width:40%;height:40px" name="save">Submit</button></p>

<p class="w3-center"><button class="w3-btn w3-green w3-round" style="width:40%;height:40px" name="resend">Resend</button></p>

</form>

<div><?php if(isset($message)) { echo $message; } ?>

</div>

<br>

<div class="w3-container w3-center w3-light-grey">

<h4><a href="rugipo.com.edu">Secure OTP Page: Final Project</a></h4>

</div>

</div>

<div class="w3-half">

</div>

</div>

</body>

</html>

otp\_process.php

<?php

session\_start();

$rndno=rand(100000, 999999);//OTP generate

$message = urlencode("otp number.".$rndno);

$to=$\_POST['email'];

$subject = "OTP";

$txt = "OTP: ".$rndno."";

$headers = "From: admin@rugipo.costandcapitalconsult.com" . "\r\n" .

"CC: adelugba.emma@yahoo.com";

mail($to,$subject,$txt,$headers);

if(isset($\_POST['btn-save']))

{

//$\_SESSION['name']=$\_POST['name'];

$\_SESSION['email']=$\_POST['email'];

//$\_SESSION['phone']=$\_POST['phone'];

$\_SESSION['otp']=$rndno;

header( "Location: otp.php" );

} ?>