**OVERVIEW**

Bank Management System project is written in Python. The project file contains a python script (main.py) and a database file.This is a simple console based system which is very easy to understand and use. Talking about the system, it contains all the basic functions which include account details, view account record, withdraws and deposit amount, balance inquiry, closing an account and edit account details. In this mini project, there is no such login system. This means he/she can use all those available features easily without any restriction. It is too easy to use,he/she can check the total bank account records easily .Talking about the features of the Bank Management System, a user can create an account by providing the name of the account holder, number, selecting amount type (Saving account or Current account) and providing an initial amount. Then the user can also deposit and withdraw money just by providing his/her account and entering the amount. For certain purpose, he/she can also check for the balance inquiry which displays the account number and amount. He/she can also view all the account holder's list. Another feature is that he/she can modify their account detail and type if they want to.

**OBJECTIVE**

The main objective of this system is to provide a secure system. Our system is password protected and it only allows authorized user to access various functions availlable in the system.Our system will help the user to locate any A/C wanted by the user. It will reduce manual work as most of the work done by computer. As all the manual work will be done automatically so it will increase work speed and reduce time consumption to complete any bank related work. It will also increase the work efficiency as few employee can handle more customers. This will give information.The project banking system has been made to automate the banking System. Through this bank management system user can manage all bank account activity like deposit money, withdraw money, transfer money from one account to another account,online payment etc. Using this bank management system user can check his account detail online like balance in account, bank statement etc. The Administrator can check bank account with a login can work out with A/C holders of the bank can withdraw/deposit cash/ cheque /DD to/from their accounts. This system is also help bank user to create New account easily. The Project makes a sincere effort to provide all the below-mentioned features to meet the requirements of the bank.

**PYTHON**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

**SQL**

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.SQL is the standard language for Relational Database System.It allows users to access data in the relational database management systems, allows users to describe the data, allows users to define the data in a database and manipulate that data, allows to embed within other languages using SQL modules, libraries & pre-compilers, allows users to create and drop databases and tables, allows users to create view, stored procedure, functions in a database, allows users to set permissions on tables, procedures and views.

**LIBRARY AND MODULES USED**

* **Datetime-**

The [datetime](https://docs.python.org/3/library/datetime.html#module-datetime) module supplies classes for manipulating dates and times.While date and time arithmetic is supported, the focus of the implementation is on efficient attribute extraction for output formatting and manipulation.Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

* **Mysql.connector-**

MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant.

Used in converting parameter values back and forth between Python and MySQL data types, for example Python datetime and MySQL DATETIME. You can turn automatic conversion on for convenience, or off for optimal performance. It includes support for protocol compression, which enables compressing the data stream between the client and server.

**SOURCE**

**CODE**

#+================================+

#| WELCOME TO |

#| /\ | \ | | |

#| /\_\_\ | \ | | |

#| / \ | \ | | |

#| BANK |

#+================================+

**# \*\*\*\*\*\*\* \*\*\*\*\*\*\* \*\*\*\*\*\*\* importing libraries \*\*\*\*\*\*\* \*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\***

from datetime import datetime

import mysql.connector

#\*\*\*\***\*\*\*\*\*\*\*** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ADMIN\*\*\*\***\*\*\*\*\*\*\*** \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

aid = 0

ap = "123"

def sep():

print()

print("===========================================")

print()

def inval():

print()

print("+================================+")

print("| INVALID INPUT |")

print("+================================+")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print()

m()

mydb = mysql.connector.connect(

host="localhost",

user="root",

passwd="admin@123",

database="p")

mycursor = mydb.cursor()

#\*\*\*\*\*\*\*\***\*\*\***\*\*\*\*\*\*\*\*\*\*\*\*\*create customer table\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

tabcustomer='''CREATE TABLE IF NOT EXISTS customers(

acno int AUTO\_INCREMENT,

n varchar(20) NOT NULL,

ln varchar(20) NOT NULL,

adhar varchar(20) NOT NULL,

dob varchar(10) NOT NULL,

city varchar(20) NOT NULL,

area varchar(20) NOT NULL,

pincode varchar(12) NOT NULL,

pno varchar(12) NOT NULL,

email varchar(30) NOT NULL,

actyp varchar(10) NOT NULL,

sms varchar(2),

balance float NOT NULL,

PRIMARY KEY(acno));'''

mycursor.execute(tabcustomer)

#\*\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\***\*\*\*\*create transactions table\*\*\*\*\*\***\*\*\*\*\*\*\***\*\*\*\*\*\*\*\*\*\*\*\*

tabtrans = '''CREATE TABLE IF NOT EXISTS transactions(

tid int AUTO\_INCREMENT,

acno int,

amount float,

type varchar(10),

date varchar(20),

PRIMARY KEY(tid),

FOREIGN KEY (acno) REFERENCES customers(acno));'''

mycursor.execute(tabtrans)

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\*\*** \*\*\*create auth table\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\*\*** \*\*\*\*\*

tabpwd = '''CREATE TABLE IF NOT EXISTS auth(

acno int,

password varchar(100) NOT NULL,

FOREIGN KEY (acno) REFERENCES customers(acno));'''

mycursor.execute(tabpwd)

mydb.commit()

#\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\*\*** \*\*\*\*\*check admin/customer\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\***

def check( level):

print(" WELCOME TO LOG IN PANEL ")

uid = int(input("Enter your Log in id : "))

pword = input("Enter your password : ")

if level == 'admin':

if uid == aid and pword == ap:

return True

elif level == 'customer':

p = Dtail("auth",("acno", uid, 'int'), "password")

if pword == p:

return True

sep()

return False

#\*\*\*\*\*\*\***\*\*\*\***\*\*\*function to handle operation available for admin\*\*\*\*\*\*\*\*\*\*\*

def mmad():

auth = check( "admin")

if auth == True:

while True:

print(" WELCOME DEAR ADMIN ")

print("+------------------------------------+")

print("| ADMIN MENU |")

print("+------------------------------------+")

print("| 1. Open New Account |")

print("+------------------------------------+")

print("| 2. Close existing Account |")

print("+------------------------------------+")

print("| 3. See all Customers details |")

print("+------------------------------------+")

print("| 4. See all Transactions details |")

print("+------------------------------------+")

print("| 5. Log out |")

print("+------------------------------------+")

print()

print("Enter your choice ")

choice = int(input('> '))

if choice == 1:

sep()

adac(mydb)

elif choice == 2:

sep()

closac()

elif choice == 3:

sep()

rtable("customers")

elif choice == 4:

sep()

rtable("transactions")

elif choice == 5:

sep()

m()

else:

inval()

#\*\*\***\*\*\*\*\*\*\*** \*\*function to handle operation available for customer\*\***\*\*\*\*\*\***\*\*\*\*

def mmc():

auth = check( "customer")

if auth == True:

while True:

print(" WELCOME DEAR CUSTOMER ")

print()

print("+--------------------------------+")

print("| CUSTOMER MENU |")

print("+--------------------------------+")

print("| 1. Transaction Menu |")

print("+--------------------------------+")

print("| 2. Log Out |")

print("+--------------------------------+")

print()

print("Enter your choice ")

choice = int(input('> '))

if choice == 1:

sep()

transmenu()

elif choice == 2:

sep()

m()

else:

inval()

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*function to add/create new account\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

def adac(mydb):

n = input("Enter your first name : ")

ln = input("Enter your last name : ")

adhar = int(input("Enter your Aadhar number : "))

dob = input("Enter your date of birth (Format DD/MM/YYYY): ")

city = input("Enter your city name :")

area = input("Enter your area name :")

pcod = input("Enter your pin code : ")

pno = input("Enter your phone number : ")

email = input("Enter your email-id : ")

actyp = input("Enter your account type [Current/Saving] : ")

sms = input("Do you want to activate SMS Banking service ? [Y/N] ")

balance = float(input("Enter opening amount : "))

password = input("Enter your password : ")

if len(adhar)!=16 or len(pcod) !=6 or len(pno) !=10:

inval()

else:

print()

new= '''INSERT INTO customers (n,ln,adhar,dob,city,area,

pincode,pno,email,actyp,sms,balance) VALUES (''' + \

"'"+n+"'" + "," + "'"+ln+"'" + "," + "'"+str(adhar)+"'" + "," + \

"'"+dob+"'" + "," + "'"+city+"'" + "," + "'"+area+"'" + "," + "'"+pcod+ \

"'" + "," + "'"+pno+"'" + "," + "'"+email+"'" + "," + "'"+actyp+"'" + \

"," + "'"+sms+"'" + "," + str(balance) + ");"

mycursor.execute(new)

acno = Dtail("customers",("adhar", adhar, 'str'), "acno")

n = "INSERT INTO auth (acno, password) VALUES (" + str(acno) + \

"," + "'" + str(password) + "'" + ");"

mycursor.execute(n)

mydb.commit()

print()

print("Account created successfully...")

sep()

#\*\*\*\*\*\*\*\*\*\***\*\*\*\***\*\*\*\*\*\*\*\*function to close account\*\*\***\*\*\*\*\***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

def closac():

print("Ennter account number to be deleted ")

acno = int(input("> "))

del1 = "DELETE FROM transactions WHERE acno = " + \

str(acno) + ';'

del2 = "DELETE FROM auth WHERE acno = " + \

str(acno) + ';'

del3 = "DELETE FROM customers WHERE acno = " + \

str(acno) + ';'

mycursor.execute(del1)

mycursor.execute(del2)

mycursor.execute(del3)

mydb.commit()

print()

print("Record deleted successfully...")

print()

sep()

#\*\*\*\*\*\*\*\*\*\*\*\*\***\*\*\*\*\*\*\***\*\*\*\*\*detail from given table\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

def Dtail(tname, cond, detail):

if cond[2] == 'str':

query = 'SELECT ' + detail + ' FROM ' + tname + ' WHERE ' + \

str(cond[0]) + '=' + "'" + str(cond[1]) + "'" + ';'

elif cond[2] == 'int':

query = 'SELECT ' + detail + ' FROM ' + tname + \

' WHERE ' + str(cond[0]) + '=' + str(cond[1]) + ';'

cursor = mydb.cursor()

cursor.execute(query)

data = cursor.fetchall()

p = data[0][0]

return p

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*function to print table\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

def rtable(tname):

cursor = mydb.cursor()

read = "SELECT \* FROM " + tname

cursor.execute(read)

d = cursor.fetchall()

t = [description[0] for description in cursor.description]

print(t)

for i in d:

print(i)

sep()

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*function to credit money\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

def deposit(acno):

amount = float(input("Enter amount to deposit : "))

today = datetime.now()

dpst = "UPDATE customers SET balance = balance + " + \

str(amount) + " WHERE acno = " + str(acno) + ";"

isnertt = "INSERT INTO transactions(acno,amount,type,date) VALUES ( " + \

str(acno) + "," + str(amount) + "," + "'" + "Credited" + "'" + ",""'" + \

str(today)[:19] + "'" + ");"

mycursor.execute(dpst)

mycursor.execute(isnertt)

mydb.commit()

print()

print("Amount deposited successfully")

sep()

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*function to deposit money\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

def withdraw(acno):

a = float(input("Enter amount to withdraw : "))

camnt = Dtail( "customers",("acno", acno, 'int'), "balance")

if a > camnt:

print("\nYou don't have sufficient balance!\n")

else:

today = datetime.now()

wdraw = "UPDATE customers SET balance = balance - " + \

str(a) + " WHERE acno = " + str(acno) + ";"

isnertt = "INSERT INTO transactions(acno,amount,type,date) VALUES ( " +\

str(acno) + "," + str(a) + "," + "'" + "Debited" + "'" + "," + "'" + \

str(today)[:19] + "'" + ");"

mycursor.execute(wdraw)

mycursor.execute(isnertt)

mydb.commit()

print()

print("Amount Withdrawn successfully...")

sep()

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*function for transaction menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

def transmenu():

acno = int(input("Enter your account number : "))

while True:

print("+--------------------------------+")

print("| Transaction Menu |")

print("+--------------------------------+")

print("| 1. Deposit |")

print("+--------------------------------+")

print("| 2. WithDraw |")

print("+--------------------------------+")

print("| 3. Balance Enquiry |")

print("+--------------------------------+")

print("| 4. Back to Main Menu |")

print("+--------------------------------+")

print()

print("Enter your choice ")

choice = int(input('> '))

if choice == 1:

sep()

deposit(acno)

elif choice == 2:

sep()

withdraw(acno)

elif choice == 3:

print("Your current balance is : ")

print("₹", Dtail("customers", ("acno", acno, 'int'), "balance"))

elif choice == 4:

m()

else:

inval()

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*function for log in menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

def m():

print()

print("+========================================+")

print("| LOGIN PANEL |")

print("+========================================+")

print("| 1. Admin Login |")

print("+========================================+")

print("| 2. Customer Login |")

print("+========================================+")

print()

print("Enter your choice ")

choice = int(input('> '))

if choice == 1:

sep()

mmad()

elif choice == 2:

sep()

mmc()

else:

inval()

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*function function for front page\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

def f():

print()

print("+===========================+")

print("| |")

print("| WELCOME TO |")

print("| |")

print("| /\ | \ | | |")

print("| /\_\_\ | \ | | |")

print("| / \ | \ | | |")

print("| |")

print("| BANK |")

print("| |")

print("+========================== +")

print()

print(" PRESS ENTER")

i=input()

sep()

m()

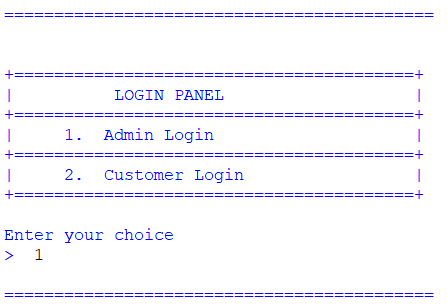
f()

**OUTPUT**

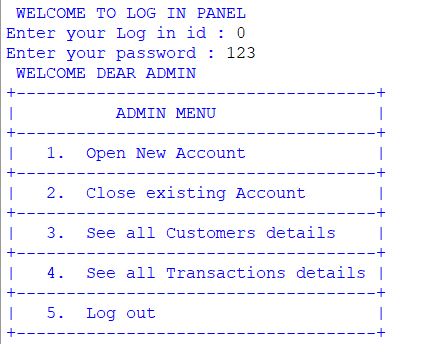
**Welcome Page**



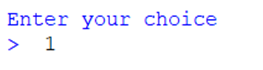
Login Panel

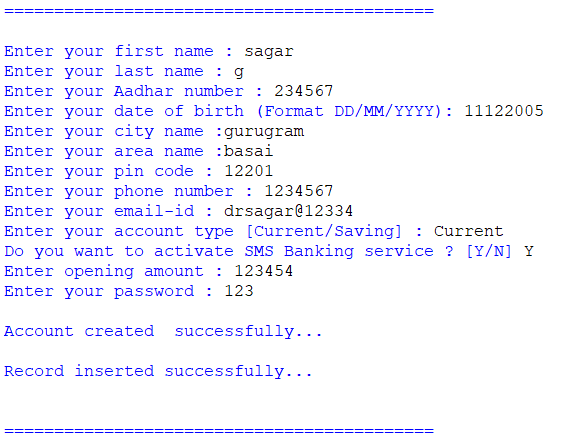


**Admin Login panel**

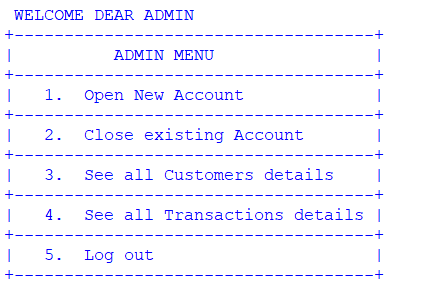


**Creating Account**

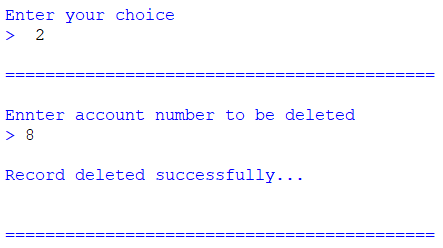
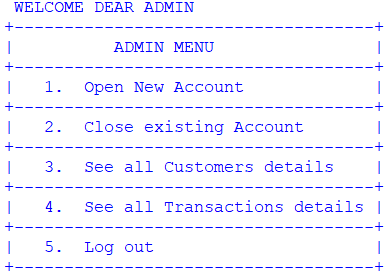




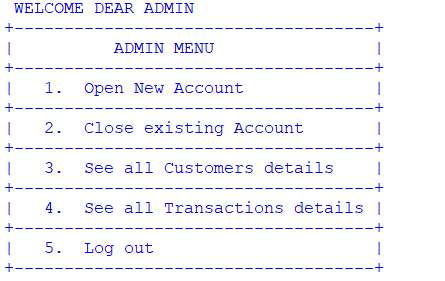
**See All Customers**

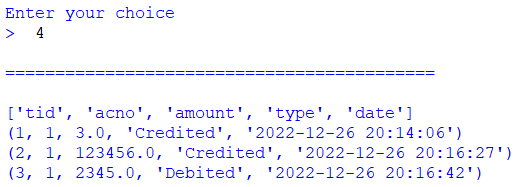
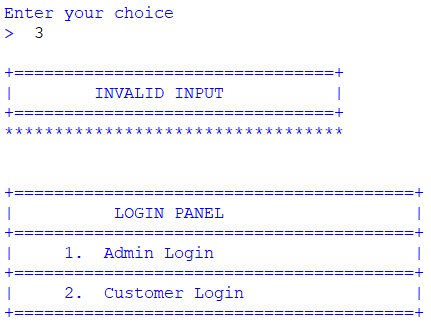
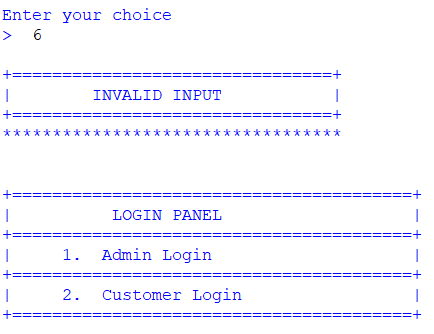
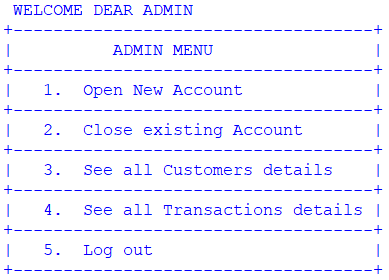




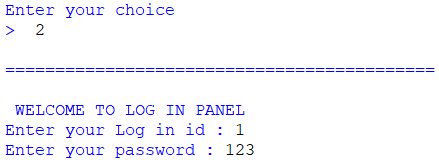


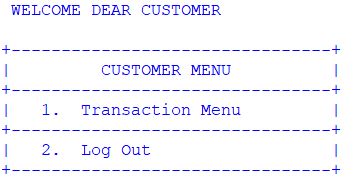
**See Transaction Details**



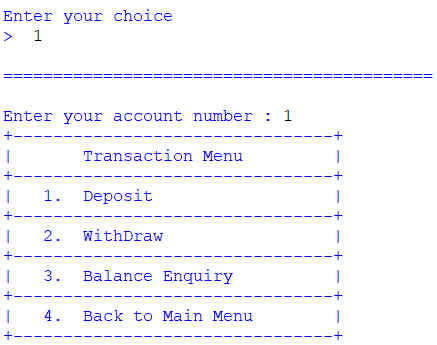
 

**Customer Login**

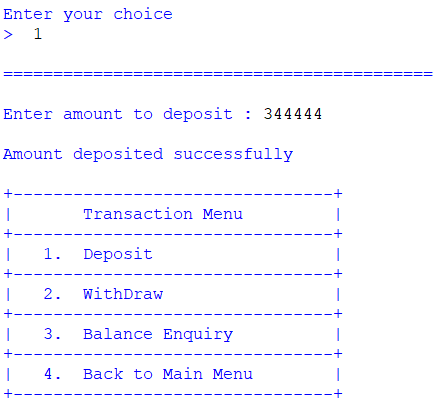




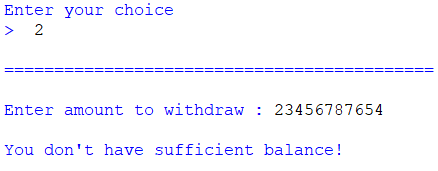
**Transaction Menu**

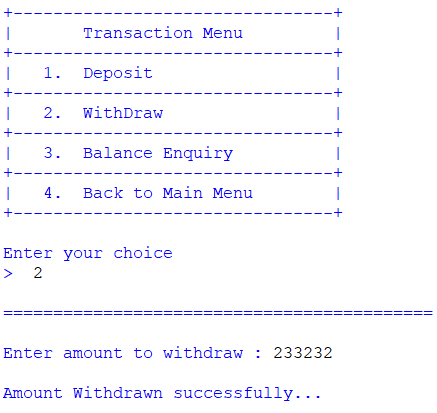


**Deposit Money**

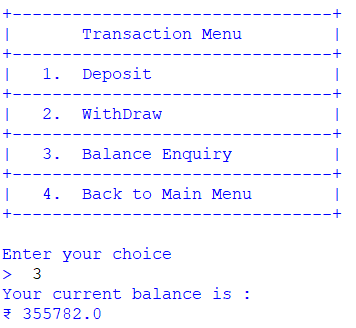


**Withdraw Money**



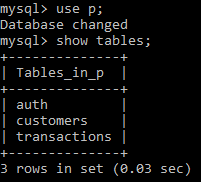


**Balance Enquiry**

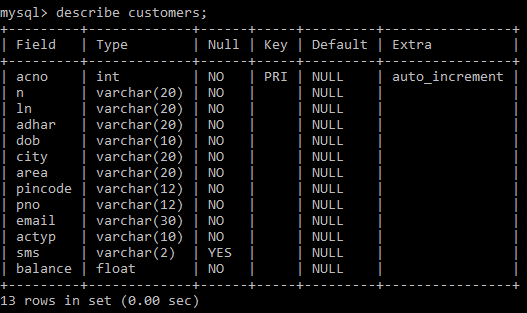
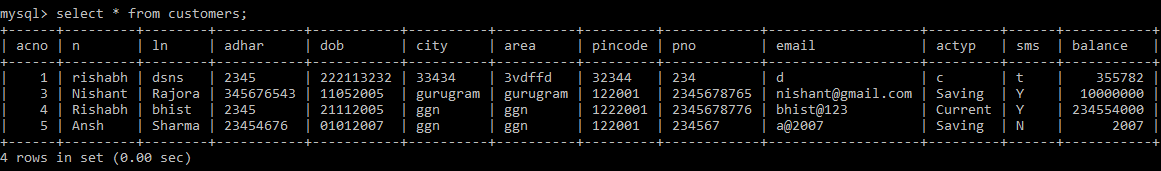


**SQL Database**

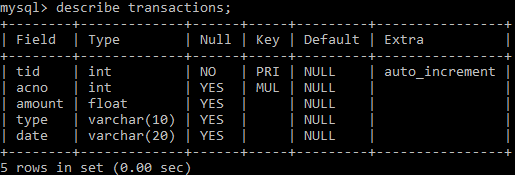
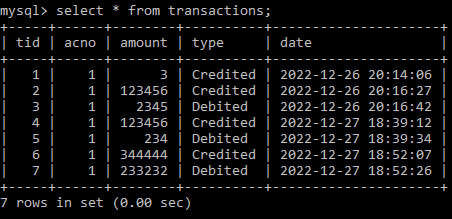
Tables used



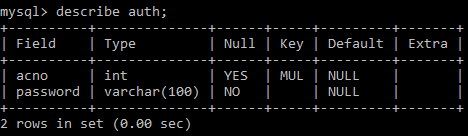
Customers Table

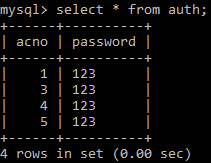
 

Transaction Table

Password Table





**SOFTWARE AND HARDWARE REQUIREMENTS**

* **Hardware requirements-**
* Operating system : Windows 10 Pro
* RAM : 4GB
* Hard Disk : 250GB
* Processor : Intel(R) Core(TM) i5-4300U CPU @ 1.90GHz 2.50GHz
* **Software requirements-**
* Windows OS
* Python

**BIBLIOGRAPHY**

* Website :

<http://www.wikipedia.org>

[www.youtube.com](http://www.youtube.com)

<https://pythonworld.in>

* Computer Science with python

By PREETI ARORA