TABLE OF CONTENTS

01
Aim 1

02
Introduction 2

03
Technical specifications 3

04
Imported Modules. 4

05
Database details. 5

06
User-defined functions. . . . 7

07
Source code. 9

08
Sample output. 20

09
Conclusion. 30

10
Bibliography. 31

AIM

Gone are the days of having to wait in long queues to operate your bank account. With the development of the Bank Management System (BMS), the time-consuming and complex processes of opening a bank account, borrowing loans, paying the monthly EMI, depositing and withdrawing money are now possible at the click of a button.

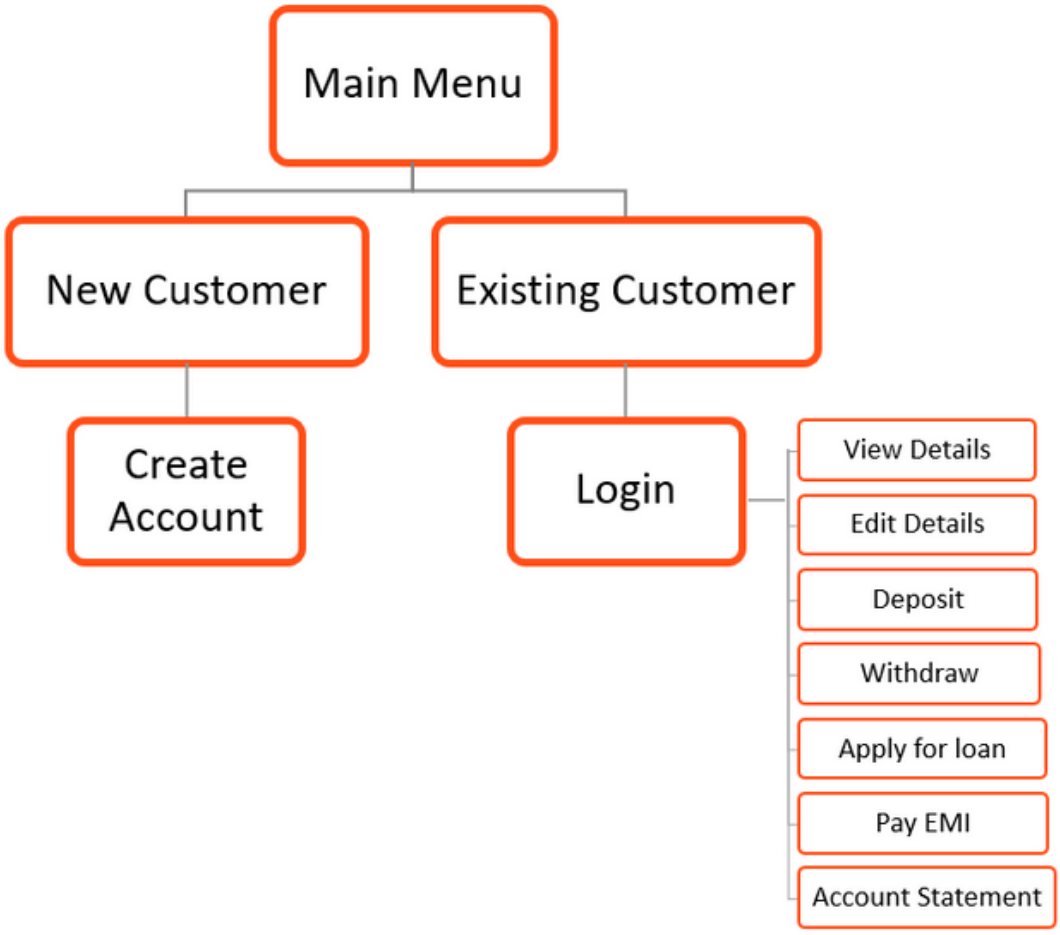
Designed with much efficiency, the BMS aims to ensure a secure and accessible banking experience for people across the country. With a user-friendly interface, the BMS intends to simplify the intimidating banking processes, thus extending the banking facility to people with limited financial literacy.

The BMS is also an attempt to enable people to manage and keep track of their own finances, reducing the large number of fraudulent activities experienced in the conventional system of banking.

INTRODUCTION

The BMS is a user-friendly interface that allows customers to efficiently manage bank activities through a menu-driven program.

On proceeding as a new customer, a computer-generated account number is provided to operate the account. Once the bank account has been created, users can log in as existing customers. The security of data is ensured through a password-protected login option.



TECHNICAL SPECIFICATIONS

HARDWARE SPECIFICATIONS:

PROCESSOR : PENTIUM DUAL CORE or Above

RAM : 2 GB

HARD DISK : 500 GB

SOFTWARE SPECIFICATIONS:

OPERATING SYSTEM : WINDOWS 10 or Above

FRONT END : Python 3 or Above (Thonny)

MIDDLEWARE : Python-MySQL Connector

BACKEND : MySQL

IMPORTED MODULES

1. import mysql.connector

- connect ( )
- cursor ( )
- fetchone ( )
- execute ()
- commit ()
- fetchall ()

2. import tabulate

- tabulate.tabulate()

3. import datetime

- datetime.date.today()

DATABASE DETAILS

DATABASE : Bank

TABLES :

```
mysql> show tables;
+-----------------+
| Tables_in_bank  |
+-----------------+
| customer        |
| loan            |
| transactions    |
+-----------------+
```

1. CUSTOMER - Stores the personal and account details of all customers.

```
mysql> desc customer;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| AccountNumber | int(10)     | NO   | PRI | NULL    | auto_increment |
| Name          | varchar(10) | NO   |     | NULL    |                |
| Age           | int(3)      | NO   |     | NULL    |                |
| Address       | varchar(15) | YES  |     | NULL    |                |
| Salary        | int(9)      | NO   |     | NULL    |                |
| Balance       | int(10)     | NO   |     | NULL    |                |
| Password      | varchar(10) | NO   |     | NULL    |                |
| Username      | varchar(10) | NO   |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
```

2. LOAN - Stores details about the loans borrowed by customers.

```
mysql> desc loan;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| AccountNumber | int(10)     | NO   | PRI | NULL    |       |
| LoanType      | varchar(10) | NO   |     | Null    |       |
| LoanAmount    | int(10)     | NO   |     | 0       |       |
| interest      | varchar(3)  | NO   |     | 0       |       |
| Duration      | float       | NO   |     | 0       |       |
| EMI           | int(10)     | NO   |     | 0       |       |
| PendingEMIs   | int(2)      | NO   |     | 0       |       |
+---------------+-------------+------+-----+---------+-------+
```

3. TRANSACTIONS - Stores details about the transactions done from an account which helps to produce the monthly account statement.

```
mysql> desc transactions;
+-----------------+-------------+------+-----+---------+----------------+
| Field           | Type        | Null | Key | Default | Extra          |
+-----------------+-------------+------+-----+---------+----------------+
| sno             | int(2)      | NO   | PRI | NULL    | auto_increment |
| AccountNumber   | int(10)     | NO   | MUL | NULL    |                |
| Activity        | varchar(10) | NO   |     | NULL    |                |
| Amount          | int(9)      | NO   |     | NULL    |                |
| TransactionDate | date        | NO   |     | NULL    |                |
+-----------------+-------------+------+-----+---------+----------------+
```

USER-DEFINED FUNCTIONS

S.NO.	FUNCTION	USE
1	`create_account()`	Asks the user to enter personal information to create the account and returns an auto-generated account number ★ Called when the user selects to continue as a **New customer**
2	`view_details(accno)`	Allows the user to view all details associated with a given account ★ Called when the user chooses option 1, **View details** from the menu
3	`edit_detail(accno)`	Allows the user to update information associated with the account ★ Called when the user chooses option 2, **Edit details** from the menu
4	`deposit(accno)`	Allows the user to deposit a desired sum of money to the account and displays the updated balance ★ Called when the user chooses option 3, **Deposit money** from the menu
5	`withdraw(accno)`	Allows the user to withdraw a desired sum of money from the account and displays the updated balance ★ Called when the user chooses option 4, **Withdraw money** from the menu

S.NO.	FUNCTION	USE
6	`loan(accno)`	Allows the user to apply for a loan by choosing the loan type, amount and duration. Displays the EMI to be paid ★ Called when the user chooses option 5, **Apply for a loan** from the menu
7	`pay_emi(accno)`	Allows the user to pay the monthly EMI ★ Called when the user chooses option 6, **Pay EMI** from the menu
8	`account_statement(accno)`	Display all the transactions for a given month in tabular form ★ Called when the user chooses option 7, **View account statement** from the menu
9	`close_account(accno)`	Deletes the account if there is no loan to be repaid ★ Called when the user chooses option 8, **Close account** from the menu

SOURCE CODE

```
import mysql.connector
import tabulate

def create_account():
    try:
        mycon=mysql.connector.connect(host='localhost',
        user='root',passwd='rujul8',database='bank')
        mycur=mycon.cursor()
        print("\nTo create an account, please enter the
        following details")
        c_name=input("Enter name: ")
        c_age=int(input("Enter age: "))
        if c_age<18:
            print("\n\033[1;31mSorry! you are not eligible to
            open an account")
        else:
            c_address=input("\033[0;30mEnter address: ")
            c_salary=int(input("Enter salary: "))
            c_amount=int(input("Enter initial amount: "))
            if c_amount<3000:
                print("\n\033[1;31mMinimum deposit required is
                3000")
                c_amount=int(input("\033[0;30mEnter initial
                amount: "))
            c_passwd=input("Create account password: ")
            c_username=input("Create account username: ")
            mycur.execute(f"insert into customer (Name,Age,
            Address,Salary,Balance,Password,Username)values
            ('{c_name}',{c_age},'{c_address}',{c_salary},
            {c_amount},'{c_passwd}','{c_username}')")
            mycon.commit()
            print("\nAccount has been created !")
            mycur.execute(f"select AccountNumber from customer
            where Name='{c_name}' and Age={c_age}and
            Address='{c_address}' and Salary={c_salary}")
            for i in mycur.fetchone():
                print("\n\033[1;36mYour account number is",i)
                mycur.execute(f"insert into loan
                (AccountNumber)values ({i})")
                mycon.commit()
```

```
     except Exception as e:
         print(e)

    mycur.close()
    mycon.close()

def close_account(accno):
    try:
        mycon=mysql.connector.connect(host='localhost',
        user='root',passwd='rujul8',database='bank')
        mycur=mycon.cursor()
        mycur.execute(f"select * from customer where
        AccountNumber={accno}")
        r=mycur.fetchone()
        if r==None:
            print("\033[1;31mInvalid account number")
        else:
            mycur.execute(f"select LoanAmount from Loan where
            AccountNumber={accno}")
            for i in mycur.fetchone():
                if i==0:
                    mycur.execute(f"delete from customer where
                    AccountNumber={accno}")
                    mycon.commit()
                    print("\n\033[1;32mAccount has been
                    deleted successfully")
                else:
                    print("\n\033[1;31mLoan pending....Account
                    cannot be closed")

    except Exception as e:
        print(e)

    mycur.close()
    mycon.close()


def deposit(accno):
    try:
        import datetime
        mycon=mysql.connector.connect(host='localhost',
        user='root',passwd='rujul8',database='bank')
        mycur=mycon.cursor()
        mycur.execute(f"select * from customer where
        AccountNumber={accno}")
        r=mycur.fetchone()
```

```
        if r==None:
            print("\033[1;31mInvalid account number")
        else:
            c_credit=int(input("\033[0;30mEnter amount to be
            deposited: "))
            mycur.execute(f"update customer set Balance=
            balance+{c_credit} where AccountNumber={accno}")
            mycon.commit()
            mycur.execute(f"select balance from customer where
            AccountNumber={accno}")

            for i in mycur.fetchone():
                balance2=i
            print("\n\033[1;32mTransaction successful!")
            print("\033[0;30mYour current balanceis",balance2)
            mycur.execute(f"insert into Transactions
            (AccountNumber, Activity,Amount,TransactionDate)
            values({accno},'credit',{c_credit},
            '{datetime.date.today()}')")
            mycon.commit()

    except Exception as e:
        print(e)

    mycur.close()
    mycon.close()


def withdraw(accno):
    try:
        import datetime
        mycon=mysql.connector.connect(host='localhost',
        user='root',passwd='rujul8',database='bank')
        mycur=mycon.cursor()
        mycur.execute(f"select * from customer where
        AccountNumber={accno}")
        r=mycur.fetchone()
        if r==None:
            print("\n\033[1;31mInvalid account number")
        else:
            c_debit=int(input("\033[0;30mEnter amount to be
            withdrawn: "))
            mycur.execute(f"select balance from customer where
            AccountNumber={accno}")
```

```
            for i in mycur.fetchone():
                balance1=i
              if balance1>=c_debit:
                mycur.execute(f"update customer set Balance=
                balance-{c_debit}where AccountNumber={accno}")
                mycon.commit()
                mycur.execute(f"select balance from customer
                where AccountNumber={accno}")

                for i in mycur.fetchone():
                    balance2=i
                print("\n\033[1;32mTransaction successful!")
                print("\033[0;30mYour current balance
                is",balance2)
                mycur.execute(f"insert into Transactions
                (AccountNumber, Activity,Amount,
                TransactionDate)values({accno},'debit',
                {c_debit},'{datetime.date.today()}')")
                mycon.commit()
            else:
                print("\033[1;31mTransaction failed! Account
                balance is not sufficient")

    except Exception as e:
        print(e)

    mycur.close()
    mycon.close()


def account_statement(accno):
    try:
        mycon=mysql.connector.connect(host='localhost',
        user='root',passwd='rujul8',database='bank')
        mycur=mycon.cursor()
        mycur.execute(f"select * from customer where
        AccountNumber={accno}")
        r=mycur.fetchone()
        if r==None:
            print("\n\033[1;31mInvalid account number")
        else:
            c_month=input("\n\033[0;30mEnter month: ")
            mycur.execute(f"select Activity,Amount,
            TransactionDate from Transactions where
            monthname(TransactionDate)='{c_month}' and
            AccountNumber={accno}")
```

```
            rs=mycur.fetchall()
            print('\n\033[0;36m'+20*'*','ACCOUNT STATEMENT',
            20*'*')
            print(tabulate.tabulate(rs,headers=['Account
            Activity','Amount','Transaction
            Date'],showindex='always',tablefmt='fancy_grid'))

    except Exception as e:
        print(e)

    mycur.close()
    mycon.close()


def loan(accno):
    try:
        mycon=mysql.connector.connect(host='localhost',
        user='root',passwd='rujul8',database='bank')
        mycur=mycon.cursor()
        mycur.execute(f"select * from customer where
        AccountNumber={accno}")
        rec=mycur.fetchone()
        if rec==None:
            print("\n\033[1;30mInvalid account number")
        else:
            mycur.execute(f"select Salary from customer where
            AccountNumber={accno}")
            rs=mycur.fetchone()
            for i in rs:
                if i<15000:
                    print("\n\033[1;31mSorry, you are not
                    eligible to apply for a loan.")
                else:
                    rec=(['Student','05 years','4%'],
                    ['Student','10 years','6%'],['Auto',
                    '05 years','6%'],['Auto','08 years','9%'],
                    ['Mortgage','10 years','2%'],['Mortgage',
                    '25 years','5%'],['Personal','03 years',
                    '4%'],['Personal','05 years','7%'])
                    print(tabulate.tabulate(rec,headers=['Loan
                    Type','Duration','Interest Rate'],
                    tablefmt='fancy_grid'))
                    print("\n\033[0;30mChoose type of loan:")
                    print("1. Student Loan")
                    print("2. Auto Loan")
                    print("3. Mortgage Loan")
                    print("4. Personal Loan")
```

```
                    ch=int(input("\nEnter choice: "))
                    p=int(input("Enter loan amount: "))
                    if p<=6*i:
                        t=float(input("Enter duration of
                        loan: "))
                        m=t*12

                        if ch==1 and t==5:
                            r=0.04/12
                            type='student'
                        elif ch==1 and t==10:
                            r=0.06/12
                            type='student'
                        elif ch==2 and t==5:
                            r=0.06/12
                            type='auto'
                        elif ch==2 and t==8:
                            r=0.09/12
                            type='auto'
                        elif ch==3 and t==10:
                            r=0.02/12
                            type='mortgage'
                        elif ch==3 and t==25:
                            r=0.05/12
                            type='mortgage'
                        elif ch==4 and t==3:
                            r=0.04/12
                            type='personal'
                        elif ch==4 and t==5:
                            r=0.07/12
                            type='personal'
                        else:
                            print("\n\033[1;31mInvalid value
                            entered")
                        x=(1+r)**m
                        y=(1+r)**m-1
                        c_EMI=p*r*x//y
                        rate=r*1200
                        mycur.execute(f"update loan set
                        LoanType='{type}',LoanAmount={p},
                        Interest={rate},Duration={t},EMI=
                        {c_EMI},PendingEMIs={m} where Account
                        Number={accno}")
                        mycon.commit()
                        print("\n\033[1;32mLoan has been
                        granted")
                        print("\033[0;30mThe EMI to be paid
                        is",c_EMI)
```

```
                    else:
                        print("\n\033[1;31mLoan request
                        denied")

    except Exception as e:
        print(e)

    mycur.close()
    mycon.close()


def pay_emi(accno):
    import datetime
    try:
        mycon=mysql.connector.connect(host='localhost',
        user='root',passwd='rujul8',database='bank')
        mycur=mycon.cursor(buffered=True)
        mycur.execute(f"select * from customer where
        AccountNumber={accno}")
        r=mycur.fetchone()
        if r==None:
            print("\n\033[1;31mInvalid account number")

        else:
            mycur.execute(f"select LoanAmount,EMI,PendingEMIs
            from loan where AccountNumber={accno}")
            r=mycur.fetchone()
            loan=r[0]
            EMI=r[1]
            num=r[2]
            if loan==0:
                print("\n\033[1;31mNo loan borrowed!")
            else:
                t=datetime.date.today()
                mycur.execute(f"select transactiondate from
                transactions where Accountnumber={accno}
                and activity='EMI' and month(transactiondate)
               ={t.month} and year(transactiondate)={t.year}")
                d=mycur.fetchone()
                for i in d:
                    if i==None:
                        mycur.execute(f"select balance from
                        customer where AccountNumber=
                        {accno}")
                        for j in mycur.fetchone():
                            balance=j
```

```
                        v1=balance-EMI
                        if v1>=0:
                            mycur.execute(f"update customer
                            set balance={v1} where
                            AccountNumber={accno}")
                            mycon.commit()

                            v2=num-1
                            mycur.execute(f"update loan set
                            PendingEMIs={v2} where
                            AccountNumber={accno}")
                            mycon.commit()

                            print("\n\033[1;32mTransaction
                            succesful!")
                            print("\033[0;30mYour available
                            balance is",v1)
                            mycur.execute(f"insert into
                            Transactions (AccountNumber,
                            Activity,Amount,TransactionDate)
                            values({accno},'EMI',{EMI},
                            '{datetime.date.today()}')")
                            mycon.commit()
                            if v2==0:
                                print("\n\033[1;32mLoan has
                                been cleared")
                        else:
                            print("\n\033[1;31mAccount balance
                            is not sufficient to pay EMI")
                    else:
                        print("\n\033[1;32mEMI for this month
                        has been paid!")

    except Exception as e:
         print(e)

    mycur.close()
    mycon.close()


def view_details(accno):
    try:
        mycon=mysql.connector.connect(host='localhost',
        user='root',passwd='rujul8',database='bank')
        mycur=mycon.cursor()
        mycur.execute(f"select * from customer where
        AccountNumber={accno}")
```

```
        r=mycur.fetchone()
        if r==None:
            print("\n\033[1;31mInvalid account number")
        else:
            mycur.execute(f"select Name,Age,Address,Salary,
            Balance,LoanAmount,LoanType,Interest,Duration,
            EMI,PendingEMIs from customer c, loan l where
            c.AccountNumber={accno} and c.AccountNumber=
            l.AccountNumber")
            r=mycur.fetchall()

            print(tabulate.tabulate(r,headers=['Name',
            'Age','Address','Salary','Balance','Loan Amount',
            'Loan Type','Interest','Duration','EMI','Pending
            EMIs'],tablefmt='fancy_grid'))

    except Exception as e:
        print(e)

    mycur.close()
    mycon.close()


def edit_details(accno):
    try:
        mycon=mysql.connector.connect(host='localhost',
        user='root',passwd='rujul8',database='bank')
        mycur=mycon.cursor()
        mycur.execute(f"select * from customer where
        AccountNumber={accno}")
        r=mycur.fetchone()
        if r==None:
            print("\n\033[1;31mInvalid account number")
        else:
            print("\n\033[0;30mPlease enter details:")
            c_name=input("Name: ")
            c_age=int(input("Age: "))
            c_address=input("Address: ")
            c_salary=int(input("Salary: "))
            c_passwd=input("Account password: ")
            c_username=input("Account username: ")
            mycur.execute(f"update customer set Name=
            '{c_name}',Age={c_age},Address='{c_address}',
            Salary={c_salary},password='{c_passwd}',username=
            '{c_username}' where AccountNumber={accno}")
            mycon.commit()
            print("\n\033[1;32mDetails have been updated")
```

```
    except Exception as e:
        print(e)

    mycur.close()
    mycon.close()



while True:
    txt="\033[1;96mWELCOME TO ABU DHABI CITY BANK!"
    s='_'
    print("\n\033[1;96m",s*35,txt,s*35)
    print("\n\033[1;90m1. Existing customer")
    print("2. New customer")
    print("3. Exit")
    op1=int(input("\n\033[0;30mEnter option number: "))
    if op1==1:
        accno=int(input("Enter account number: "))
        user=input("Enter username: ")
        pw=input("Enter password: ")
        mycon=mysql.connector.connect(host='localhost',
        user='root',passwd='rujul8',database='bank')
        mycur=mycon.cursor()
        mycur.execute(f"select password,username from customer
        where AccountNumber={accno}")
        r=mycur.fetchone()
        if r==None:
            print("Account does not exist")
        elif user==r[1] and pw==r[0]:
            print("\n\033[1;32m Login successful!")
            while True:
                print("\n\033[1;30m 1. View details")
                print(" 2. Edit details")
                print(" 3. Deposit money")
                print(" 4. Withdraw money")
                print(" 5. Apply for a loan")
                print(" 6. Pay EMI")
                print(" 7. View account statement")
                print(" 8. Close account")
                print(" 9. Exit\n")
                op2=int(input("\n\033[0;30mEnter option
                number: "))
                if op2==1:
                    view_details(accno)
                elif op2==2:
                    edit_details(accno)
                elif op2==3:
                    deposit(accno)
```

```
                elif op2==4:
                    withdraw(accno)
                elif op2==5:
                    loan(accno)
                elif op2==6:
                    pay_emi(accno)
                elif op2==7:
                    account_statement(accno)
                elif op2==8:
                    close_account(accno)
                elif op2==9:
                    print("\n\033[1;36mThank you for choosing
                    Abu Dhabi City Bank!")
                    break
                else:
                    print("Invalid option number")
        else:
            print("\033[1;31mIncorrect username or password")
            continue
    elif op1==2:
        create_account()
        print("\n\033[0;30mTo operate your account choose
        option 1")
    elif op1==3:
        print("\n\033[1;36mThank you for choosing Abu Dhabi
        City Bank!")
        break
    else:
        print("Invalid option number")
```

SAMPLE OUTPUT

```
_____ WELCOME TO ABU DHABI CITY BANK! _____

1. Existing customer
2. New customer
3. Exit

Enter option number: 2

To create an account, please enter the following details
Enter name: Tina
Enter age: 17

Sorry! you are not eligible to open an account
```

```
_____ WELCOME TO ABU DHABI CITY BANK! _____

1. Existing customer
2. New customer
3. Exit

Enter option number: 2

To create an account, please enter the following details
Enter name: Rujul
Enter age: 35
Enter address: Al Zahiyah
Enter salary: 40000
Enter initial amount: 2000

Minimum deposit required is 3000
Enter initial amount: 4500
Create account password: rujul123
Create account username: rujulm

Account has been created !

Your account number is 1010003

To operate your account choose option 1
```

```
_____ WELCOME TO ABU DHABI CITY BANK! _____

1. Existing customer
2. New customer
3. Exit

Enter option number: 1
Enter account number: 10100000
Enter username: rujulm
Enter password: rujul123

Account does not exist
```

```
_____ WELCOME TO ABU DHABI CITY BANK! _____

1. Existing customer
2. New customer
3. Exit

Enter option number: 1
Enter account number: 1010003
Enter username: rujulm
Enter password: rujul

Incorrect username or password
```

```
_____ WELCOME TO ABU DHABI CITY BANK! _____

1. Existing customer
2. New customer
3. Exit

Enter option number: 1
Enter account number: 1010003
Enter username: rujulm
Enter password: rujul123

Login successful!

1. View details
2. Edit details
3. Deposit money
4. Withdraw money
5. Apply for a loan
6. Pay EMI
7. View account statement
8. Close account
9. Exit
```

```
Enter option number: 1
```

Name	Age	Address	Salary	Balance	Loan Amount	Loan Type	Interest	Duration	EMI	Pending EMIs
Rujul	35	Al Zahiyah	40000	4500	0	Null	0	0	0	0

```
1. View details
2. Edit details
3. Deposit money
4. Withdraw money
5. Apply for a loan
6. Pay EMI
7. View account statement
8. Close account
9. Exit

Enter option number: 2

Please choose the information to be edited:
1. Address
2. Salary
3. Account password
4. Account username
Enter option number: 2
Salary: 45000

Details have been updated

1. View details
2. Edit details
3. Deposit money
4. Withdraw money
5. Apply for a loan
6. Pay EMI
7. View account statement
8. Close account
9. Exit

Enter option number: 1
```

Name	Age	Address	Salary	Balance	Loan Amount	Loan Type	Interest	Duration	EMI	Pending EMIs
Rujul	35	Al Zahiyah	45000	4500	0	Null	0	0	0	0

```
1. View details
2. Edit details
3. Deposit money
4. Withdraw money
5. Apply for a loan
6. Pay EMI
7. View account statement
8. Close account
9. Exit


Enter option number: 3
Enter amount to be deposited: 200
Transaction successful!
Your current balance is 4700
```

```
1. View details
2. Edit details
3. Deposit money
4. Withdraw money
5. Apply for a loan
6. Pay EMI
7. View account statement
8. Close account
9. Exit

Enter option number: 4
Enter amount to be withdrawn: 5000
Transaction failed! Account balance is not sufficient
```

```
1. View details
2. Edit details
3. Deposit money
4. Withdraw money
5. Apply for a loan
6. Pay EMI
7. View account statement
8. Close account
9. Exit

Enter option number: 4
Enter amount to be withdrawn: 1000

Transaction successful!
Your current balance is 3700
```

```
1. View details
2. Edit details
3. Deposit money
4. Withdraw money
5. Apply for a loan
6. Pay EMI
7. View account statement
8. Close account
9. Exit
```

```
Enter option number: 5
```

Loan Type	Duration	Interest Rate
Student	05 years	4%
Student	10 years	6%
Auto	05 years	6%
Auto	08 years	9%
Mortgage	10 years	2%
Mortgage	25 years	5%
Personal	03 years	4%
Personal	05 years	7%

```
Choose type of loan:
1. Student Loan
2. Auto Loan
3. Mortgage Loan
4. Personal Loan

Enter choice: 2
Enter loan amount: 2500000

Loan request denied
```

```
1. View details
2. Edit details
3. Deposit money
4. Withdraw money
5. Apply for a loan
6. Pay EMI
7. View account statement
8. Close account
9. Exit
```

```
Enter option number: 5
```

Loan Type	Duration	Interest Rate
Student	05 years	4%
Student	10 years	6%
Auto	05 years	6%
Auto	08 years	9%
Mortgage	10 years	2%
Mortgage	25 years	5%
Personal	03 years	4%
Personal	05 years	7%

```
Choose type of loan:
1. Student Loan
2. Auto Loan
3. Mortgage Loan
4. Personal Loan

Enter choice: 2
Enter loan amount: 250000
Enter duration of loan: 8

Loan has been granted
The EMI to be paid is 3662.0
```

```
1. View details
2. Edit details
3. Deposit money
4. Withdraw money
5. Apply for a loan
6. Pay EMI
7. View account statement
8. Close account
9. Exit

Enter option number: 1
```

Name	Age	Address	Salary	Balance	Loan Amount	Loan Type	Interest	Duration	EMI	Pending EMIs
Rujul	35	Al Zahiyah	45000	3700	250000	auto	9	8	3662	96

```
 1. View details
 2. Edit details
 3. Deposit money
 4. Withdraw money
 5. Apply for a loan
 6. Pay EMI
 7. View account statement
 8. Close account
 9. Exit


Enter option number: 4
Enter amount to be withdrawn: 500
```

```
Transaction successful!
Your current balance is 3200

 1. View details
 2. Edit details
 3. Deposit money
 4. Withdraw money
 5. Apply for a loan
 6. Pay EMI
 7. View account statement
 8. Close account
 9. Exit

Enter option number: 6

Account balance is not sufficient to pay EMI

 1. View details
 2. Edit details
 3. Deposit money
 4. Withdraw money
 5. Apply for a loan
 6. Pay EMI
 7. View account statement
 8. Close account
 9. Exit

Enter option number: 3
Enter amount to be deposited: 2000

Transaction successful!
Your current balance is 5200
```

```
1. View details
2. Edit details
3. Deposit money
4. Withdraw money
5. Apply for a loan
6. Pay EMI
7. View account statement
8. Close account
9. Exit

Enter option number: 6

Transaction succesful!
Your available balance is 1538

1. View details
2. Edit details
3. Deposit money
4. Withdraw money
5. Apply for a loan
6. Pay EMI
7. View account statement
8. Close account
9. Exit

Enter option number: 1
```

Name	Age	Address	Salary	Balance	Loan Amount	Loan Type	Interest	Duration	EMI	Pending EMIs
Rujul	35	Al Zahiyah	45000	1538	250000	auto	9	8	3662	95

```
1. View details
2. Edit details
3. Deposit money
4. Withdraw money
5. Apply for a loan
6. Pay EMI
7. View account statement
8. Close account
9. Exit

Enter option number: 6

EMI for this month has been paid!
```

```
1. View details
2. Edit details
3. Deposit money
4. Withdraw money
5. Apply for a loan
6. Pay EMI
7. View account statement
8. Close account
9. Exit

Enter option number: 7

Enter month: december
```

********** ACCOUNT STATEMENT **********

	Account Activity	Amount	Transaction Date
0	credit	200	2022-12-08
1	debit	1000	2022-12-08
2	debit	500	2022-12-08
3	credit	2000	2022-12-08
4	EMI	3662	2022-12-08

```
1. View details
2. Edit details
3. Deposit money
4. Withdraw money
5. Apply for a loan
6. Pay EMI
7. View account statement
8. Close account
9. Exit

Enter option number: 8

Loan pending....Account cannot be closed
```

```
 1. View details
 2. Edit details
 3. Deposit money
 4. Withdraw money
 5. Apply for a loan
 6. Pay EMI
 7. View account statement
 8. Close account
 9. Exit


Enter option number: 9
```

```
                  WELCOME TO ABU DHABI CITY BANK!                  

1. Existing customer
2. New customer
3. Exit

Enter option number: 3

Thank you for choosing Abu Dhabi City Bank!
```

CONCLUSION

The requirements as expected in the beginning were fulfilled. All menu options and functions are working successfully. The BMS thus proves to be an efficient banking system, made user-friendly by the use of Python.

However, a few limitations observed are the lack of options such as Fixed Deposits, Demand Drafts, and Money Transfer. Also, the BMS is easy to hack into allowing data to be manipulated by anyone once the password has been by cracked.

The program was tested quite a few times, obtaining satisfactory results. Hence this software project has been developed to fulfill the academic learning in CBSE Class XII, computer science curriculum

BIBLIOGRAPHY

- Computer Science with Python textbook for CBSE Class XII - Preeti Arora

- https://docs.python.org

- https://stackoverflow.com