

SIMPLE BANKING SYSTEM USING PHP

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By
HANOK KUMAR.S (Reg. No. 39290028)

TONY YESU DASS.B (Reg. No. 39290114)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

**SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

APRIL 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI– 600119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **HANOK KUMAR.S (Reg. No. 39290028)** and **B.TONT YESU DASS (Reg. No. 39290114)** who carried out the project entitled “**SIMPLE BANKING SYSTEM USING PHP**” under my supervision from November 2020 to April 2021.

Internal Guide

Dr. MERCY PAUL SELVAN, M.E., Ph.D.,

Head of the Department

Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **HANOK KUMAR.S (Reg. No. 39290028)/TONY YESU DASS.B(39290114)**

hereby declare that the Project Report entitled" **SIMPLE BANKING SYSTEM
USING PHP**" done by me

under the guidance of **Dr. MERCY PAUL SELVAN** is submitted in partial
fulfillment of the requirements for the award of Bachelor of Science degree in
Computer Science.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D.**, Dean, School of Computing and **Dr. S. VIGNESHWARI, M.E., Ph.D.**, and **Dr. L. LAKSHMANAN, M.E., Ph.D.**, Head of the Department, Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. MERCY PAUL SELVAN, M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways For the Completion of the project.

ABSTRACT

The content of an abstract must answer or address the needs of every issue happen in bank management system project. These issues could be lacking of security in manual managing of bank accounts or to address the efficiency of banking transactions by making the management system an online project. For example is:

To create a project for resolving a customer's financial applications in a banking environment in order to meet the needs of an end banking user by giving multiple ways to accomplish banking chores. Also, to provide additional features to the user's work space that aren't available in a traditional banking project.

This project abstract for Bank Management System is important because you can foresee the function of the system upon creating it. Next to the abstract, we'll discuss the modules of this system.

TABLE OF CONTENTS

Chapter No.	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
	1.1 OVERVIEW OF PROJECT	1
2	LITERATURE SURVEY	2
3	AIM AND SCOPE OF PRESENT INVESTIGATION	5
	3.1 AIM OF THE PROJECT	5
	3.2 SCOPE AND OBJECTIVE	5
	3.3 SYSTEM REQUIREMENTS	6
	3.3.1 HARDWARE REQUIREMENTS	6
	3.3.2 SOFTWARE REQUIREMENTS	6
	3.4 SOFTWARE USED	7
	3.4.1 PHP/WHAT IS PHP?	7
	3.4.2 WHAT CAN PHP DO	7
	3.4.3 PHP CHARACTERISTICS	7
	3.4.4 APPLICATIONS OF PHP	8
	3.5 WHAT IS XAMPP?	9
	3.5.1 COMPONENTS OF XAMPP	9
	3.5.2 INSTALLATION PROCESS OF XAMPP	11
	3.5.3 TESTING XAMPP INSTALLATION	12
	3.5.4 XAMPP CONTROL PANEL	12

4	EXPERIMENTAL OR MATERIAL METHODS	14
	4.1.1 USED AND PURPOSE	14
	4.1.2 OBJECTIVE OF THE PROJRCT	14
	4.2 DESING CONCEPT	14
	4.2.1 BANKING SYSTEM	15
5	RESULTS AND PERFORMANCE ANALYSIS	21
	5.1 SNAPSHOT OF PAGES	
	5.2 FEATURES	22
6	CONCLUSION AND FUTURE ENHANCEMENT	23
	6.1 CONCLUSION	23
	REFERENCES	24
	APPENDIX	26
	A. SOURCE CODE	26

LIST OF FIGURES

FIGURE NO:	FIGURE NAME	PAGE NO
3.5.3	XAAMP	12
5.1	TRANSACTIONS DETAILS	21
5.2	ADMIN HOME PAGE	22
5.3	TRANSFER FUND PAGE	22
5.4	ADMIN TRANSACTIONS LIST PAGE	23
5.5	CLIENT DASHBOARD	24
5.6	CLIENT TRANSACTIONS	25

CHAPTER 1

1. INTRODUCTION

1.1 OVERVIEW OF PROJECT

The main objective of the project is to develop online Banking system for banks. In present system all banking work is done manually. User have to visit bank to Withdrawal or Deposit amount. In present bank system it is also difficult to find account information of account holder. In this bank management system we will automate all the banking process. In our bank management system user can check his balance online and he can also transfer money to other account online. In this Software you can keep record for daily Banking transactions. The main purpose of developing bank management system is to design an application, which could store bank data and provide an interface for retrieving customer related details with 100% accuracy.

This bank management system also allow user to add new customer account, delete account and user can also modify existing user account information. Using this system user can also search any individual account in few seconds. Using our bank management system user can also check any translation in any account. Our system also provide security check to reduce fraud. The system will check the user's existence in the database and provide the set of services with respect to the role of the user.

CHAPTER 2

2. LITERATURE SURVEY

2.1 Existing System of Banking Management

The existing bank system is slow as every task is being performed by the human being and comparing the computer task speed with a computer is not fair. The complexity of this system is increased when an increase in the number of customers and with that there will be a number of transactions will be performed now everything needs to log in to a file for reference in the future which is simply not the kind of scenario we need at this time.

Some improvements by executing the proposed system:

- More secure information will give a layer of security of authentication and authorization.
- Required very little manpower.
- Simplify the problem of editing.
- Maintain the clearance level by the hierarchy.

The information will be secure from the different types of disasters as there will be an automatic backup system for the customer and bank information.

- Maintain data integrity Validate the manual calculations avoid calculation error.
- Safeguard the data accuracy.
- More reliable and efficient.
- More user-friendly interface.

Features of Branch Module:

- Admin can manage the branch
- Admin can edit/delete the branch
- Admin can see the list of all branch
- Customer can see branch

Customer Module :

The main aim for developing this module is provide all the functionality related to customer. It tracks all the information and details of the customer. We have developed all type of CRUD (Create, Read, Update and Delete) operations of the customer. This is a role based module where admin can perform each and every operations on data but the customer will be able to view only his/her data, so access level restrictions has also been implemented on the project. Here students can get php projects with database free download.

Features of Customer Module:

- Admin can add new customer
- Admin can see the list of customer details
- Only admin can edit and update the record of the customer
- Admin will be able to delete the records of the customer
- All customer forms are validated on client side using JavaScript
- Customer will be able to see his details
- Customer will be able to update his details

Account Module :

The main aim for developing this module is to manage the account data wise. All accounts will be managed by admin and customer will be able able to see account. Admin can see the list of all the account and filter it according to the customers. We have developed all type of CRUD (Create, Read, Update and Delete) operations of the account.

Features of Account Module:

Admin can manage the account
Admin can edit/delete the account
Admin can see the list of all account
Customer can see account

Account Type Module :

The main objective for developing this module is to manage the account type. So all account company will be managed by admin and customer will be able to see the account type. We have many collections of free download php projects with source code.

Features of Account Type Module:

Admin can manage the account type
Admin can edit/delete the account type
Admin can see the list of all account type
Customer can see the account type

CHAPTER 3

3. AIM AND SCOPE OF PRESENT INVESTIGATION

3.1 AIM OF THE PROJECT

The main objective for developing this module is to manage the branch of the bank. This Branch module is an important module in this project Online Banking System which has been developed on PHP and MySQL. Here we are providing the best php projects free download with source code and executable file. So all branch will be managed by admin and customer will be able to branch. Admin can see the list of all the branch area and filter it according to the customers. Customer can see all the branch details.

3.2 SCOPE AND OBJECTIVE

- Classification of genre can be very valuable to explain some interesting problems such as creating song references, tracking down related songs, discovering societies that will like that specific song, sometimes it can also be used for survey purposes.
 - Automatic musical genre classification can assist humans or even replace them in this process and would be of a very valuable addition to music information retrieval systems. In addition to this, automatic classification of music into genres can provide a framework for development and evaluation of features for any type of content-based analysis of musical signals
1. Developing a machine learning model that classifies music into genres shows that there exists a solution which automatically classifies music into its genres based on various different features, instead of manually entering the genre.
 2. Another objective is to reach a good accuracy so that the model classifies new music into its genre correctly
 3. This model should be better than at least a few preexisting models.

3.3 SYSTEM REQUIREMENTS

3.3.1 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. The minimal hardware requirements are as follows,

- ❖ Hard disk: 500 GB and above.
- ❖ Processor: i3 and above.
- ❖ Ram : 4GB and above.

3.3.2 Software Requirements

Software requirements deals with defining resource requirements and prerequisites that needs to be installed on a computer to provide functioning of an application. The minimal software requirements are as follows,

- **Download and Install any local web server such as XAMPP/WAMP.**

Tools :

- Download XAMPP

Server :

- Chrome/Mozilla fire fox

3.4 SOFTWARE USED:

3.4.1 PHP/What is PHP?

- PHP files can contain text, HTML, CSS, JavaScript, and PHP code
- PHP code is executed on the server, and the result is returned to the browser as plain HTML
- PHP files have extension ".php"

3.4.2 What Can PHP Do?

- PHP can generate dynamic page content
- PHP can create, open, read, write, delete, and close files on the server
- PHP can collect form data
- PHP can send and receive cookies
- PHP can add, delete, modify data in your database
- PHP can be used to control user-access
- PHP can encrypt data

With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

Why PHP?

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
 - PHP is compatible with almost all servers used today (Apache, IIS, etc.)
 - PHP supports a wide range of databases
 - PHP is free. Download it from the official PHP resource: www.php.net
 - PHP is easy to learn and runs efficiently on the server side
-
- PHP 7 has improved Error Handling
 - PHP 7 supports stricter Type Declarations for function arguments
 - PHP 7 supports new operators (like the spaceship operator: `<=>`)

3.4.2 PHP Characteristics

- All variables in PHP are denoted with a leading dollar sign (\$).
- The value of a variable is the value of its most recent assignment.
- Variables are assigned with the = operator, with the variable on the left-hand side and the expression to be evaluated on the right.
- Variables can, but do not need, to be declared before assignment.
- Variables in PHP do not have intrinsic types – a variable does not know in advance whether it will be used to store a number or a string of characters.
- Variables used before they are assigned have default values.
- PHP does a good job of automatically converting types from one to another when necessary.
- PHP variables are Perl-like.

3.4.3 Applications of PHP

There is a wide range of ***application of php programming language*** in several digital platforms. Learning php is not the only thing, one should know how to deploy its several functions correctly in a particular application and what are the different fields where it is applicable.

Php programming language is not only limited to write server side scripts but it can also be used to develop several GUI(Graphic User Interface) based applications and creating Pdf files, php mailing lists, picture gallery etc.

Infact, there are many php frameworks which helps to build dynamic web pages and web applications such as Laravel, CakePhp, CodeIgniter etc.

These php frameworks have several built in mechanisms which helps to create pdf files, php mailing lists, picture gallery, templates, plugins etc.

Now we are going to provide you some of the important applications of php programming language in different fields with their brief information.

3.5.1 What is XAMPP?

XAMPP is an abbreviation where ***X stands for Cross-Platform, A stands for Apache, M stands for MYSQL***

, and the Ps stand for PHP and Perl, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, MariaDB, PHP, and Perl.

XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl

is a programming language used for web development, PHP is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL. The detailed description of these components is given below.

3.5.2 Components of XAMPP

As defined earlier, XAMPP is used to symbolize the classification of solutions for different technologies. It provides a base for testing of projects based on different technologies through a personal server. XAMPP is an abbreviated form of each alphabet representing each of its major components. This collection of software contains a web server named **Apache**, a database management system named **MariaDB** and scripting/programming languages such as **PHP** and **Perl**. X denotes Cross-platform, which means that it can work on different platforms such as Windows

, Linux, and macOS.

Many other components are also part of this collection of software and are explained below.

1. **Cross-Platform:** Different local systems have different configurations of operating systems installed in it. The component of cross-platform has been included to increase the utility and audience for this package of Apache distributions. It supports various platforms such as packages of Windows, Linus, and MAC OS.
2. **Apache:** It is an HTTP a cross-platform web server. It is used worldwide for delivering web content. The server application has made free for installation and used for the

community of developers under the aegis of Apache Software Foundation. The remote server of Apache delivers the requested files, images, and other documents to the user.

3. **MariaDB:** Originally, MySQL DBMS was a part of XAMPP, but now it has been replaced by MariaDB. It is one of the most widely used relational DBMS, developed by MySQL. It offers online services of data storage, manipulation, retrieval, arrangement, and deletion.
4. **PHP:** It is the backend scripting language primarily used for web development. PHP allows users to create dynamic websites and applications. It can be installed on every platform and supports a variety of database management systems. It was implemented using C language. PHP stands for **Hypertext Processor**. It is said to be derived from Personal Home Page tools, which explains its simplicity and functionality.
5. **Perl:** It is a combination of two high-level dynamic languages, namely Perl 5 and Perl 6. Perl can be applied for finding solutions for problems based on system administration, web development, and networking. Perl allows its users to program dynamic web applications. It is very flexible and robust.
6. **phpMyAdmin:** It is a tool used for dealing with MariaDB. Its version 4.0.4 is currently being used in XAMPP. Administration of DBMS is its main role.
7. **OpenSSL:** It is the open-source implementation of the Secure Socket Layer Protocol and Transport Layer Protocol. Presently version 0.9.8 is a part of XAMPP.
8. **XAMPP Control Panel:** It is a panel that helps to operate and regulate upon other components of the XAMPP. Version 3.2.1 is the most recent update. A detailed description of the control panel will be done in the next section of the tutorial.
9. **Webalizer:** It is a Web Analytics software solution used for User logs and provide details about the usage.
10. **Mercury:** It is a mail transport system, and its latest version is 4.62. It is a mail server, which helps to manage the mails across the web.
11. **Tomcat:** Version 7.0.42 is currently being used in XAMPP. It is a servlet based on JAVA to provide JAVA functionalities.
12. **Filezilla:** It is a File Transfer Protocol Server, which supports and eases the transfer operations performed on files. Its recently updated version is 0.9.41.

3.5.3 INSTALLATION PROCESS OF XAMPP

As discussed in the earlier part of the tutorial, XAMPP is a cross-platform stack of software that provides web solutions based on technologies like [MariaDB](#)

, Apache Server, [Perl](#)

, and [PHP](#)

. Further, it is supported by many file formats, such as .EXE, .ZIP and .7z- .7zip. Out of the three, the .EXE extension is the easiest to operate upon while installation. In this topic, we will discuss steps to be followed to download and install XAMPP software successfully on your desktops. Since it is a cross-platform software, it is supported by a number of operating systems, including [Windows](#)

, Linux, and MAC OS. The process to be followed for installation of [XAMPP](#) will be explained for all three operating systems:

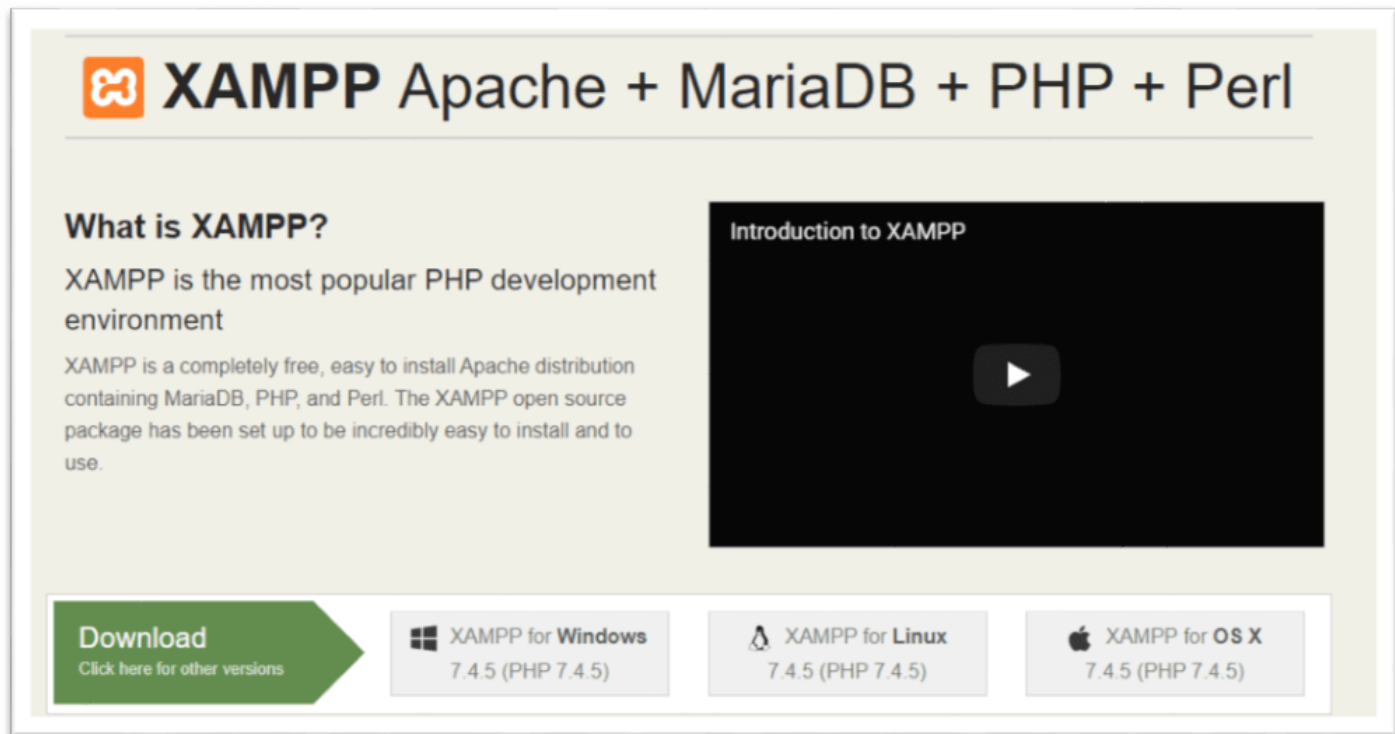
The installation process in Windows

STEP 1- Open any web browser and visit <https://www.apachefriends.org/index.html>

. On the home page, you can find the option to download XAMPP for three platforms- Windows, MAC, and [Linux](#)

. Click on **XAMPP for Windows**. The latest version available on the website is **7.4.5**.

As soon as you click on it, a message displaying the automatic start of download appears on the screen.



STEP 2- After the download is completed, double click the .exe extension file to start the process of installation.

STEP 3- A pop-up screen with the message asking you to allow to make changes on your desktop appears. Click "YES" to continue the process.

STEP 4- Click to **Allow access** or deactivate the firewall and any other antivirus software because it can hamper the process of installation. Thus, it is required to temporarily disable any antivirus software or security firewall till the time all the XAMPP components have been installed completely.

STEP 5- Just before the installation, a pop-up window appears with a warning to **disable UAC**. User Account Control (UAC) interrupts the XAMPP installation because it restricts the access to write to the C: drive. Therefore, it is suggested to disable it for the period of installation.

STEP 6- After clicking the .exe extension file, the XAMPP setup wizard opens spontaneously. Click on "NEXT" to start the configuration of the settings.

STEP 7- After that, a 'Select Components' panel appears, which gives you the liberty to choose amongst the separate components of the XAMPP software stack for the installation. To get a complete localhost server, it is recommended to install using the default options of containing all available components. Click "NEXT" to proceed further.

STEP 8- The setup is now ready to install, and a pop-up window showing the same appears on the screen. Click "NEXT" to take the process forward.

STEP 9- Select the location where the XAMPP software packet needs to be installed. The original setup creates a folder titled XAMPP under C:\ for you. After choosing a location, click "NEXT".

STEP 10- After choosing from all the previously mentioned preferences (like language and learn more bitnami dialogue box) click to start the installation. The setup wizard will unpack and install the components to your system. The components are saved to the assigned directory. This process may takes a few minutes to complete. The progress of the installation in terms of percentage is visible on the screen.

STEP 11- After the successful installation of the XAMPP setup on your desktop, press the "FINISH" button.

On clicking the FINISH button, the software automatically launches, and the CONTROL PANEL is visible. The image below shows the appearance of the final result.

3.5.4 Testing XAMPP Installation

Xampp allows us to work on a local server and test a local copy of websites using PHP code and mysql database. Any local copy can be accessed on the browser using a URL if xampp is active. XAMPP is a cross-platform software so that it can be used on any operating system.

3.5.4 XAMPP Control Panel:

XAMPP Control Panel provides us with complete control over all xampp components. This panel can be used to start and stop various modules. Modules available on xampp control panel are:-

- Apache
- MySQL
- FileZilla
- Mercury
- Tomcat

Testing XAMPP Installation:

In the control panel, click on the start action for the Apache module. This will instruct XAMPP to initiate the Apache webserver. Now open any web browser and type: `http://localhost` or `127.0.0.1`, you should see the XAMPP screen. It means you have XAMPP installed properly on your system.

Follow these steps to check whether the XAMPP server is configured correctly or not, and we can create a PHP test page. This page can be stored under XAMPP's localhost and can run on the browser.

Steps to create a PHP page:-

1. Open the XAMPP directory present in C Drive and choose the htdocs folder (C:\xampp\htdocs for standard installations). This directory contains all the data required to run a web page.
2. Please create a new folder **Test** for the test page in htdocs.
3. Open notepad and type the following code and save the file in the Test folder.
4. Make sure you have saved the file with .php extension.

4. CHAPTER

4.1 USES AND PURPOSE

Customer:

These are the main source of business for the bank. The number of the customer will improve the position in the market. There are different types of customers from the common people to the businessman and everyone has an account on the priority of they require it.

The different types of customers are:

- Individual Account holders
- Joint account holders
- Partnership firm holders
- Limited liability companies
- Clubs and Associations
- Trusts

These are the persons or group which can have the account on the bank for their individual or business or non-profit work.

Account:

Every customer will become a customer when they open an account in the bank start depositing the money or take some other service. The account enables the customer to take advantage of the facilities provided by the bank. Every customer has their unique account number and the bank will identify you by only that account number.

The account number will be the same for all the branches of that particular bank. This will hold the balance in account, interest which is provided by the bank to that customer and if the customer is active this will be defined by the last transaction done by the customer forex. The saving account usually remains active for 6 months with no transaction after that it needs to be reactivated.

Further the types of account a customer can have are:

- Savings Accounts
- Current Account
- Checking Accounts

These are some types of accounts users can have according to their use and priority. All different types of accounts have their own benefit for the customer. Some other things a customer can opt for are a locker or fixed deposits.

Transactions:

Every time an account holder performs some activity on the account it will be updated through transactions this is like logs but only showing the required details. Any time a customer makes any changes in an account like pay or deposit it will be through transactions. This helps in keeping the track of cash flow in the bank.

Also, help in managing the correct information if there is some data loss to the bankside or if there is any query at the customer side.

FEATURES PROPOSED IN THE CURRENT BANK MANAGEMENT SYSTEM

The main objective of the system is to automate all the banking process with improved performance and realize the vision of paperless banking. Salient features of the proposed bank management system is given below.

Using this bank management system any information can be easily searched. User can view all the details of the customer.

Using this system user can create new customer account and maintain its data efficiently and effectively. All records of account and customer are stored in separate files. Which are maintained constantly update by system.

Manage large number of customer details with ease. Particular A/c information can be modified A particular customer record can be modified for one or more field's customer name, address by providing A/c number.

Create a statistical report to facilitate the finance department work. Activities like updating, modification, deletion of records should be easier. A customer record can be easily deleted by authorize user by providing A/c number.

The proposed system provides faster data access, data entry and retrieval.

The proposed system is more efficient, fast, reliable, user friendly. Over and above the proposed system does not have any possibility of data loss during processing.

4.2.OBJECTIVES OF THE PROJECT

- To provide the customer with the facility of fast and computerized banking process.
 - Provide the easy way for opening and closing the banking account.
 - To maintain the database of customer for better management.
 - In this project various kinds of reports are generate like list of all the banking customer account whose balance is less then opening amount for cash withdrawal.i
 - Database of customer history is also maintained for better service to the costumer
 - Various kinds of queries like Interest rate, information about available balance on customer's account.
 - To provide ATM and Banking management system.
 - To provide easily display account on banking and Atm
- IMPORTANCE 7 Importance Of B.M.S.:- It has a great importance in Banks, which has many transactions like account opening, account closing, depositing, withdrawals, etc these can not be managed easily manually. So one system can be required which can easily manage the wide activities of managing the various transactions taking place. We can overcome this problem by using Banking Management system. Advantage Of Using Computer Over Manual System:- When size of a Bank grows, Department, Staff, customers and the transactions etc also increase manually we can manage up to some limit but when the size of work become large problem occurs. So Banking Management System helps in this type of problems. We can easily keep the information related to the banks and easily maintain and update the information. In short, the advantage of using computers over the manual management system can be understood through the following parts:- 1. Time saving 2. Economical 3. Quick 4. Accurate 5. Better control 6. Increase

4.3 Designing concept:

Design Goals:- Here our approach is to develop an interactive system, which accepts inputs, options and provides the user with the required outputs. After some give and take, the design goals suggest that the user have the following features: 1. The system should have good interface. 2. To facilitate the user, a number of messages should be designed which guide the action. 3. Care should be taken to keep the user input simple to ensure accuracy of input. 4. The system should be flexible.

System Design Phase:- • Data gathering • Detail system Input design Output design Control design Database design • Program writing.

Input Design:- Input design specifies how data are accepted for processing. This involves:-
→ What data to input.
→ What medium to use.
→ How the data should be arranged or coded.
→ The dialog to guide users in providing input.
→ Data items and transactions validation to detect error.
→ Method of performing input validation.

Output Design:- For many Ends-users output is the main reason for developing the system and the basis on which they will evaluate the usefulness of the application. Output design involves.
→ Determines what information to present.
→ Decide whether to display or print information.
→ Presentation in an acceptable format.

Control Design:- While entering data or in requesting the performance of certain function there may occur errors, which could be in proper use of system. Control provides ways to:-
→ Ensure that only authorized users access the system.
→ Guarantee that transaction is acceptable.
→ Validate the data for accuracy.
→ Determine whether any necessary data have been omitted.

Database Design:- The collection of data is usually referred to as database. The database contains information about the particulars of organization. The management of data involves the definition of structures of the storage of information and the provision of mechanism for the manipulation of information.

17 INPUT DESIGN In this we decide the following input design details:
→ What data to input
→ What medium to use
→ How the data should be arranged or coded
→ The dialogue to guide customer in providing input
→ Data items and transactions needing validation to detect errors
→ Methods for performing cash withdrawing and steps to follow when errors occur
→ The design of input also includes specifying the means by which customer and system operators direct the system in which actions to take.
→ Inaccurate input data are the most common use of error in data processing, errors entered by the data entry operator can be controlled by the input design is the process of converting user originated inputs to a computer based format.
→ The goals of designing

input data are to make a data entry as easy as possible, logical and free from errors. In entering data operators need to know the following points:-
→ Allocated space for each field
→ Field sequence, which must match in the source documents
→ The format in which data fields are entered .The data may be of different formats but the customer might be required to enter it only one format. 18 OUTPUT DESIGN Output design is the most important and the direct source of information to the customer, efficient, intelligent output design should improve system's relationship with the customer and in decision making a form of output is a hard copy from the printer. Output, refers to the results and information that are generated by the system. Output is the main reason for developing the system and the basis on which they will evaluate the usefulness of the application. When designing output, we have accomplished the following:-
→ Determine what information to present
→ Decide whether to display, print, or "speak" the information and select the output medium
→ Arrange the presentation of information in an acceptable format
→ Decide how to distribute the output to intended recipient. The output design is specified on layout forms, sheets that describe the location characteristics, and format of the column headings and pagin.

4.3.1 BANKING SYSTEM :

Banking Management system. It has a great importance in Banks, which has many transactions like account opening, account closing, depositing, withdrawals, etc these can not be managed easily manually. So one system can be required which can easily manage the wide activities of managing the various transactions taking place. We can overcome this problem by using Banking Management system.

1. CUSTOMERS RECORDS: It holds all the information regarding the customers having their account in the bank. 21
2. ACCOUNT OPENING: The customers wanting to open their account in the bank can do so by creating a new account.
3. TRANSACTIONS: The basic function of bank is to receive deposits from public, withdrawals on the demand of customers, transfer of money from one account to another, etc
4. ACCOUNT CLOSING: When any customer wants to close their account, the bank can do so by deleting their particular account.
5. PRINT REPORT: After a customer's transaction, the bank prints out the report on the

balance of the customer's account .

6. CALCULATE INTEREST: Calculation of interest on the deposited money, the loans etc are done by the bank.

CHAPTER 5

5. RESULTS AND PERFORMANCE ANALYSIS

5.1 TRANSACTIONS DETAILS

The screenshot shows the 'Transactions' page in the 'Online Banking System - Admin' interface. The page has a sidebar with navigation links: Dashboard, Account Management, Transaction (selected), Transactions, Deposit, Withdraw, Transfer, Announcements, Maintenance, and Settings. The main content area displays a table of transactions with the following data:

#	Account #	Name	Amount	Transaction	Date Created
1	6231415	Smith, JohnD	1,000.00	Withdraw	2021-07-14 15:56:12
2	6231415	Smith, JohnD	1,000.00	Deposits	2021-07-14 15:55:54
3	6231415	Smith, JohnD	5,000.00	Deposits	2021-07-14 15:49:15
4	6231415	Smith, JohnD	1,000.00	Transferred to 10140715	2021-07-14 15:35:16
5	10140715	Blake, ClaireC	1,000.00	Transferred from 6231415	2021-07-14 15:35:16
6	6231415	Smith, JohnD	1,000.00	Withdraw	2021-07-14 15:25:20
7	6231415	Smith, JohnD	3,000.00	Deposits	2021-07-14 15:23:21
8	6231415	Smith, JohnD	3,000.00	Transferred to 10140715	2021-07-14 13:51:04
9	10140715	Blake, ClaireC	3,000.00	Transferred from 6231415	2021-07-14 13:51:04
10	10140715	Blake, ClaireC	2,500.00	Withdraw	2021-07-14 13:37:59

Showing 1 to 10 of 13 entries

This is a **Simple Banking System** Project that I'd developed in **PHP**. The project manages the bank's clients' accounts and handles the basic transactions of the clients. The system has 2 types of users which are the **Admin** and **Clients**. The system allows the client to save the records of their **Deposits**, **Withdraws**, and **Fund Transfers**.

Features

FEATURES:

Public

- ☐ Login Page
- ☐ Announcement Page
- ☐ About us Page

Client-Side

- ☐ Dashboard Page (display the account number and current balance)
- ☐ List of Transactions History
- ☐ Deposit
- ☐ Withdraw
- ☐ Fund Transfer

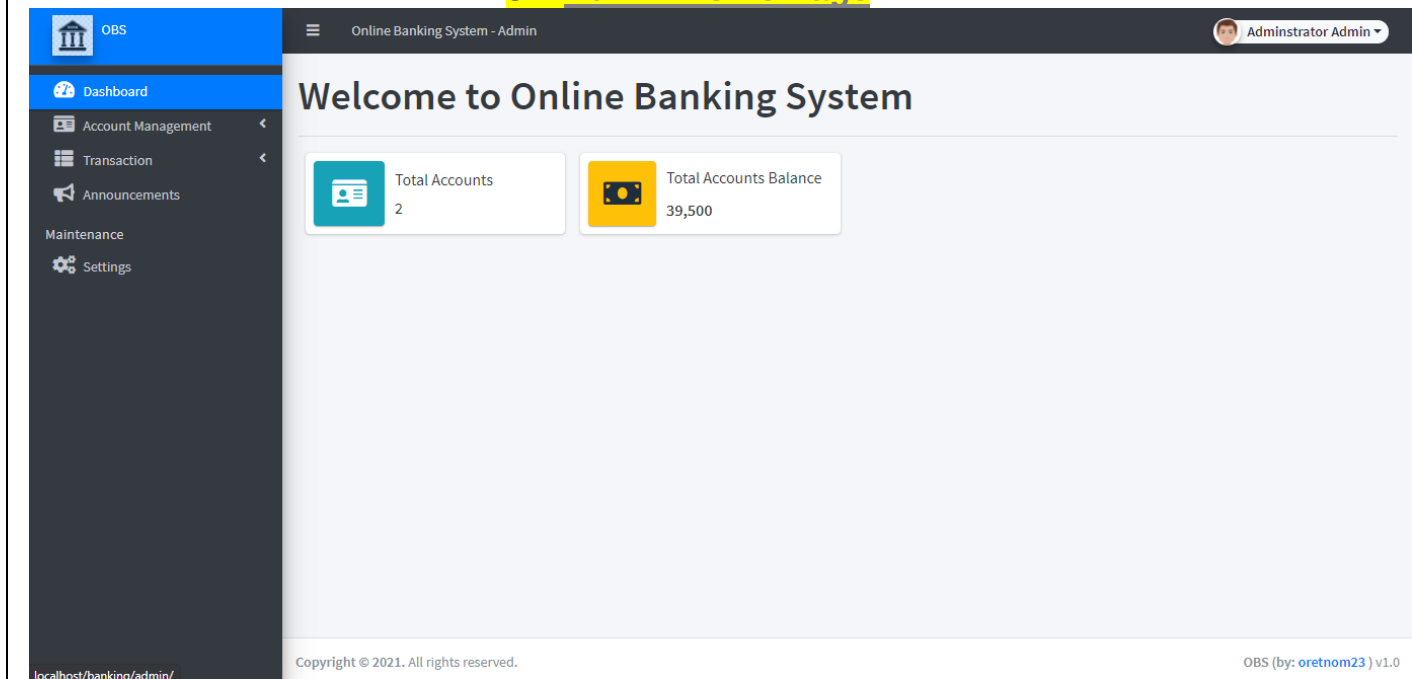
- ☐ Manage System Credentials

Admin Side

- ☐ Dashboard Page
- ☐ List of All Transactions History
- ☐ Deposit for Client Withdraw for Client
- Fund Transfer for Client
- Manage System Credentials
- Manage System Settings/Inf

SNAPSHOTS:

5.2 Admin Home Page



5.3 Transfer Fund Page


OBS

Online Banking System - Admin


Administrator Admin

Dashboard

Account Management

Transaction

Transactions

Deposit

Withdraw

Transfer

Announcements

Maintenance

Settings

Deposit

Account Number

10140715

Transfer To

6231415

Name

Blake, Claire C

Name

Smith, John D

Balance

21500

Deposit Amount

2000

Submit

Cancel

Copyright © 2021. All rights reserved.

OBS (by: oretnom23) v1.0

5.4 Admin Transaction List Page


OBS

Online Banking System - Admin


Administrator Admin

Dashboard

Account Management

Transaction

Transactions

Deposit

Withdraw

Transfer

Announcements

Maintenance

Settings

Transactions

Show 10 entries

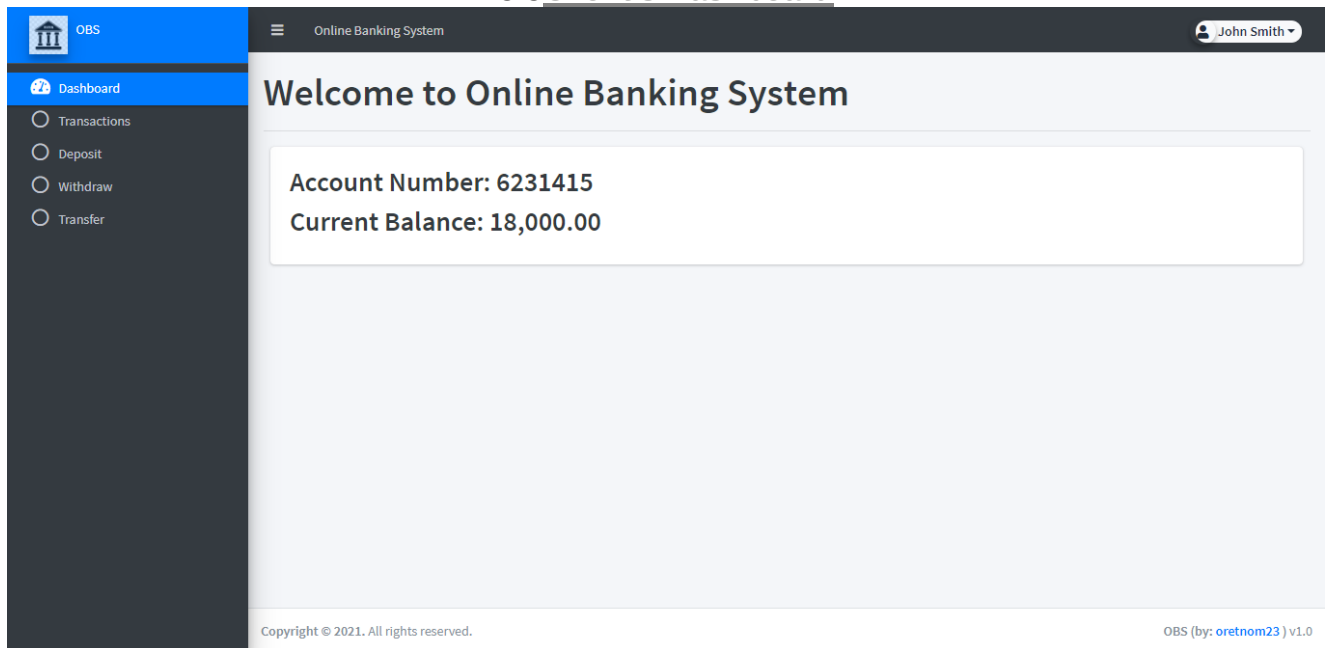
Search:

#	Account #	Name	Amount	Transaction	Date Created
1	6231415	Smith, JohnD	1,000.00	Withdraw	2021-07-14 15:56:12
2	6231415	Smith, JohnD	1,000.00	Deposits	2021-07-14 15:55:54
3	6231415	Smith, JohnD	5,000.00	Deposits	2021-07-14 15:49:15
4	6231415	Smith, JohnD	1,000.00	Transferred to 10140715	2021-07-14 15:35:16
5	10140715	Blake, ClaireC	1,000.00	Transferred from 6231415	2021-07-14 15:35:16
6	6231415	Smith, JohnD	1,000.00	Withdraw	2021-07-14 15:25:20
7	6231415	Smith, JohnD	3,000.00	Deposits	2021-07-14 15:23:21
8	6231415	Smith, JohnD	3,000.00	Transferred to 10140715	2021-07-14 13:51:04
9	10140715	Blake, ClaireC	3,000.00	Transferred from 6231415	2021-07-14 13:51:04
10	10140715	Blake, ClaireC	2,500.00	Withdraw	2021-07-14 13:37:59

Showing 1 to 10 of 13 entries

Previous
1
2
Next

5.5 Client's Dashboard



5.6 Client's Transaction List

OBS

Dashboard

Transactions

Deposit

Withdraw

Transfer

Online Banking System

John Smith

Transactions

Show 10 entries

Search:

#	Date Created	Transaction	Amount
1	2021-07-14 15:56:12	Withdraw	1,000.00
2	2021-07-14 15:55:54	Deposits	1,000.00
3	2021-07-14 15:49:15	Deposits	5,000.00
4	2021-07-14 15:35:16	Transferred to 10140715	1,000.00
5	2021-07-14 15:25:20	Withdraw	1,000.00
6	2021-07-14 15:23:21	Deposits	3,000.00
7	2021-07-14 13:51:04	Transferred to 10140715	3,000.00
8	2021-07-14 13:32:03	Deposits	5,000.00
9	2021-07-14 12:03:58	Beginning balance	10,000.00

Showing 1 to 9 of 9 entries

Previous

1

Next

Copyright © 2021. All rights reserved.

OBS (by: oretnom23) v1.0

CHAPTER 6

6. CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION:

Bank management system is a virtualization of transactions in banking system. **The banking system are used manual working but when we used online banking system it is totally virtualization process which avoid manual process and converts it in automatic process .**

REFERENCES

- [1] From all the information we can create a bank management system. The whole process of banking is showed in the code like creating new account, displaying details, deposits, withdraw. Hence we can conclude that Project is accomplished successfully.
- [2] REFERENCES:
- [3] 1. Budd, T. (1997b), An Introduction to Object-Oriented Programming, 2nd edn, Addison-Wesley.
- [4] 2. Ghezzi, C., Jayazeri, M. & Mandrioli, D. (1998), Fundamentals of Software Engineering, 2nd edn, Prentice-Hall. 3. K. Appel and W. Haken. (1976), Every Planar Map is 4-colorable, Bull. Amer. Math. Soc., vol. 82, pp. 711-712

APPENDIX

A. SOURCE CODE

MACHINE TRAINING CODE:

```
-- phpMyAdmin SQL Dump
-- version 5.1.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Jul 14, 2021 at 10:03 AM
-- Server version: 10.4.19-MariaDB
-- PHP Version: 8.0.7

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `banking_db`
--
```

--

-- Table structure for table `accounts`

--

```
CREATE TABLE `accounts` (  
  `id` int(30) NOT NULL,  
  `account_number` varchar(50) NOT NULL,  
  `pin` text NOT NULL,  
  `firstname` varchar(250) NOT NULL,  
  `lastname` varchar(250) NOT NULL,  
  `middlename` varchar(250) NOT NULL,  
  `email` text NOT NULL,  
  `password` text NOT NULL,  
  `generated_password` text NOT NULL,  
  `balance` float NOT NULL,  
  `date_created` datetime NOT NULL DEFAULT current_timestamp(),  
  `date_updated` datetime DEFAULT NULL ON UPDATE current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

--

-- Dumping data for table `accounts`

--

```
INSERT INTO `accounts` (`id`, `account_number`, `pin`, `firstname`, `lastname`,  
  `middlename`, `email`, `password`, `generated_password`, `balance`,  
  `date_created`, `date_updated`) VALUES  
(1, '6231415', '12352', 'hanok', 'kumar', 's', 'hanokkumar@gmail.com', 'arvi1807', '',  
  18000, '2021-07-14 12:03:58', '2021-07-14 15:56:12'),  
(2, '10140715', '14369', 'arvind', 'kumar', 's', 'arvind@gmail.com', '', 'arvindk1', 21500,  
  '2021-07-14 12:05:19', '2021-07-14 15:35:16');
```

-- -----

--

-- Table structure for table `announcements`

--

```
CREATE TABLE `announcements` (  
  `id` int(30) NOT NULL,  
  `title` text NOT NULL,  
  `announcement` text NOT NULL,  
  `date_created` datetime NOT NULL DEFAULT current_timestamp(),  
  `date_updated` datetime DEFAULT NULL ON UPDATE current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

--

-- Dumping data for table `announcements`

--

```
INSERT INTO `announcements` (`id`, `title`, `announcement`, `date_created`,  
`date_updated`) VALUES  
(1, 'new announcement 1', '&lt;p&gt;This is a sample Announcement only.&lt;/p&gt;',  
'2021-07-14 14:10:09', '2021-07-14 14:11:30'),  
(3, 'Sample', '&lt;hr style=&quot;margin: 0px; padding: 0px; clear: both; border-top:  
0px; height: 1px; background-image: linear-gradient(to right, rgba(0, 0, 0, 0), rgba(0,  
0, 0, 0.75), rgba(0, 0, 0, 0)); color: rgb(0, 0, 0); font-family: &quot;Open  
Sans&quot;, Arial, sans-serif; font-size: 14px; text-align: center;&quot;&gt;&lt;/div  
id=&quot;Content&quot; style=&quot;margin: 0px; padding: 0px; position: relative;  
color: rgb(0, 0, 0); font-family: &quot;Open Sans&quot;, Arial, sans-serif;  
font-size: 14px; text-align: center;&quot;&gt;&lt;/div id=&quot;bannerL&quot;  
style=&quot;margin: 0px 0px 0px -160px; padding: 0px; position: sticky; top: 20px;  
width: 160px; height: 10px; float: left; text-align: right;&quot;&gt;&lt;/div&gt;&lt;/div  
id=&quot;bannerR&quot; style=&quot;margin: 0px -160px 0px 0px; padding: 0px;  
position: sticky; top: 20px; width: 160px; height: 10px; float: right; text-align:  
left;&quot;&gt;&lt;/div&gt;&lt;/div class=&quot;boxed&quot; style=&quot;margin: 10px
```

28.7969px; padding: 0px; clear: both;><div id="lipsum" style="margin: 0px; padding: 0px; text-align: justify;><p style="margin-right: 0px; margin-bottom: 15px; margin-left: 0px; padding: 0px;><Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam et tempus leo. Nulla id sagittis libero. Cras est nisi, consequat ut vulputate eget, mattis non ante. Phasellus a lorem at nunc venenatis commodo. Donec vitae cursus augue. Donec eleifend molestie laoreet. Praesent dictum arcu ac congue molestie. Etiam nisl risus, blandit vel scelerisque eu, ultricies at mauris. Suspendisse non bibendum magna. Vestibulum porttitor enim a elit feugiat bibendum eu malesuada eros. Etiam eu est at neque dictum efficitur. Integer fermentum porttitor scelerisque.</p><p style="margin-right: 0px; margin-bottom: 15px; margin-left: 0px; padding: 0px;><Nunc a leo rutrum, congue ex sit amet, laoreet tortor. Nunc at bibendum sapien. Cras libero nunc, varius quis ultricies non, finibus in quam. Aenean quis justo vitae purus ultrices luctus. Curabitur viverra non lacus vehicula malesuada. Phasellus convallis mattis libero eget accumsan. Ut sollicitudin mattis enim, bibendum eleifend felis euismod et. Etiam ut libero purus. Aenean vel sceleris</p></div></div></div>','2021-07-14 14:33:41', NULL);

-- -----

--

-- Table structure for table `system\_info`

--

```
CREATE TABLE `system_info` (
  `id` int(30) NOT NULL,
  `meta_field` text NOT NULL,
  `meta_value` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

--

-- Dumping data for table `system\_info`

--

```

INSERT INTO `system_info` (`id`, `meta_field`, `meta_value`) VALUES
(1, 'name', 'Online Banking System'),
(6, 'short_name', 'OBS'),
(11, 'logo', 'uploads/1626243720_bank.jpg'),
(13, 'user_avatar', 'uploads/user_avatar.jpg'),
(14, 'cover', 'uploads/1626249540_dark-bg.jpg');

```

```

-- -----

```

```

--

```

```

-- Table structure for table `transactions`

```

```

--

```

```

CREATE TABLE `transactions` (
  `id` int(30) NOT NULL,
  `account_id` int(30) NOT NULL,
  `type` tinyint(4) NOT NULL COMMENT '1=Cash in, 2= Withdraw, 3=transfer',
  `amount` float NOT NULL,
  `remarks` text DEFAULT NULL,
  `date_created` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

```

--

```

```

-- Dumping data for table `transactions`

```

```

--

```

```

INSERT INTO `transactions` (`id`, `account_id`, `type`, `amount`, `remarks`,
`date_created`) VALUES
(1, 1, 1, 10000, 'Beginning balance', '2022-07-14 12:03:58'),
(2, 2, 1, 20000, 'Beginning balance', '2022-07-14 12:05:19'),
(3, 1, 1, 5000, 'Deposits', '2021-07-14 13:32:03'),
(6, 2, 1, 2500, 'Withdraw', '2021-07-14 13:37:59'),
(7, 1, 3, 3000, 'Transferred to 10140715', '2021-07-14 13:51:04'),
(8, 2, 3, 3000, 'Transferred from 6231415', '2021-07-14 13:51:04'),

```

```

(10, 1, 1, 3000, 'Deposits', '2021-07-14 15:23:21'),
(11, 1, 1, 1000, 'Withdraw', '2021-07-14 15:25:20'),
(12, 1, 3, 1000, 'Transferred to 10140715', '2021-07-14 15:35:16'),
(13, 2, 3, 1000, 'Transferred from 6231415', '2021-07-14 15:35:16'),
(14, 1, 1, 5000, 'Deposits', '2021-07-14 15:49:15'),
(15, 1, 1, 1000, 'Deposits', '2021-07-14 15:55:54'),
(16, 1, 1, 1000, 'Withdraw', '2021-07-14 15:56:12');

-----

--

-- Table structure for table `users`
--

CREATE TABLE `users` (
  `id` int(50) NOT NULL,
  `firstname` varchar(250) NOT NULL,
  `lastname` varchar(250) NOT NULL,
  `username` text NOT NULL,
  `password` text NOT NULL,
  `avatar` text DEFAULT NULL,
  `last_login` datetime DEFAULT NULL,
  `type` tinyint(1) NOT NULL DEFAULT 0,
  `date_added` datetime NOT NULL DEFAULT current_timestamp(),
  `date_updated` datetime DEFAULT NULL ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--

-- Dumping data for table `users`
--

INSERT INTO `users` (`id`, `firstname`, `lastname`, `username`, `password`,
`avatar`, `last_login`, `type`, `date_added`, `date_updated`) VALUES
(1, 'Administrator', 'Admin', 'admin', '0192023a7bbd73250516f069df18b500',

```

```
'uploads/1624240500_avatar.png', NULL, 1, '2021-01-20 14:02:37', '2021-06-21  
09:55:07');
```

```
--
```

```
-- Indexes for dumped tables
```

```
--
```

```
--
```

```
-- Indexes for table `accounts`
```

```
--
```

```
ALTER TABLE `accounts`  
  ADD PRIMARY KEY (`id`);
```

```
--
```

```
-- Indexes for table `announcements`
```

```
--
```

```
ALTER TABLE `announcements`  
  ADD PRIMARY KEY (`id`);
```

```
--
```

```
-- Indexes for table `system_info`
```

```
--
```

```
ALTER TABLE `system_info`  
  ADD PRIMARY KEY (`id`);
```

```
--
```

```
-- Indexes for table `transactions`
```

```
--
```

```
ALTER TABLE `transactions`  
  ADD PRIMARY KEY (`id`);
```

```
--
```

```
-- Indexes for table `users`
```

```
--
```

```

ALTER TABLE `users`
  ADD PRIMARY KEY (`id`);

--

-- AUTO_INCREMENT for dumped tables
--

--

-- AUTO_INCREMENT for table `accounts`
--
ALTER TABLE `accounts`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--

-- AUTO_INCREMENT for table `announcements`
--
ALTER TABLE `announcements`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--

-- AUTO_INCREMENT for table `system_info`
--
ALTER TABLE `system_info`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=15;

--

-- AUTO_INCREMENT for table `transactions`
--
ALTER TABLE `transactions`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=17;

--

-- AUTO_INCREMENT for table `users`
--

```

```
ALTER TABLE `users`
```

```
  MODIFY `id` int(50) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;  
COMMIT;
```

```
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

```
/*!40101 SET
```

```
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
```

```
| /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```