A Project Report On

BANK MANAGEMENT SYSTEM

Submitted By-

Name: PRATUL SINGH RAGHAVA

Class – XII, Section – B

Khaitan Public School, Sahibabad

Session – 2020-2021

Under the guidance of – Mr. S. S. Das
(PGT – Computer Science)
Department of Computer Science
Khaitan Public School
Sahibabad

CERTIFICATE

This is to certify that **Pratul Singh Raghava** of class XII-B has prepared the report on the project entitled "**Bank Management System**". The report is result of his efforts and endeavours. The report is found worthy of acceptance as final project report for the subject Computer Science (083) of class XII. He has prepared the report under my guidance.

(S. S. Das)
PGT (Computer Science)
Department of Computer Science
Khaitan Public School
Sahibabad

Project On “Banking System”

DBMS: MySQL

Host : localhost

User: root

Pass: root

DataBase: Bank

Table Structure: (Images Below)

```
mysql> USE BANK;
Database changed
mysql> SHOW TABLES;
+----------------+
| Tables_in_bank |
+----------------+
| account        |
| amt            |
+----------------+
2 rows in set (0.00 sec)

mysql> DESC ACCOUNT;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| Accno    | int          | NO   | PRI | NULL    |       |
| Name     | varchar(25)  | NO   |     | NULL    |       |
| Age      | int          | NO   |     | NULL    |       |
| occu     | varchar(15)  | NO   |     | NULL    |       |
| Address  | varchar(50)  | NO   |     | NULL    |       |
| mob      | bigint       | YES  |     | NULL    |       |
| aadharno | bigint       | YES  |     | NULL    |       |
| amt      | double(20,5) | YES  |     | NULL    |       |
| AccType  | varchar(15)  | NO   |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
9 rows in set (0.04 sec)

mysql> DESC AMT;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| Accno      | int          | YES  |     | NULL    |       |
| Amtdeposit | double(20,5) | NO   |     | NULL    |       |
| month      | varchar(15)  | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

SQL Commands to create both the tables:

Account table:

```
mysql> CREATE TABLE ACCOUNT(
Accno INT(15) NOT NULL PRIMARY KEY,
Name VARCHAR(25) NOT NULL,
Age INT(5) NOT NULL,
Occu VARCHAR(15) NOT NULL,
Address VARCHAR(50) NOT NULL,
Mob BIGINT NOT NULL,
Aadharno BIGINT,
amt DOUBLE(20,5),
AccType VARCHAR(15) NOT NULL );
```

Amt table:

```
mysql> CREATE TABLE amt(
Accno INT(15),
Amtdeposit DOUBLE(20,5),
month VARCHAR(15)  NOT NULL );
```

Python Code:

```
import os, platform, mysql.connector, pandas as pd

mydb=mysql.connector.connect(host="localhost",user="root", password="root",
database="Bank")

mycursor=mydb.cursor()


def AccInsert():

   L1={'Accno':int(input("Enter the Account number : ")),

   'name':input("Enter the Customer Name: "),

   'age':int(input("Enter Age of Customer : ")),

   'occu':str(input("Enter Occupation of Customer :")),

   'Address':input("Enter the Address of the Customer : "),

   'Mob':int(input("Enter the Mobile number : ")),

   'Aadharno':int(input("Enter the Aadhar number : ")),

   'Amt':float(input("Enter the Money Deposited : ")),

   'AccType':input("Enter the Account Type (Saving/RD/PPF/Current) : ")}


   sql="Insert into ACCOUNT(Accno ,Name,Age,occu,Address,Mob,Aadharno,amt,AccType)
   values(%s,%s,%s, %s,%s,%s, %s,%s,%s)"

    mycursor.execute(sql,tuple(L1.values()))

    sql="Insert into Amt(Accno, amtdeposit, month) values(%s,%s,%s)"

    mycursor.execute(sql,(L1['Accno'],L1['Amt'],str(input("Enter month of Deposit : "))))

    mydb.commit()


def AccView():


    print("Select the search criteria :\n1. Acc no\n2. Name\n3. Mobile\n4. Aadhar\n5.  View
    All")

    ch=int(input("Enter the choice : "))
```

```
    if ch==1:
        sql="select * from account where Accno=%s"
        mycursor.execute(sql,(int(input("Enter ACC no : ")),))
    elif ch==2:
        sql="select * from account where Name=%s"
        mycursor.execute(sql,(str(input("Enter Name : ")),))
    elif ch==3:
        sql="select * from account where Mob=%s"
        mycursor.execute(sql,(int(input("Enter Mobile No : ")),))
    elif ch==4:
        sql="select * from account where Aadharno=%s"
        mycursor.execute(sql,(str(input("Enter Aadhar : ")),))
    elif ch==5:
        sql="select * from account"
        mycursor.execute(sql)
    res=mycursor.fetchall()
    pd.set_option('display.max_columns', None)
    print("The Customer details are as follows :",
          pd.DataFrame(res,columns=['AcNo','Name','Age','Occn','Add','Mob','Aadh','Amt','AccTy'
          ]))


def AccDeposit():
    L2={'Accno':int(input("Enter the Account number : ")),'Amtdeposit':int(input("Enter the
    Amount to be deposited : ")),'month':str(input("Enter month of Deposit : "))}
    sql="Insert into amt(Accno,Amtdeposit,Month) values(%s,%s,%s)"
    mycursor.execute(sql,tuple(L2.values()))
    mydb.commit()
```

```
def accView():

    print("Please enter the details to view the Money details :")

    Accno=int(input("Enter the Account number of the Customer whose amount is to be
    viewed : "))

    sql="Select Account.Accno,Account.Name,Account.Age,Account.occu,Account.Address,
    Account.Mob,Account.Aadharno,Account.Amt,Account.AccType, sum(amt.Amtdeposit),
    amt.month from Account INNER JOIN amt ON Account.Accno=amt.Accno where
    Account.Accno=%s"

    mycursor.execute(sql,(Accno,))

    res=mycursor.fetchall()

    pd.set_option('display.max_columns', None)

    print("The Customer details are as follows : ",
          pd.DataFrame(res,columns=['AccNo','Name','Age','Occn','Add','Mob','Aadh','Amt',
          'AccTy','Tot_Deposit','Dep_Month']))


def closeAcc():

    Accno=int(input("Enter the Account number of the Customer to be closed : "))

    sql="Delete from amt where Accno=%s"

    mycursor.execute(sql,(Accno,))

    sql="Delete from Account where Accno=%s"

    mycursor.execute(sql,(Accno,))

    mydb.commit()


def MenuSet(): #Function For The Student Management System


    print(""" \n\n\nWelcome to the Banking System \n

          This system has the facility of maintaing Banking Records of Customers.

          This system facilitates Selective and Complete Addition/Deletion of Customer
          Banking Records\n\n""")

    print("Enter 1 : To Add Customer\nEnter 2 : To View Customer\nEnter 3 : To Deposit
          Money\nEnter 4 : To Close Account\nEnter 5 : To View All Money Details of a
          Customer")
```

```
    try:
        userInput = int(input("Please Select An Above Option:")) #Will Take Input From User
    except ValueError:
        exit("\nHey! That's Not A Number") #Error Message
    else:
        print("\n")
        a={1:'AccInsert()',2:'AccView()',3:'AccDeposit()',4:'closeAcc()',5:'accView()'}
        if(userInput in a):
            eval(a[userInput])
        else:
            print("Enter correct choice. . . ")

MenuSet()

def runAgain():
    runAgn = input("\nwant To Run Again y/n: ")
    while(runAgn.lower() == 'y'):
        if(platform.system() == "Windows"):
            print(os.system('cls'))
        else:
            print(os.system('clear'))
        MenuSet()
        runAgn = input("\nwant To Run Again y/n: ")

runAgain()
```

Table of contents

PYTHON Overview

Object-Oriented Programming

PYTHON fully supports object-oriented programming, including the four pillars of object-oriented development:

1. Encapsulation
2. Data hiding
3. Inheritance
4. Polymorphism
5. Standard Libraries

Standard PYTHON consists of three important parts:

The core language giving all the building blocks including variables, data types and literals, etc.

The PYTHON Standard Library giving a rich set of functions manipulating files, strings, etc.

Project Synopsis

The proposed system of **Banking System** has been designed keeping in mind the objective facilitate a fully computerized automated system over the existing manual system. It has been observed that the present system contains many loop-holes which lead to faulty system outputs and data redundancy. Lots of manual/handwritten processes are involved in the current system which have been improved and converted to automated forms to produce error-less outputs.

The proposed system has the following silent features:

1. All the modules are loosely coupled and highly cohesive.
2. Lesser user intervention is achieved.
3. Variety of reports generation introduced.
4. Hassle free data entry.

Hardware Requirement

CPU	:	Intel i3 or higher processor
RAM	:	1GB or higher
Hard Disk	:	1TB or higher
Keyboard	:	Required for data entry
Mouse	:	Not required
CD Drive	:	Not required

Software Requirement

Operating System	:	Windows 7 or any higher OS
PYTHON Compiler	:	PYTHON

Installation Process:

The Banking System software is very easy to install and use.

The following steps need to follow to use this software:

1. Copy all the .py files under project directory of C: drive.
2. Open the main project file named **bankingsystem.py** to compile and RUN to execute.
3. Follow the instructions and use main menu for necessary actions.

BANKING SYSTEM - CODING

```
import os, platform, mysql.connector, pandas as pd

mydb=mysql.connector.connect(host="localhost",user="root", password="root",
database="Bank")

mycursor=mydb.cursor()


def AccInsert():

    L1={'Accno':int(input("Enter the Account number : ")),

    'name':input("Enter the Customer Name: "),

    'age':int(input("Enter Age of Customer : ")),

    'occu':str(input("Enter Occupation of Customer :")),

    'Address':input("Enter the Address of the Customer : "),

    'Mob':int(input("Enter the Mobile number : ")),

    'Aadharno':int(input("Enter the Aadhar number : ")),

    'Amt':float(input("Enter the Money Deposited : ")),

    'AccType':input("Enter the Account Type (Saving/RD/PPF/Current) : ")}


    sql="Insert into ACCOUNT(Accno
    ,Name,Age,occu,Address,Mob,Aadharno,amt,AccType) values(%s,%s,%s,
    %s,%s,%s, %s,%s,%s)"

    mycursor.execute(sql,tuple(L1.values()))

    sql="Insert into Amt(Accno, amtdeposit, month) values(%s,%s,%s)"

    mycursor.execute(sql,(L1['Accno'],L1['Amt'],str(input("Enter month of Deposit :
        "))))

    mydb.commit()


def AccView():

    print("Select the search criteria :\n1. Acc no\n2. Name\n3. Mobile\n4.
    Aadhar\n5. View All")
```

```
    ch=int(input("Enter the choice : "))
    if ch==1:
        sql="select * from account where Accno=%s"
        mycursor.execute(sql,(int(input("Enter ACC no : ")),))
    elif ch==2:
        sql="select * from account where Name=%s"
        mycursor.execute(sql,(str(input("Enter Name : ")),))
    elif ch==3:
        sql="select * from account where Mob=%s"
        mycursor.execute(sql,(int(input("Enter Mobile No : ")),))
    elif ch==4:
        sql="select * from account where Aadharno=%s"
        mycursor.execute(sql,(str(input("Enter Aadhar : ")),))
    elif ch==5:
        sql="select * from account"
        mycursor.execute(sql)
    res=mycursor.fetchall()
    pd.set_option('display.max_columns', None)
    print("The Customer details are as follows :",
          pd.DataFrame(res,columns=['AcNo','Name','Age','Occn','Add','Mob','Aadh','Am
          t','AccTy']))


def AccDeposit():
    L2={'Accno':int(input("Enter the Account number :
    ")),'Amtdeposit':int(input("Enter the Amount to be deposited :
    ")),'month':str(input("Enter month of Deposit : "))}
    sql="Insert into amt(Accno,Amtdeposit,Month) values(%s,%s,%s)"
    mycursor.execute(sql,tuple(L2.values()))
    mydb.commit()
```

```
def accView():
    print("Please enter the details to view the Money details :")
    Accno=int(input("Enter the Account number of the Customer whose amount is to
    be viewed : "))
    sql="Select
    Account.Accno,Account.Name,Account.Age,Account.occu,Account.Address,
    Account.Mob,Account.Aadharno,Account.Amt,Account.AccType,
    sum(amt.Amtdeposit), amt.month from Account INNER JOIN amt ON
    Account.Accno=amt.Accno where Account.Accno=%s"
    mycursor.execute(sql,(Accno,))
    res=mycursor.fetchall()
    pd.set_option('display.max_columns', None)
    print("The Customer details are as follows : ",
          pd.DataFrame(res,columns=['AccNo','Name','Age','Occn','Add','Mob','Aadh','A
    mt', 'Accty','Tot_Deposit','Dep_Month']))


def closeAcc():
    Accno=int(input("Enter the Account number of the Customer to be closed : "))
    sql="Delete from amt where Accno=%s"
    mycursor.execute(sql,(Accno,))
    sql="Delete from Account where Accno=%s"
    mycursor.execute(sql,(Accno,))
    mydb.commit()


def MenuSet(): #Function For The Student Management System
    print(""" \n\n\nWelcome to the Banking System \n
          This system has the facility of maintaing Banking Records of Customers.
          This system facilitates Selective and Complete Addition/Deletion of
          Customer Banking Records\n\n""")
```

```
    print("Enter 1 : To Add Customer\nEnter 2 : To View Customer\nEnter 3 : To
    Deposit     Money\nEnter 4 : To Close Account\nEnter 5 : To View All Money
    Details of a          Customer")


    try:
        userInput = int(input("Please Select An Above Option:"))
    except ValueError:
        exit("\nHey! That's Not A Number") #Error Message
    else:
        print("\n")
        a={1:'AccInsert()',2:'AccView()',3:'AccDeposit()',4:'closeAcc()',5:'accView()'}
        if(userInput in a):
            eval(a[userInput])
        else:
            print("Enter correct choice. . . ")


MenuSet()


def runAgain():
    runAgn = input("\nwant To Run Again y/n: ")
    while(runAgn.lower() == 'y'):
        if(platform.system() == "Windows"):
            print(os.system('cls'))
        else:
            print(os.system('clear'))
        MenuSet()
        runAgn = input("\nwant To Run Again y/n: ")
runAgain()
```

OUTPUT

```
Welcome to the Banking System

This system has the facility of maintaing Banking Records of Customers.

This system facilitates Selective and Complete Addition/Deletion of Customer Banking Records
```

Snapshot of welcome screen

```
Enter 1 : To Add Customer
Enter 2 : To View Customer
Enter 3 : To Deposit Money
Enter 4 : To Close Account
Enter 5 : To View All Money Details of a Customer


Please Select An Above Option:
```

Snapshot of Main menu

```
Please Select An Above Option:1


Enter the Account number : 102
Enter the Customer Name: Bravo
Enter Age of Customer : 27
Enter Occupation of Customer :Teacher
Enter the Address of the Customer : H.No.123, Sector-1, Mumbai
Enter the Mobile number : 9786453120
Enter the Aadhar number : 400450056006
Enter the Money Deposited : 78000
Enter the Account Type (Saving/RD/PPF/Current) : Saving
Enter month of Deposit : March
```

Snapshot of New Customer Addition

```
Please Select An Above Option:2


Select the search criteria :
1. Acc no
2. Name
3. Mobile
4. Aadhar
5. View All
Enter the choice : 1

Enter ACC no : 102

The Customer details are as follows :

    AcNo   Name  Age     Occn                          Add         Mob  \
0   102  Bravo   27  Teacher  H.No.123, Sector-1, Mumbai  9786453120

           Aadh      Amt   AccTy
0  400450056006  78000.0  Saving
```

Snapshot of Account Details Viewing Screen

```
Please Select An Above Option:3


Enter the Account number : 102
Enter the Amount to be deposited : 16000
Enter month of Deposit : March

want To Run Again y/n: y
```

Snapshot of Amount Deposit Screen

```
Please Select An Above Option:4


Enter the Account number of the Customer to be closed : 101
Account closed Successfully!

want To Run Again y/n: y
```

Snapshot of Customer Account Closing Screen

```
Please Select An Above Option:5


Please enter the details to view the Money details :

Enter the Account number of the Customer whose amount is to be viewed : 102

The Customer details are as follows :

    AccNo   Name  Age     Occn                          Add         Mob  \
0     102  Bravo   27  Teacher  H.No.123, Sector-1, Mumbai  9786453120

           Aadh      Amt   AccTy  Tot_Deposit Dep_Month
0  400450056006  78000.0  Saving      94000.0     March
```

Snapshot of Customer's Money Details View