Final Project

Mouhamed Thioune

CIS 344

```sql
CREATE DATABASE banks_portal;
USE banks_portal;
CREATE TABLE accounts
(
accountId INT NOT NULL AUTO_INCREMENT,
ownername VARCHAR(45) NOT NULL,
owner_snn INT NOT NULL,
balance DECIMAL(10,2) DEFAULT 0.00,
account_status VARCHAR(45),
PRIMARY KEY(accountId)
);
```

First step:

- The first task is to create a database named "banks_portal."
- Using the Database with the command USE
- The next step is to create a table named "accounts" within the "banks_portal" database with the following attributes: accountId, ownerName, owner_ssn, balance, and account_status.
- The accountId attribute is set as the primary key, and the balance attribute has a default value of 0.00.

```sql
CREATE TABLE IF NOT EXISTS transactions
(
transactionId INT NOT NULL AUTO_INCREMENT,
accountId INT NOT NULL,
transactiontype VARCHAR(45) NOT NULL,
transactionamount DECIMAL(10,2) NOT NULL,
PRIMARY KEY(transactionId)
);
```

Creating table Transactions

- create a table named "transactions" within the "banks_portal" database.
- With the following attributes: transactionId, accountId, transactionType, and transactionAmount.
- The transactionId attribute is set as the primary key.
- The IF NOT EXISTS clause ensures that the table is created only if it doesn't already exist.

Populating the tables accounts and transactions

- With the command INSERT INTO by adding the given values

```
INSERT INTO accounts
(ownerName, owner_ssn, balance, account_status)
VALUES (1, "Maria Josef", 123456789, 10000.00, "active"),
(2, "Linda Jones", 987654327, 2600.00, "inactive"),
(3, "John McGrail", 222222222, 100.50, "active"),
(4, "Patty Luna", 111111111, 509.75, "inactive");
```

```
INSERT INTO transactions
(accountId, transactionType, transactionAmount)
VALUES (1, "deposit", 650.98),
(3, "withdraw", 899.87),
(3, "deposit", 350.00);
```

Select * From accounts;

- To show the table with the values

```
SELECT * FROM accounts;
```

accountid	ownername	owner_snn	balance	account_stat...
1	Maria Josef	123456789	10000.00	active
2	Linda Jones	987654327	2600.00	inactive
3	John McGrail	222222222	100.50	active
4	Patty Luna	111111111	509.75	inactive
NULL	NULL	NULL	NULL	NULL

```
DELIMITER //
CREATE PROCEDURE accountTransactions(IN accountID INT)
BEGIN
    SELECT * FROM Transactions
    WHERE accountID = accountID;
END //
DELIMITER ;
```

Creating the "accountTransactions" Stored Procedure

- The procedure takes an accountID parameter and selects the corresponding transactions using a SELECT statement.

```
DELIMITER //
CREATE PROCEDURE deposit(
IN accountID INT,IN amount DECIMAL(10,2))
BEGIN
  INSERT INTO transactions (accountId, transactionType, transactionAmount)
  VALUES (accountID, "deposit", amount);

  UPDATE accounts SET balance = balance + amount WHERE accountId = accountID;
END//
DELIMITER ;

DELIMITER //
```

Creating the ”deposit" Stored Procedure

- In here, the code defines a stored procedure that accepts an accountID and amount input parameters. It uses an INSERT INTO statement to add a new row to the "transactions" table with the provided values. Then, it uses an UPDATE statement to increase the account's balance in the "accounts" table by the specified amount.

```
DELIMITER //
CREATE PROCEDURE withdraw(IN accountID INT,IN amount DECIMAL(10,2))
BEGIN
  INSERT INTO transactions (accountId, transactionType, transactionAmount)
  VALUES (accountID, "withdraw", amount);

  UPDATE accounts SET balance = balance - amount WHERE accountId = accountID;
END//
DELIMITER ;
```

Creating the "withdraw" Stored Procedure

- The procedure takes accountID and amount parameters, inserts a new row into the "transactions" table, and updates the corresponding account's balance in the "accounts" table.

Import portalDatabase.py in Python and modify the default parameter to connect it to MySQL server

Opening portalDatabase.py in Python

portalDatabase.py - /Users/ahmedthioune/Desktop/CIS 344/Final Project/port..

```
import mysql.connector
from mysql.connector import Error


class Database():
    def __init__(self,
                 host="localhost",
                 port="3306",
                 database="banks_portal",
                 user='root',
                 password='Amigost2'):

        self.host       = host
        self.port       = port
        self.database   = database
        self.user       = user
        self.password   = password
        self.connection = None
        self.cursor     = None
        self.connect()

    def connect(self):
        try:
            self.connection = mysql.connector.connect(
                host         = self.host,
                port         = self.port,
                database     = self.database,
                user         = self.user,
                password     = self.password)

            if self.connection.is_connected():
                return
        except Error as e:
            print("Error while connecting to MySQL", e)


    def getAllAccounts(self):
        if self.connection.is_connected():
            self.cursor= self.connection.cursor();
            query = "select * from accounts"
```

Running portalServer.py in Python

portalServer.py - /Users/ahmedthioune/Desktop/CIS 344/portalServer.py (3.11....

```
from http.server import HTTPServer, BaseHTTPRequestHandler
from os import curdir, sep
from portalDatabase import Database
import cgi

class PortalRequestHandler(BaseHTTPRequestHandler):

    def __init__(self, *args):
        self.database = Database()
        BaseHTTPRequestHandler.__init__(self, *args)

    def do_POST(self):

        try:
            if self.path == '/addAccount':
                self.send_response(200)
                self.send_header('Content-type','text/html')
                self.end_headers()
                form = cgi.FieldStorage(
                    fp=self.rfile,
                    headers=self.headers,
                    environ={'REQUEST_METHOD': 'POST'}
                )

                owner_name     = form.getvalue("oname")
                owner_ssn = int(form.getvalue("owner_ssn"))
                balance    = float(form.getvalue("balance"))
                acct_status = "active"
                ##Call the Database Method to a add a new student
                '''
                    Example call: self.database.addAccount(student_name, ownerNa
                '''

                print("grabbed values",owner_name,owner_ssn,balance)

                self.wfile.write(b"<html><head><title> Bank's Portal </title></h
```

Localhost:8000/

Open localhost:8000/ to make sure this opens

Bank's Portal

Home| Add Account| Withdraw| Deposit | Search Transactions| Delete Account

All Accounts

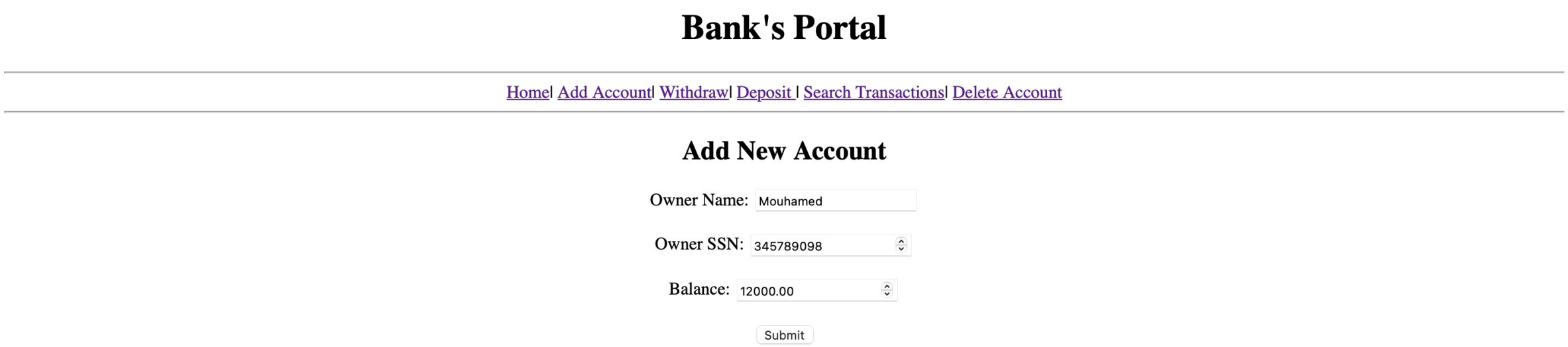
Account ID	Account Owner	Balance	Status
1	Maria Josef	10000.00	active
2	Linda Jones	2600.00	inactive
3	John McGrail	100.50	active
4	Patty Luna	509.75	inactive

```
def addAccount(self, ownerName, owner_ssn, balance, status):
    ''' Complete the method to insert an
                account to the accounts table'''
    cursor = self.db_connection.cursor()
    query = "INSERT INTO accounts (owner_name, owner_ssn, balance, status) VALUES (%s, %s, %s, %s)"
    values = (ownerName, owner_ssn, balance, status)
    cursor.execute(query, values)
     cursor.close()
    pass
```

addAccount

- **The code for addAccount in the source code portalDatabase.py in Python**

The results


Bank's Portal
Home| Add Account| Withdraw| Deposit | Search Transactions| Delete Account
Add New Account
Owner Name: Mouhamed
Owner SSN: 345789098
Balance: 12000.00
Submit

Link of my repository:

https://github.com/ahmedthioune/Mouhamed