|  |  |
| --- | --- |
| Roll No.: B055 | Name: Arpit Kubadia |
| Class: B.Tech CS B | Module: 4 |
| Project: 1 | Date: |
| Grade: | Batch: B3 |

**Code:**

import sqlite3 as sql

import os

from datetime import datetime

os.system('clear')

con = sql.connect('bank\_data.db')

cur = con.cursor()

def signup\_func():

try:

account\_type = input("\nInput Account Type (s: saving account c: current account) : ")

balance = int(input("Amount initiated : "))

if (account\_type is 'c' and balance < 5000) or (balance < 0):

print("Invalid Amount")

exit()

print("Enter your details:")

name = input("\nFull Name : ")

email = input("\nEmail: ")

telephone = input("\nPhone no:")

date\_of\_birth=input("\nDOB(dd/mm/yyyy):")

address = input("\nAddress: ")

password = input("\nPlease enter your desired password.")

while (len(password)<8):

password = input("Pass should have more than 8 characters. Please retry.")

cur.execute("INSERT INTO customer(name,email,telephone,date\_of\_birth,address,password) VALUES(?,?,?,?,?,?);",(name,email,telephone,date\_of\_birth,address,password))

con.commit()

account\_status = "open"

account\_lock = 1

cur.execute("INSERT INTO accounts(balance,account\_type,account\_status,account\_lock) VALUES(?,?,?,?);",(balance,account\_type,account\_status,account\_lock))

con.commit()

value=cur.execute("SELECT cust\_id FROM customer order by cust\_id desc limit 1;")

for row in value:

account\_no = row[0]

print("\nYour Account number : ",account\_no)

val='NULL'

cur.execute('INSERT INTO logTable(cust\_id,account\_no,deposit,withdraw,transfer,balance) VALUES(?,?,?,?,?,?);',(account\_no,account\_no,val,val,val,balance))

con.commit()

except:

print("There is some error please try again\n\n")

def signin\_func():

i=0

while (i<3):

try:

account\_no = int(input("Enter your Account Number : "))

password = input("Enter Password : ")

value = cur.execute("SELECT password FROM customer WHERE cust\_id = (?);",(account\_no,))

for row in value:

passwd=row[0]

if (passwd == password):

print("success")

profile\_func()

exit()

else:

print('Wrong credentials. Try again')

i+=1

except Exception as e:

print(e)

import traceback

traceback.print\_exc()

print("You have made 3 failed attemps. The account is locked")

cur.execute("UPDATE accounts SET account\_lock= WHERE account\_no=(?);",(account\_no,))

con.commit()

def admin\_signin\_func():

user\_id = input("Enter Administrator user name.")

password = input("Enter Administrator password.")

if user\_id == 'admin' and password == 'adminpass':

print("Admin Login successful")

exit()

else:

print("Wrong credentials.Try Again")

cust\_id = account\_no=1

def change\_addr():

address = input("Enter address : ")

cur.execute('UPDATE customer SET address = (?) WHERE cust\_id = (?);',(address, cust\_id))

print("Address successfully changed\n")

con.commit()

def deposit():

amount = int(input("Enter Amount to be deposited : "))

value = cur.execute('SELECT balance FROM accounts WHERE account\_no = (?);', (account\_no,))

for row in value:

bal=row[0]

bal = bal + amount

print("New balance is : ",bal)

cur.execute('UPDATE accounts SET balance = (?) WHERE account\_no = (?);', (bal, account\_no))

con.commit()

cur.execute('INSERT INTO logTable(cust\_id,account\_no,deposit,withdraw,transfer,balance) VALUES (?,?,?,?,?,?);', (account\_no,account\_no,amount,'NULL','NULL', bal))

con.commit()

value = cur.execute('SELECT transaction\_id FROM logTable ORDER BY transaction\_id DESC limit 1;')

for row in value:

print("Your transaction ID is : ", row[0])

def withdraw():

amount = int(input("Enter Amount to be withdrawl : "))

value = cur.execute('SELECT balance FROM accounts WHERE account\_no = (?);', (account\_no,))

for row in value:

bal = row[0]

if (bal - amount < 0):

print('Not enough balance')

exit()

else:

bal = bal-amount

print("New balance is : ",bal)

cur.execute('UPDATE accounts SET balance = (?) WHERE account\_no = (?);', (bal, account\_no))

cur.execute('INSERT INTO logTable(cust\_id,account\_no,deposit,withdraw,transfer,balance) VALUES (?,?,?,?,?,?);',(account\_no, account\_no,"NULL", amount,"NULL", bal))

con.commit()

value = cur.execute('SELECT transaction\_id FROM logTable ORDER BY transaction\_id DESC limit 1;')

for row in value:

print("Your transaction ID is : ", row[0])

def get\_statement():

value = cur.execute('SELECT account\_no, transaction\_id, deposit, withdraw, transfer, balance FROM logTable WHERE account\_no = (?);',(account\_no,))

for row in value:

print("Account\_no",row[0])

print("Transaction\_id",row[1])

print("Deposit", row[2])

print("Withdraw", row[3])

print("Transfer", row[4])

print("Balance", row[5])

def transfer\_money():

acc2 = int(input('Enter Account No. to transfer money : '))

a = cur.execute('SELECT account\_no FROM accounts WHERE account\_no = (?);', (acc2,))

value = cur.execute('SELECT balance FROM accounts WHERE account\_no = (?);', (acc2,))

for row in value:

bal = row[0]

print(bal)

if(a):

amount = int(input('Enter amount to Transfer : '))

value = cur.execute('SELECT balance FROM accounts WHERE account\_no = (?);', (account\_no,))

for row in value:

my = row[0]

if ((my - amount) < 0):

print('Not enough balance')

else:

my = my - amount

bal = bal + amount

cur.execute('UPDATE accounts SET balance = (?) WHERE account\_no = (?);', (my, account\_no))

cur.execute('INSERT INTO logTable(cust\_id,account\_no,deposit,withdraw,transfer,balance) VALUES (?,?,?,?,?,?);',(account\_no,account\_no,"NULL","NULL",amount,my))

cur.execute('UPDATE accounts SET balance = (?) WHERE account\_no = (?);', (bal, acc2))

print("Transfer Complete")

con.commit()

value = cur.execute('SELECT transaction\_id FROM logTable ORDER BY transaction\_id DESC limit:1;')

for row in value:

print("Your trabsaction ID is : ", row[0])

else:

exit()

def close\_account():

cur.execute('UPDATE accounts SET account\_status=(?) WHERE account\_no = (?);', ("closed", account\_no,))

con.commit()

print("Account closed")

def logout():

print("logging out")

exit()

def profile\_func():

print("\n1. Address Change\n2. Money Deposit\n3. Money Withdrawal\n4. Transfer Money\n5. Print Statement\n6. Closure\n7. Customer Logout")

choice\_dict={1:change\_addr,2:deposit,3:withdraw,4:transfer\_money,5:get\_statement,6:close\_account,7:logout}Account

choice = int(input("Enter your choice : "))

choice\_dict[choice]()

if \_\_name\_\_ == '\_\_main\_\_':

while(True):

try:

print("\n1. Sign Up\n2. Sign In\n3. Admin Sign In\n4. Quit\n")

choice = int(input("Enter your choice :"))

choice\_dict = {

1: signup\_func,

2: signin\_func,

3: admin\_signin\_func,

4: exit

}

choice\_dict[choice]()

except KeyError:

print("Try Again")

**Output:**

1. Sign Up

2. Sign In

3. Admin Sign In

4. Quit

Enter your choice :2

Enter your Account Number : 7001

Enter Password : yashyash

success

1. Address Change

2. Money Deposit

3. Money Withdrawal

4. Transfer Money

5. Print Statement

6. Account Closure

7. Customer Logout

Enter your choice : 2

Enter Amount to be deposited : 10000

New balance is : 15000

Your transaction ID is : 3

Enter your Account Number : 7001

Enter Password : yashyash

success

1. Address Change

2. Money Deposit

3. Money Withdrawal

4. Transfer Money

5. Print Statement

6. Account Closure

7. Customer Logout

Enter your choice : 5

Account\_no 1

Transaction\_id 3

Deposit 10000

Withdraw NULL

Transfer NULL

Balance 15000

Enter your Account Number : 6

Enter Password : u

Wrong credentials. Try again

Enter your Account Number : 7001

Enter Password : yashyash

success

1. Address Change

2. Money Deposit

3. Money Withdrawal

4. Transfer Money

5. Print Statement

6. Account Closure

7. Customer Logout

Enter your choice : 6

Account closed

Enter your Account Number : 

**Test Cases:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case id** | **Test cases** | **Priority** | **Preconditions** | **Input test data** | **Steps to be executed** | **Expected results** | **Actual results** | **Pass/fail** |
| 1 | Test if New user is added in database | A | Must know his/ her details  Database connection must be established | All the inputs asked in Form | 1)Fill the Form | User added  New account number should be generated and given to user | User added  New account number is generated and given to user | Pass |
| 2 | Deposit money | A | Must know account no. and password | Input should be a integer value and same amount of money should be deposited to bank | Sign-in and choose withdraw | New amount deposited should be added to previous account balance | New amount deposited is added to previous account balance | Pass |
| 3 | Withdraw Money | A | Must know account no. and password | Input should be a integer value and same amount of money should be Withdrawn to bank | Sign-in and choose withdraw | New amount should be Subtracted from previous account balance | New amount is Subtracted from previous account balance | Pass |