## Project Documentation: Banking System Frontend using PHP

## Table of Contents

1. [Project Overview](#project-overview)
2. [Database Schema](#database-schema)
3. [Forms and Scripts for Data Insertion](#forms-and-scripts-for-data-insertion)
   * Customer
   * Account
   * Transaction
   * Loan
   * LoanRepayment
   * Branch
   * Employee
   * CustomerAccount
   * AccountType
   * TransactionType
   * AuditLog
   * CustomerFeedback
4. [Setup and Installation](#setup-and-installation)
5. [Usage](#usage)
6. [Contributing](#contributing)
7. [License](#license)

## Project Overview

This project aims to develop a frontend interface for a banking system using PHP. The system enables users to insert data into a MySQL database through various forms. The database schema includes tables for managing customers, accounts, transactions, loans, repayments, branches, employees, and other banking operations.

## Database Schema

The database schema is designed to handle all the necessary data for a banking system. The primary tables include:

* **Customer**: Stores information about customers, such as name, date of birth, address, phone number, email, identification number, KYC status, and customer since date.
* **Account**: Contains details about customer accounts, including account number, account type, balance, date opened, customer ID, branch ID, and status.
* **Transaction**: Records transaction details like transaction date, amount, transaction type, account ID, description, and related account ID.
* **Loan**: Manages loan information including loan number, loan type, amount, interest rate, start and end dates, customer ID, and status.
* **LoanRepayment**: Tracks loan repayments with fields for repayment date, amount, loan ID, and payment method.
* **Branch**: Holds branch information such as branch name, address, phone number, and manager ID.
* **Employee**: Stores employee data including name, position, branch ID, contact number, email, and supervisor ID.
* **CustomerAccount**: Maps the relationship between customers and their accounts.
* **AccountType**: Defines the types of accounts available.
* **TransactionType**: Specifies the types of transactions.
* **AuditLog**: Keeps a log of significant actions and events.
* **CustomerFeedback**: Collects feedback from customers.

## Forms and Scripts for Data Insertion

Each table in the database is associated with a form and a PHP script to handle data insertion. Here's an overview of the process for each table:

### Customer

#### Form

The customer form collects data such as name, date of birth, address, phone number, email, identification number, KYC status, and customer since date.

#### PHP Script

The PHP script (insert\_customer.php) processes the form data and inserts it into the Customer table in the database. It connects to the database, prepares an SQL statement with placeholders for the data, binds the actual values to the placeholders, and executes the statement to insert the data.

php

Copy code

<?php

// insert\_customer.php

// Connect to your MySQL database (replace with your credentials)

$servername = "localhost";

$username = "username";

$password = "password";

$dbname = "your\_database";

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Prepare and bind parameters

$stmt = $conn->prepare("INSERT INTO Customer (Name, DateOfBirth, Address, PhoneNumber, Email, IdentificationNumber, KYCStatus, CustomerSince) VALUES (?, ?, ?, ?, ?, ?, ?, ?)");

$stmt->bind\_param("ssssssss", $name, $date\_of\_birth, $address, $phone\_number, $email, $identification\_number, $kyc\_status, $customer\_since);

// Set parameters and execute

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

$date\_of\_birth = $\_POST['date\_of\_birth'];

$address = $\_POST['address'];

$phone\_number = $\_POST['phone\_number'];

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

$identification\_number = $\_POST['identification\_number'];

$kyc\_status = $\_POST['kyc\_status'];

$customer\_since = $\_POST['customer\_since'];

$stmt->execute();

echo "New customer added successfully";

$stmt->close();

$conn->close();

?>

### Account

#### Form

The account form collects data such as account number, account type ID, balance, date opened, customer ID, branch ID, and status.

#### PHP Script

The PHP script (insert\_account.php) processes the form data and inserts it into the Account table in the database. It follows a similar process to the Customer script: connecting to the database, preparing an SQL statement, binding values, and executing the statement.

php

Copy code

<?php

// insert\_account.php

// Connect to your MySQL database (replace with your credentials)

$servername = "localhost";

$username = "username";

$password = "password";

$dbname = "your\_database";

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// Prepare and bind parameters

$stmt = $conn->prepare("INSERT INTO Account (AccountNumber, AccountTypeID, Balance, DateOpened, CustomerID, BranchID, Status) VALUES (?, ?, ?, ?, ?, ?, ?)");

$stmt->bind\_param("siissis", $account\_number, $account\_type\_id, $balance, $date\_opened, $customer\_id, $branch\_id, $status);

// Set parameters and execute

$account\_number = $\_POST['account\_number'];

$account\_type\_id = $\_POST['account\_type\_id'];

$balance = $\_POST['balance'];

$date\_opened = $\_POST['date\_opened'];

$customer\_id = $\_POST['customer\_id'];

$branch\_id = $\_POST['branch\_id'];

$status = $\_POST['status'];

$stmt->execute();

echo "New account added successfully";

$stmt->close();

$conn->close();

?>

### Transaction

#### Form

The transaction form gathers details such as transaction date, amount, transaction type ID, account ID, description, and related account ID.

#### PHP Script

The PHP script (insert\_transaction.php) handles the data insertion into the Transaction table, following a similar pattern to the previous scripts.

### Loan

#### Form

The loan form collects information such as loan number, loan type, amount, interest rate, start and end dates, customer ID, and status.

#### PHP Script

The PHP script (insert\_loan.php) processes the loan data and inserts it into the Loan table.

### LoanRepayment

#### Form

The loan repayment form collects repayment date, amount, loan ID, and payment method.

#### PHP Script

The PHP script (insert\_loan\_repayment.php) handles the insertion of loan repayment data into the LoanRepayment table.

### Branch

#### Form

The branch form collects data such as branch name, address, phone number, and manager ID.

#### PHP Script

The PHP script (insert\_branch.php) processes the form data and inserts it into the Branch table.

### Employee

#### Form

The employee form gathers details such as name, position, branch ID, contact number, email, and supervisor ID.

#### PHP Script

The PHP script (insert\_employee.php) handles the insertion of employee data into the Employee table.

### CustomerAccount

#### Form

The customer account form collects customer ID, account ID, and relationship type.

#### PHP Script

The PHP script (insert\_customer\_account.php) processes the form data and inserts it into the CustomerAccount table.

### AccountType

#### Form

The account type form collects account type name and description.

#### PHP Script

The PHP script (insert\_account\_type.php) handles the insertion of account type data into the AccountType table.

### TransactionType

#### Form

The transaction type form gathers transaction type name and description.

#### PHP Script

The PHP script (insert\_transaction\_type.php) processes the form data and inserts it into the TransactionType table.

### AuditLog

#### Form

The audit log form collects table name, action type, action time, user ID, and previous data.

#### PHP Script

The PHP script (insert\_audit\_log.php) handles the insertion of audit log data into the AuditLog table.

### CustomerFeedback

#### Form

The customer feedback form collects customer ID, feedback date, feedback type, and comments.

#### PHP Script

The PHP script (insert\_customer\_feedback.php) processes the form data and inserts it into the CustomerFeedback table.

## Setup and Installation

1. **Install XAMPP**: Download and install XAMPP from the official website.
2. **Start Apache and MySQL**: Open the XAMPP control panel and start the Apache and MySQL services.
3. **Create a Database**: Open phpMyAdmin (accessible via the XAMPP control panel) and create a new database for the project.
4. **Create Tables**: Use the provided SQL schema to create the necessary tables in your database.
5. **Place PHP Files**: Save the PHP form and script files in the htdocs directory of your XAMPP installation.

## Usage

1. **Open a Browser**: Open a web browser and navigate to http://localhost/form\_name.php (replace form\_name with the name of the form you want to use, e.g., customer\_form.php).
2. **Fill Out Forms**: Fill out the forms with the required information and submit.
3. **Data Insertion**: The data will be inserted into the corresponding table in the database.

## Contributing

Contributions to this project are welcome. Please submit a pull request or open an issue for any improvements or bug fixes.

## License

This project is licensed under the MIT License.

4o