

ATM MANAGER

**COMPUTER SCIENCE PROJECT**

**PROJECT FILE 2020-21**

SUBMITTED TO:

Ms. MEENA DOGRA

NAME: UJJWAL JAIN

CLASS: 12-C

BOARD ROLL NO:

ACKNOWLEDGEMENT

The project shall be incomplete without showing gratitude and reverence to those who have helped me in making this assignment successful. First of all, I would thank lord almighty who has always been with me and showered his blessings upon me. It is a pious reference to express my most profound gratitude and deep regards to my guide Ms. Meena Dogra for her exemplary guidance providing me constant encouragement as well as information throughout the course of this project and clearing my doubts regarding the topic through which I learnt many new things.

I would also like to thank my parents who have helped me and guided me at every step. I would also like to thank my friends who helped me in finalizing this project within the limited time frame.

**index**

|  |  |  |
| --- | --- | --- |
| SR. NO. | CONTENT | PAGE |
| 1. | Certificate | 3 |
| 2. | Technology used  2.1 Frontend : Python  2.2 Backend : MySQL | 4 |
| 3. | System Implementation  3.1 The Hardware used  3.2 The Software used | 7 |
| 4. | Case study of the Project | 8 |
| 5. | Advantages to End-User | 9 |
| 6. | Facilities Available To an user | 10 |
| 7. | Project Details   * 1. Frontend : Python   7.1.1 Modules used   * 1. Backend : MySQL      1. Database used      2. Tables used | 11 |
| 8. | Project Code | 12 |
| 9. | Output Screens | 17 |
| 10. | Future Scope of the Project | 21 |
| 11. | Bibliography | 22 |
| 12. | Teacher Remarks | 23 |
| 13. | Software Disk | 24 |

**Certificate**

This is to certify that

**UJJWAL JAIN**

student of

Class XII- C

Dewan Public School, Meerut Cantt.

has successfully completed

the project work on

**ATM MANAGER**

under the guidance of

**Ms. MEENA DOGRA**

During the period of making this project

he was found to be punctual

and hard working.

**TEACHER’S SIGNATURE TEACHER’S SIGNATURE**

**(Internal) (External)**

**2. technology USED**

2.1 FRONT-END: PYTHON

Python is a dynamic, object-oriented and general-purpose programming language. It offers strong support for integrating with other technologies, higher programmer productivity throughout the development lifecycle, and is particularly well suited for large or complex projects with changing requirements.

It was created by Guido Van Rossum when he was working at CWI which is a National Research Institute for Mathematics and Computer Science in Netherlands. The language was released in 1991. Python got its name from a BBC comedy series from seventies- “Monty Python’s Flying Circus”. Python can be used to follow both Procedural approach and Object Oriented approach of programming. It is free to use.

Python’s clean object-oriented design and extensive support libraries offer two to tenfold the programmer productivity seen with languages like C, C++, C#, Java, VB and Pert. Python is available for most operating systems, including Windows, UNIX, Linux and Mac OS.

Python is being used to develop applications for varied domains likeWeb applications, Desktop applications, GUI applications, Mobile applications, Gaming applications, Data Analysis, Data Sciences, Data Visualization, Machine learning and Scientific applications.

Python is being used by a number of popular websites and organizations, some of them being, Google, Facebook, YouTube, NASA, IBM. Yahoo, Instagram, Reddit, Amazon, Uber, Pinterest and Netflix.

**Some of the features of Python are:**

1. **Free and open source**

Python is freely downloadable from <https://www.python.org/downloads/> **(Free)**. The source code of Python is available **(open).**

1. **High level language**

It is close to human-understandable language that makes it easy to understand, read and manage.

1. **General purpose programming language**

Can be used to write code/software for a wide variety of application domains.

1. **Interpreted**

Executes the source code line by line. Execution stops when either an error is encountered in any code statement or all the lines have been successfully executed.

1. **Object-oriented programming language**

Python is an Object-Oriented Programming language and supports all the features of OOP.

1. **Dynamically typed language**

In Python, the type of an object/variable is decided at runtime, not in advance. The type of an object is determined by the value assigned to it.

1. **Integrated language**

We can easily integrated Python with other languages like C,C++ etc.

1. **GUI programming**

Python offers various libraries like Tkinter, wxPython for making a Graphical user interface for our applications.

1. **Huge Community**

Python has a huge community of developers that are available for support of all types.

1. **Portable**

Same python code can execute on Windows, MAC, Linux, Unix without making any changes.

1. **Easy to code and read**

With its simple and concise syntax and short code length, Python is considered to be a Beginner - friendly programming language.

1. **Large standard library** - **“Batteries included”**

Python’s standard library provides a rich set of modules and functions so that users do not have to write their own code for every single thing.

2.2 BACK-END: MySQL

MySQL, the most popular Open Source SQL Database Management System, is developed, distributed, and supported by Oracle Corporation.

**Some of the features of MySQL are:**

1. **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, we need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

1. **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment.

The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases.

1. **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the MySQL website <http://www.mysql.com/> and use it without paying anything. If one wishes, one may study the source code and change it to suit his/her needs. The MySQL software uses the GPL (GNU General Public License, to define what we may and may not do with the software in different situations.

The MySQL website <http://www.mysql.com/> also provides the latest information about MySQL software.

1. **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

MySQL Server can run comfortably on a desktop or laptop, alongside our other applications, web servers, and so on, requiring little or no attention. If we dedicate an entire machine to MySQL, we can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

1. **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

MySQL Server as an embedded library can be linked into an application to get a smaller, faster, easier-to-manage standalone product.

1. **Works on many different platforms.**

MySql can run on multiple platforms like Windows, Linux and Unix.

**3. SYSTEM IMPLEMENTATION**

3.1 Hardware used

PC with INTEL i7 processor with 8GB RAM and other required devices.

3.2 Software used

- Microsoft Windows® 10 as Operating System.

- Python 3.8.6 as Front-end Development environment.

- MySQL as Back-end.

- MS-Word 2016 for documentation.

**4. CASE STUDY OF PROJECT**

An ATM is a useful device, used to perform several banking tasks digitally at any time irrespective of the work hours of a bank. ATMs enable individuals to make banking transactions without the help of an actual teller. Also, customers can avail banking services without having to visit a bank branch. Most ATM transactions can be availed with the use of a debit or credit card. This project named “ATM MANAGER” is a useful program created using Python, MySQL and MySQL-connector to perform functions of an ATM. This is the future of banking as people can avail banking facilities at any time in any nearby ATM. This is a user- friendly, well organised and easy to use program. This program can be used to perform various tasks, some of them are as follows:-

* Add details of a new customer.
* Setting first Credit/Debit Card PIN
* Changing existing Credit/Debit Card PIN
* Depositing Money
* Withdrawing Money
* Check Existing Balance

**5. ADVANTAGES TO END USER**

In the modern age people prefer to use ATM to perform transactions instead of performing them conventionally through a bank. The program helps user to easily perform all the banking functions that can be performed using an ATM.

1. Reduced Cost

Although, setting up the ATM software requires initial financial investment but over a longer period of time it leads to significant cost reductions. Using the program leads to reduction in requirement of human resurces and other maintenance costs.

1. Improved Customer Experience

Using the program reduces human interference in transactions performed by a user which leads to an improved customer experience.

1. Improves Resource Usage

Using the program helps in reducing the unnecessary usage of resources such as humans, papers, stamps and many other resources generally used in banks.

**6. FACILITIES AVAILABLE TO USER**

* Add New Account Details
* Create First Pin
* Change an Existing Pin
* Deposit money
* Withdraw Money(Limit of 25,000)
* Check Balance

**7. Project Details**

* 1. Frontend: Python

7.1.1Modules used

* Mysql.connector
* Random
  1. Backend: MySQL

7.2.1Database used

* ATM

7.2.2 Table used

* CUSTOMER

**8. project code**

'’’----------------------------------------------------------ATM MANAGING PROGRAM--------------------------------------------------  
---------------------------------------------------------DEVELOPED BY->UJJWAL JAIN------------------------------------------------  
-------------------------------------------------------------------CLASS-> XII-C--------------------------------------------------------------------------------------------------------------------SCHOOL->DEWAN PUBLIC SCHOOL--------------------------------------------'''  
l=[]  
import mysql.connector as my  
import random  
#MODULE TO CONNECT MYSQL  
def MySqlconnection():  
    Us=input('Enter your server username:')  
    Pd=input('Enter your server password:')  
    try:  
        global con  
        global cur  
        a=5  
        con=my.connect(host='localhost',user=Us,password=Pd)  
        print('Connection Successful!')  
        cur=con.cursor()  
        query="create database if not exists ATM "  
        cur.execute(query)  
        cur.execute("use ATM")  
        con.commit()  
         
    except:  
        a=4  
        if a==5:  
            pass  
        elif a==4 :  
            MySqlconnection()

**#MODULE TO ADD NEW CUSTOMER**  
def newcustomer():  
    query="CREATE TABLE IF NOT EXISTS CUSTOMER(Account\_Number Varchar(20) PRIMARY KEY,CNAME

VARCHAR(30), AGE INT(3),PHONE VARCHAR(12) ,DEBIT\_CARD\_NUMBER VARCHAR(17) UNIQUE,

DEBIT\_CARD\_PIN INT(5), CREDIT\_CARD\_NUMBER VARCHAR(17) UNIQUE,CREDIT\_CARD\_PIN INT(5),

BALANCE INT(15) DEFAULT 0, KYC\_TYPE VARCHAR(20))"  
    cur.execute(query)  
    print("Please Fill Information Carefully")  
    def account\_no():  
        global acc  
        acc=random.randrange(300000000000,399999999999)  
        acc=str(acc)  
        cur.execute("select Account\_Number from CUSTOMER")  
        rec=cur.fetchall()  
        if acc in rec:  
            account\_no()  
        else:  
            cur.execute("insert into CUSTOMER(ACCOUNT\_NUMBER) values ('%s')" % acc)  
            con.commit()  
            print('Your ACCOUNT NUMBER is',acc)  
    account\_no()  
    cname=input("Please enter name:")  
    age=int(input("Please Enter Age:"))  
    phone=input('Please enter Phone number:')  
    kyc=input('Please enter KYC Type:')  
    #CREDIT CARD GENERATION  
    c\_card=input('DO YOU WANT A CREDIT CARD?(Y/N):')  
    if c\_card.lower()=='y':  
        def credit():  
            cred=random.randrange(9000000000000000,9999999999999999)  
            cred=str(cred)  
            cur.execute("select CREDIT\_CARD\_Number from CUSTOMER")  
            rec=cur.fetchall()  
            if cred in rec:  
                credit()  
          else:  
                cur.execute("update CUSTOMER set CREDIT\_CARD\_NUMBER=%s where ACCOUNT\_NUMBER=

%s" % (cred, acc))  
                con.commit()  
                print('Your Credit Card Number is',cred)  
        credit()  
    else:  
        pass

**#DEBIT CARD GENERATION**    d\_card=input('DO YOU WANT A DEBIT CARD?(Y/N):')  
    if d\_card.lower()=='y':  
        def debit():  
            debt=random.randrange(6000000000000000,6999999999999999)  
            debt=str(debt)  
            cur.execute("select DEBIT\_CARD\_Number from CUSTOMER")  
            rec=cur.fetchall()  
            if debt in rec:  
                debit()  
            else:  
                cur.execute("update CUSTOMER set DEBIT\_CARD\_NUMBER=%s where ACCOUNT\_NUMBER=%s"

% (debt, acc))  
                con.commit()  
                print('Your Debit Card Number is',debt)  
        debit()  
    else:  
        pass  
     
    cur.execute("update CUSTOMER set CNAME='%s' where ACCOUNT\_NUMBER='%s'" % (cname, acc))  
    con.commit()  
    cur.execute("update CUSTOMER set AGE=%s where ACCOUNT\_NUMBER='%s'" % (age, acc))  
    con.commit()  
    cur.execute("update CUSTOMER set PHONE='%s' where ACCOUNT\_NUMBER='%s'" % (phone, acc))  
    con.commit()  
    cur.execute("update CUSTOMER set KYC\_TYPE='%s' where ACCOUNT\_NUMBER='%s'" % (kyc, acc))  
    con.commit()  
    print('New Customer Added Successfully!')

**#MODULE TO TAKE PIN AS AN INPUT**  
def pin():  
    global l  
    global x  
    x=int(input("ENTER FOUR DIGIT PIN:"))  
    y=int(input("RE-ENTER THE FOUR DIGIT PIN:"))  
    if x==y:  
        print("PIN Matched!")  
        l=0  
  
    elif x!=y:  
        l=1  
        i=0  
        while i<2:  
            k=i+1  
            print('PIN Not Matched! Please Try Again... ')  
            print("This is Your" ,k," Attempt and you are left with only",3-k,"Attempts")  
            i=i+1  
        else:  
            print("You have Reached Your Maximum Limit")

**#MODULE TO CREATE FIRST PIN**  
def New\_PIN():  
    a=input("Enter The Type of Card(Credit/Debit):")  
    if a.lower()=="credit":  
        m=input("Enter The Credit Card Number:")  
        pin()  
        if l==0:  
            query=("update CUSTOMER set CREDIT\_CARD\_PIN=%s where CREDIT\_CARD\_NUMBER=%s")  
            cur.execute(query,(x,m))  
            con.commit()  
            print("Password Set Successfully! ")  
        elif l==1:  
            pass  
         
    elif a.lower()== "debit":  
        n=input("Enter The Debit Card Number:")  
        pin()  
        if l==0:  
            query=("update CUSTOMER set DEBIT\_CARD\_PIN=%s where DEBIT\_CARD\_NUMBER=%s")  
            cur.execute(query,(x,n))  
            con.commit()  
            print("Password Set Successfully! ")  
        elif l==1:  
            pass  
  
**#MODULE TO CHANGE EXISTING PIN**  
def CHANGE\_PIN():  
    a=input("Enter The Type of Card(Credit/Debit):")  
    if a.lower()=="credit":  
        m=input("Enter The Credit Card Number:")  
        n=int(input("Enter Your Existing PIN:"))  
        query=("select CREDIT\_CARD\_PIN from CUSTOMER where CREDIT\_CARD\_NUMBER=%s") % m  
        cur.execute(query)  
        data=cur.fetchone()  
        try:  
            if n in data:  
                pin()  
                query=("update CUSTOMER set CREDIT\_CARD\_PIN=%s where CREDIT\_CARD\_NUMBER=%s")  
                cur.execute(query,(x,m))  
                con.commit()  
                print('PIN SUCCESSFULLY CHANGED')  
            else:  
                print('THE ENTERED CREDENTIALS DO NOT MATCH')  
  
        except TypeError:  
            print("Please check your card details!!")  
  
    elif a.lower()=="debit":  
        m=input("Enter The Debit Card Number:")  
        n=int(input("Enter Your Existing PIN:"))  
        query=("select DEBIT\_CARD\_PIN from CUSTOMER where DEBIT\_CARD\_NUMBER= %s") % m  
        cur.execute(query)  
        data=cur.fetchone()  
        try:  
            if n in data:  
                pin()  
                cur.execute("update CUSTOMER set DEBIT\_CARD\_PIN=%s where DEBIT\_CARD\_NUMBER=%s"

% (x,m))  
                con.commit()  
                print('PIN SUCCESSFULLY CHANGED')  
  
            else:  
                print('THE ENTERED CREDENTIALS DO NOT MATCH')  
  
        except TypeError:  
            print("Please check your card details!!")  
           

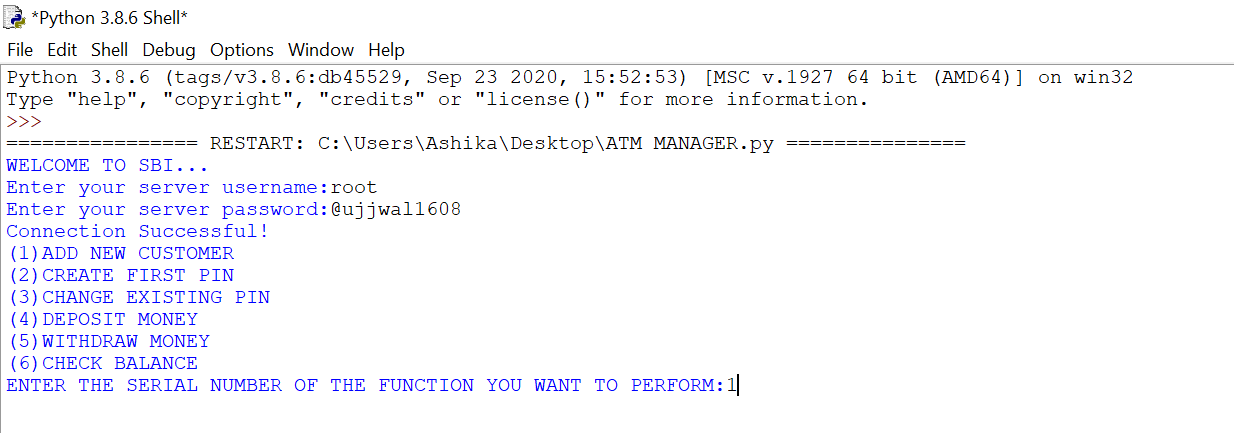
**#MODULE TO DEPOSIT MONEY**  
def Deposit():  
    n=input("Enter The Debit Card Number:")  
    cur.execute("select DEBIT\_CARD\_NUMBER from CUSTOMER")  
    data=cur.fetchall()  
    n=(n,)  
    if n in data:  
        x=int(input("Enter Your PIN:"))  
        for a in n:  
            query=("select DEBIT\_CARD\_PIN FROM CUSTOMER where DEBIT\_CARD\_NUMBER='%s'") % a  
            cur.execute(query)  
            inp=cur.fetchone()  
            if x in inp:  
                    d=int(input("Enter the Amount You Want To Deposit:"))  
                    cur.execute("update CUSTOMER set BALANCE= BALANCE+ %s where DEBIT\_CARD\_NUMBER='%s'" % (d,a))  
                    con.commit()  
                    cur.execute("select BALANCE from CUSTOMER where DEBIT\_CARD\_NUMBER='%s'" % a)  
                    data=cur.fetchone()  
                    for i in data:  
                        print('\*'\*60,'AMOUNT DEPOSITED IS',d,'\*'\*60)  
                        print('\*'\*58,'YOUR CURRENT BALANCE IS', i,'\*'\*58)  
                        print('\*'\*60,'Thank You For Visiting Us!','\*'\*60)  
                        print('\*'\*52,'Transaction Complete! Please Remove Your Card!','\*'\*52)  
  
            else:  
                print("Transaction Failed! Incorrect PIN!!!")  
    else:  
        print('Transaction Failed! Details Not Matched!')  
             
**#MODULE TO WITHDRAW MONEY**  
def withdraw():  
    n=input("Enter The Debit Card Number:")  
    cur.execute("select DEBIT\_CARD\_NUMBER from CUSTOMER")  
    data=cur.fetchall()  
    n=(n,)  
    if n in data:  
        for a in n:  
            x=int(input("Enter Your PIN:"))  
            cur.execute("select DEBIT\_CARD\_PIN from CUSTOMER where DEBIT\_CARD\_NUMBER=%s" % a)  
            dat=cur.fetchone()  
            if x in dat:  
                d=int(input("Enter the Amount You Want To Withdraw:"))  
                if d<= 25000:  
                    cur.execute("select BALANCE FROM CUSTOMER WHERE DEBIT\_CARD\_NUMBER=%s" % a)  
                    bal=cur.fetchone()  
                    for z in bal:  
                        if (z-d)>=0:  
                            query=("update CUSTOMER set BALANCE=BALANCE-%s where DEBIT\_CARD\_NUMBER=% s")  
                            cur.execute(query,(d,a))  
                            con.commit()  
                            query=("select BALANCE from CUSTOMER where DEBIT\_CARD\_NUMBER=%s") % n  
                            cur.execute(query)  
                            data=cur.fetchone()  
                            for i in data:  
                                print('\*'\*57,'AMOUNT WITHDRAWN IS',d,'\*'\*57)  
                                print('\*'\*52,'YOUR CURRENT BALANCE IS',i,'\*'\*52)  
                                print('\*'\*48,'Transaction Complete! Please Remove Your Card!','\*'\*48)

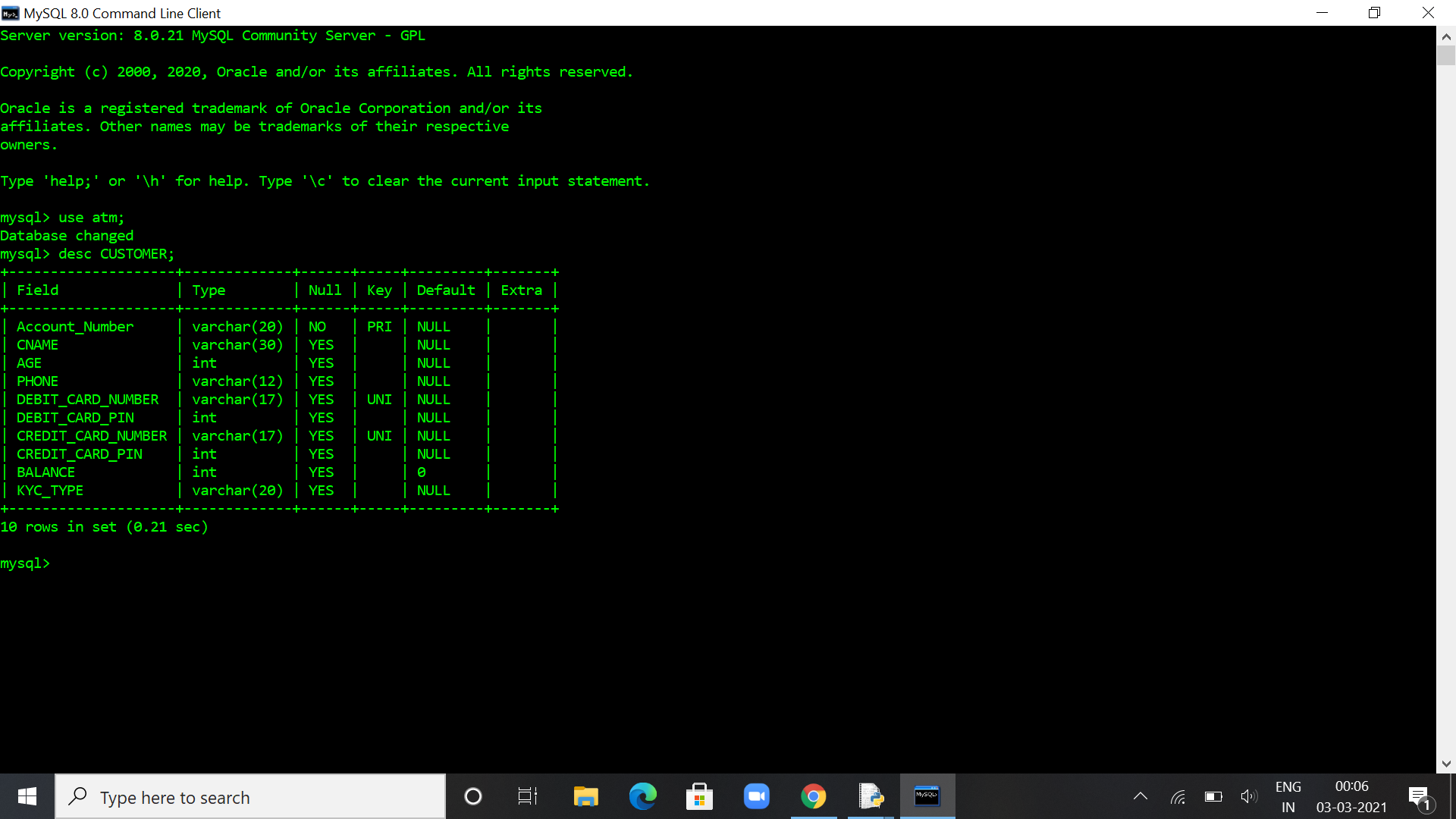
elif (z-d)<0:  
                            print('LOW BALANCE IN YOUR ACCOUNT!')  
                            print('CANNOT WITHDRAW THIS AMOUNT')  
  
                else:  
                    print('MAXIMUM WITHDRAWABLE AMOUNT PER TRANSACTION IS 25000')  
  
            else:  
                print("Transaction Failed! Incorrect PIN!!!")  
    else:  
        print('Transaction Failed! Details Not Matched!')

**#MODULE TO CHECK BALANCE**  
def check\_bal():  
    n=input("Enter The Debit Card Number:")  
    cur.execute("select DEBIT\_CARD\_NUMBER from CUSTOMER")  
    data=cur.fetchall()  
    n=(n,)  
    if n in data:  
        for a in n:  
            x=int(input("Enter Your PIN:"))  
            cur.execute("select DEBIT\_CARD\_PIN from CUSTOMER where DEBIT\_CARD\_NUMBER=%s" % a)  
            dat=cur.fetchone()  
            if x in dat:  
                cur.execute("select BALANCE from CUSTOMER where DEBIT\_CARD\_NUMBER=%s" % a)  
                data=cur.fetchone()  
                for i in data:  
                    print('\*'\*57,"Your Current Balance Is", i,'\*'\*57)  
  
            else:  
                print("Transaction Failed! Incorrect PIN!!!")  
    else:  
        print('Transaction Failed! Details Not Matched!')  
  
**#EXECUTION BLOCK**  
print('WELCOME TO SBI...')  
MySqlconnection()  
def execution():  
    print('(1)ADD NEW CUSTOMER')  
    print('(2)CREATE FIRST PIN')  
    print('(3)CHANGE EXISTING PIN')  
    print('(4)DEPOSIT MONEY')  
    print('(5)WITHDRAW MONEY')  
    print('(6)CHECK BALANCE')  
    z=int(input('ENTER THE SERIAL NUMBER OF THE FUNCTION YOU WANT TO PERFORM:'))  
    if z==1:  
        newcustomer()  
    elif z==2:  
        New\_PIN()  
    elif z==3:  
        CHANGE\_PIN()    
    elif z==4:  
        Deposit()  
    elif z==5:  
        withdraw()  
    elif z==6:  
        check\_bal()  
    else:  
        print('FUNCTION NOT FOUND')  
        e=input('DO YOU WANT TO RE-TRY ANY FUNCTION(Y/N):')  
        if e in "yY":  
            execution()  
        else:  
            pass  
    i=input("DO YOU WANT TO PERFORM ANY OTHER TASK(Y/N):")  
    if i in 'yY':  
        execution()  
    else:  
        cur.close()  
        print('THANK YOU FOR VISITING SBI ATM 😊!')  
     
execution()      
print("")  
print("")  
print('''ATM MANAGING PROGRAM  
DEVELOPED BY->UJJWAL JAIN  
CLASS-> XII-C  
SCHOOL->DEWAN PUBLIC SCHOOL''')

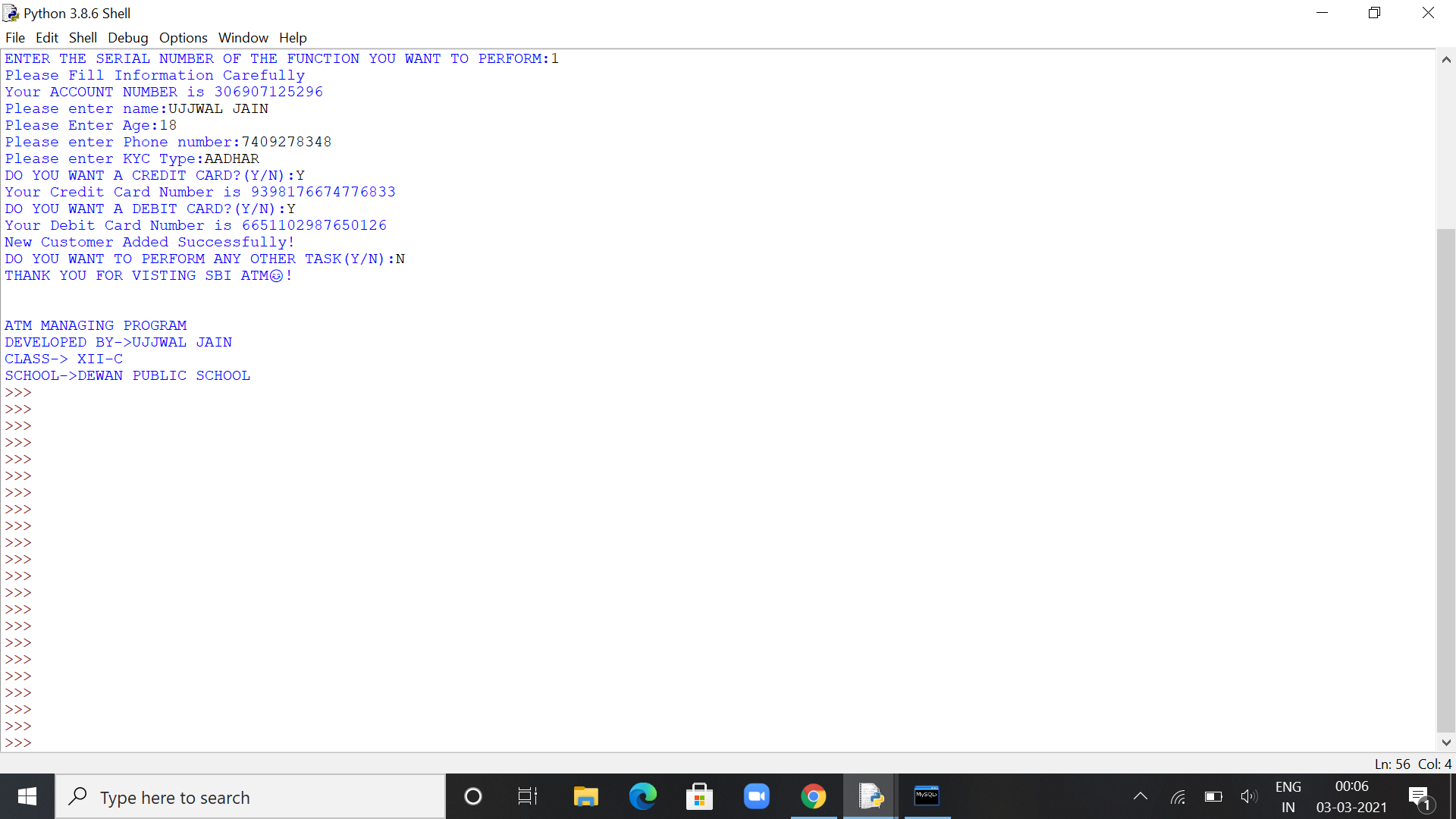
**9. OUTPUT SCREENS**

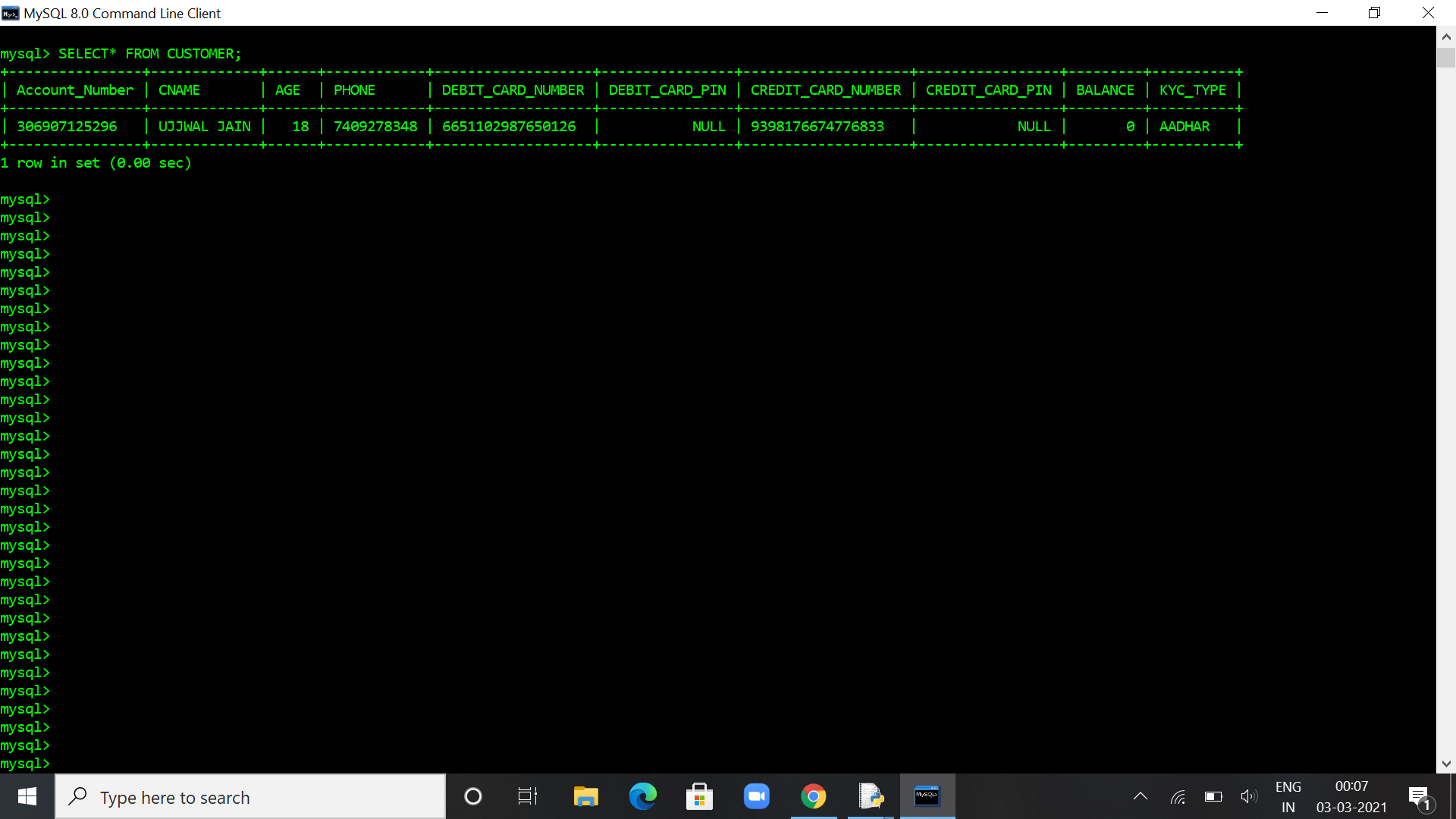
* Connecting to MySql



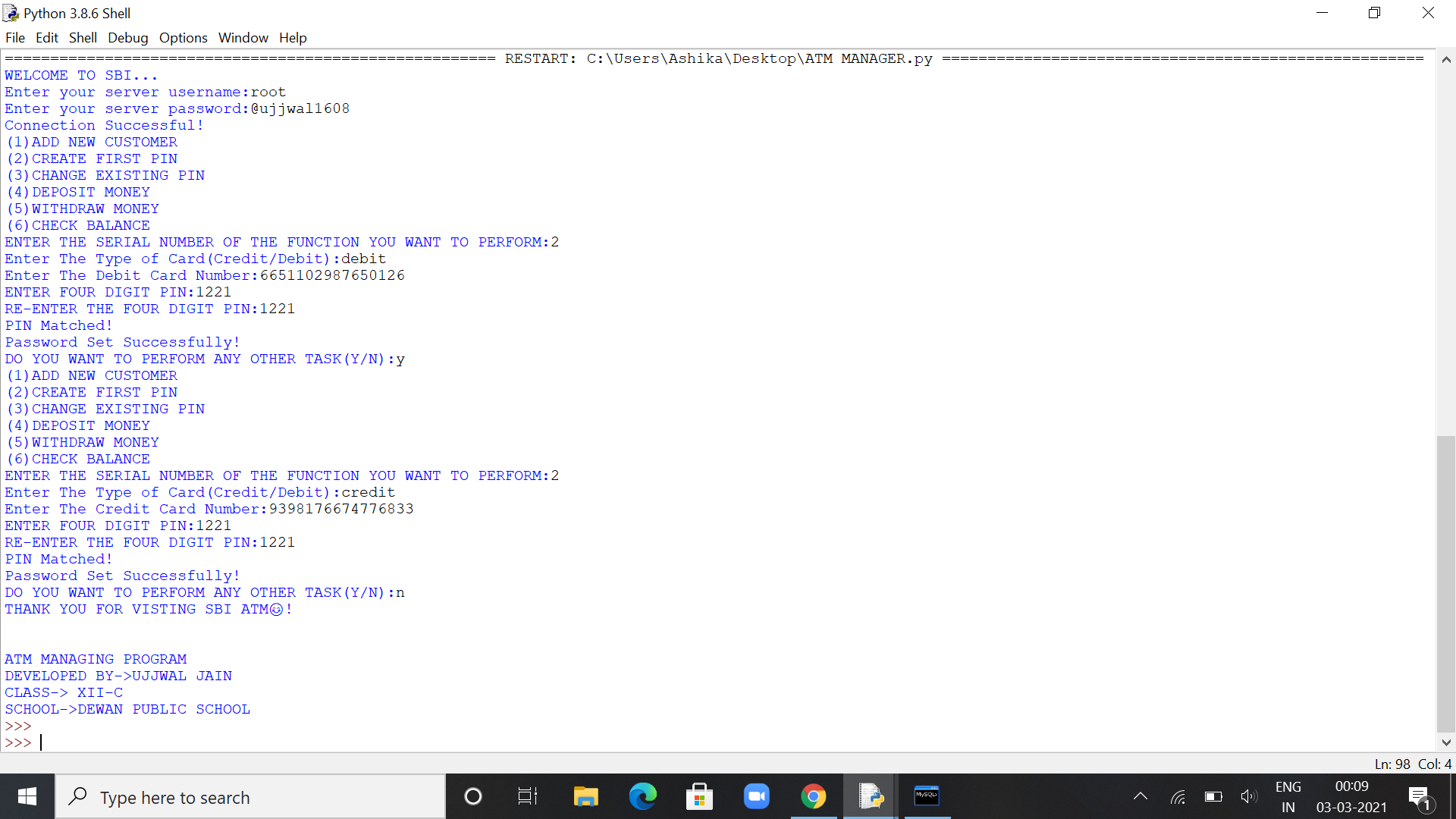
****

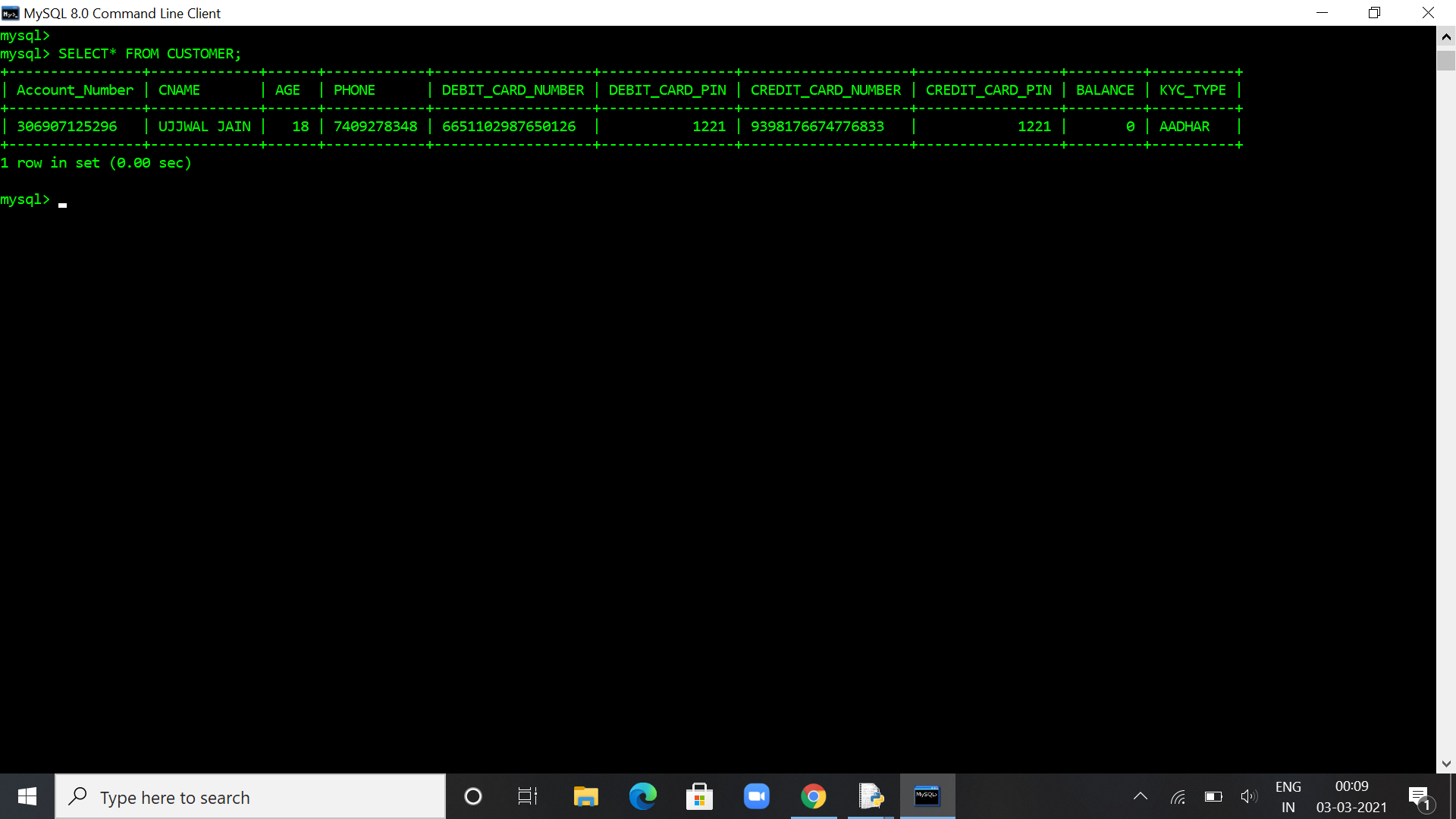
* First function

****

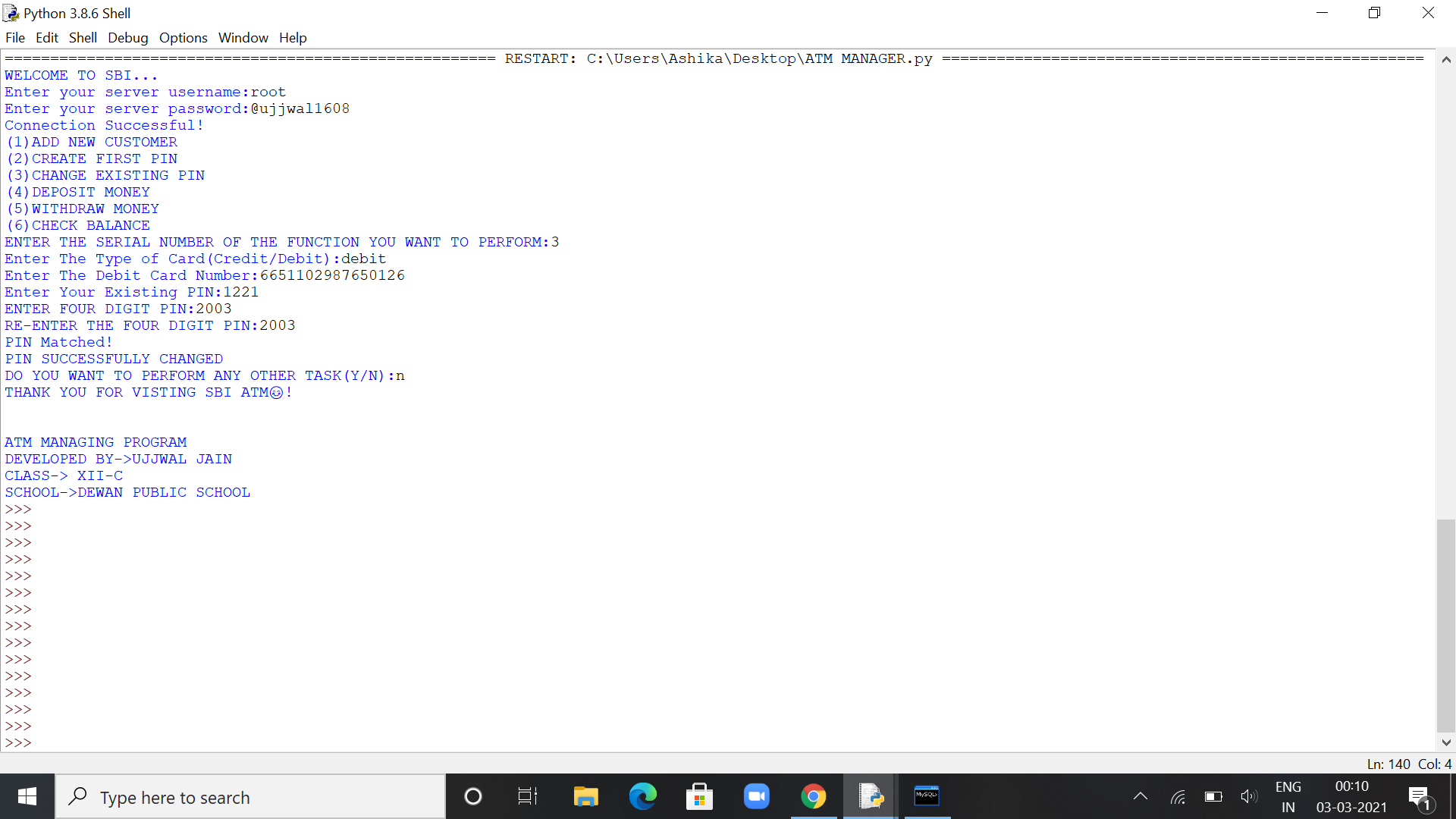
****

* Second function

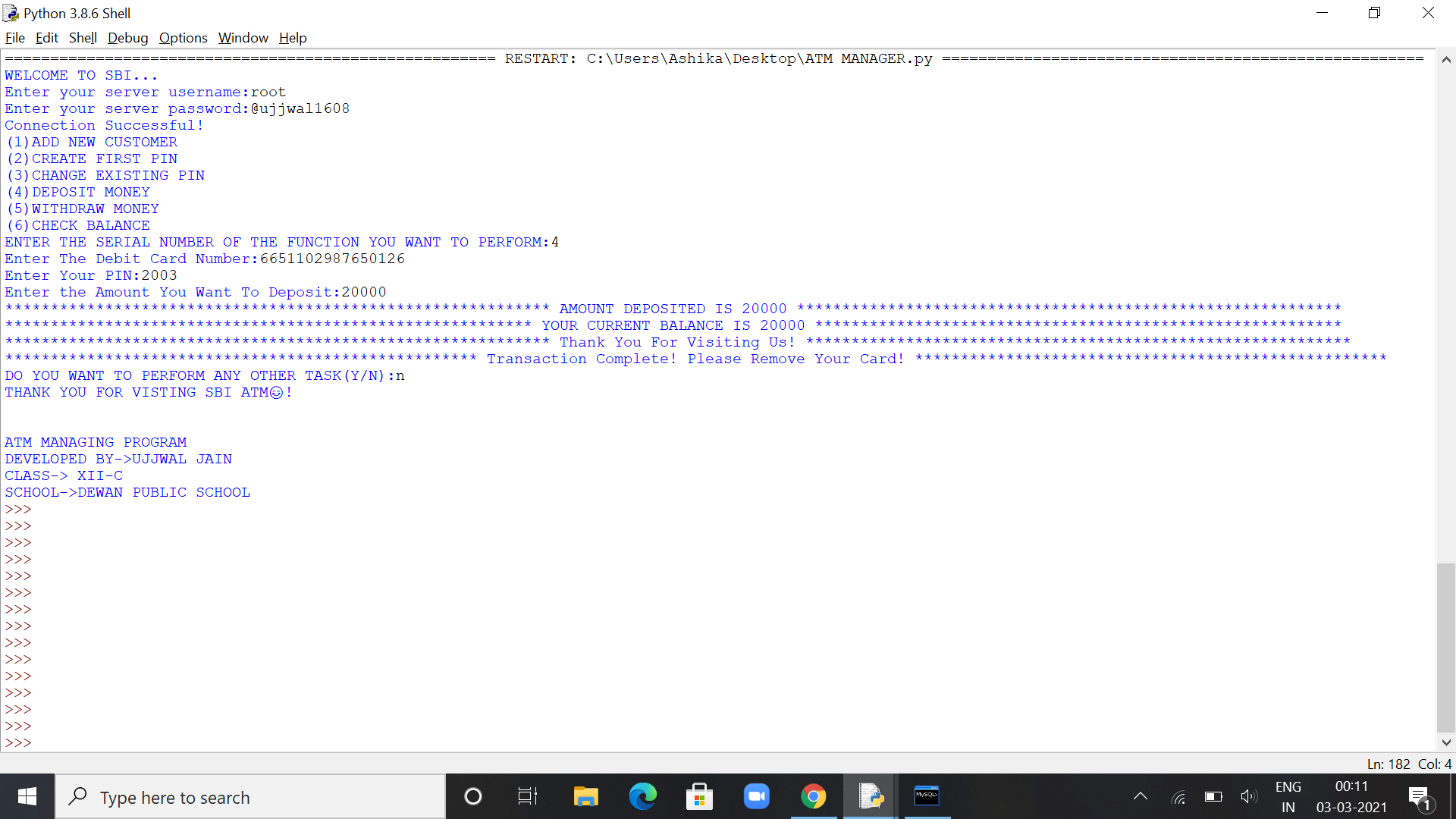
****

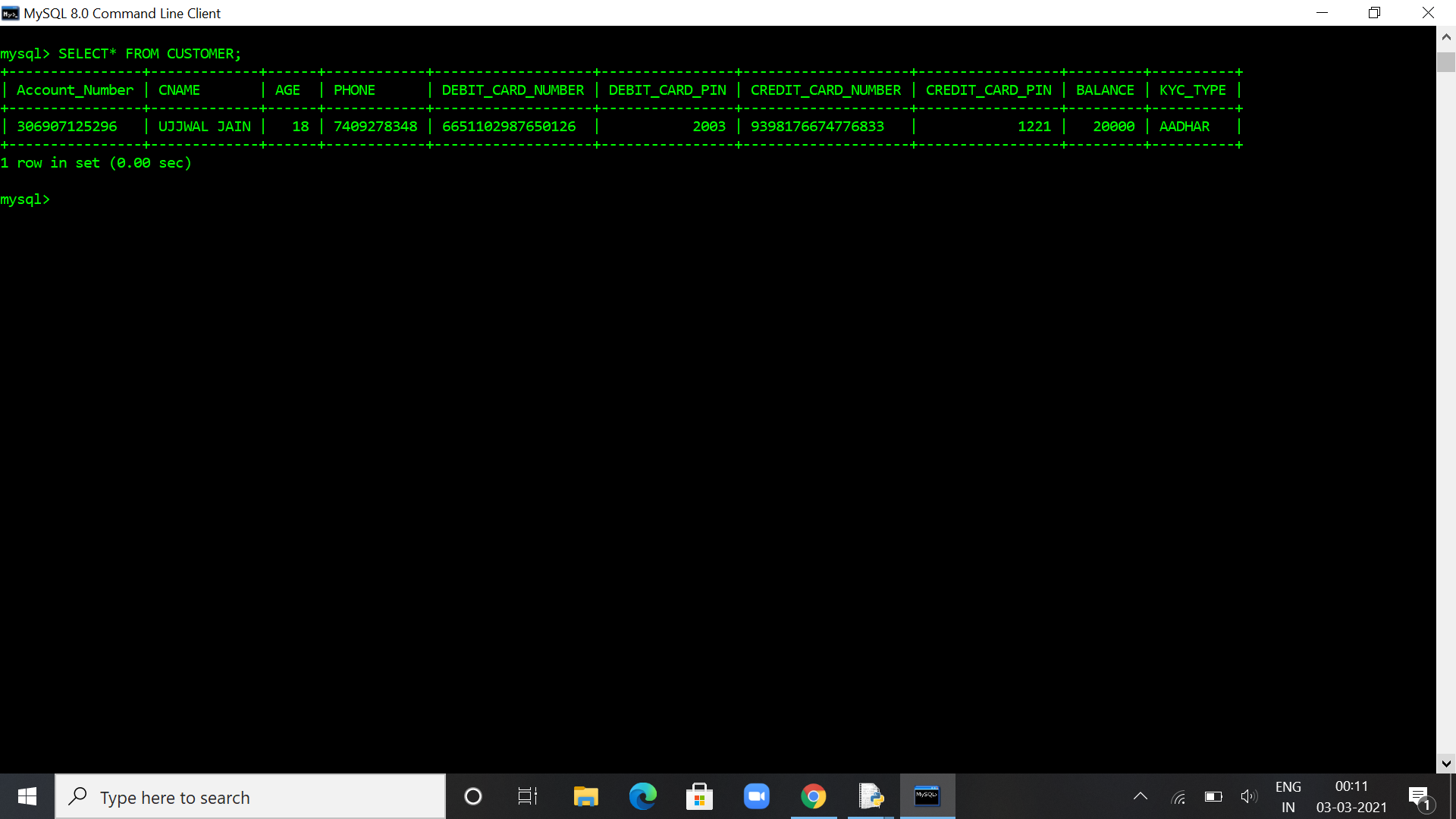
****

* Third function

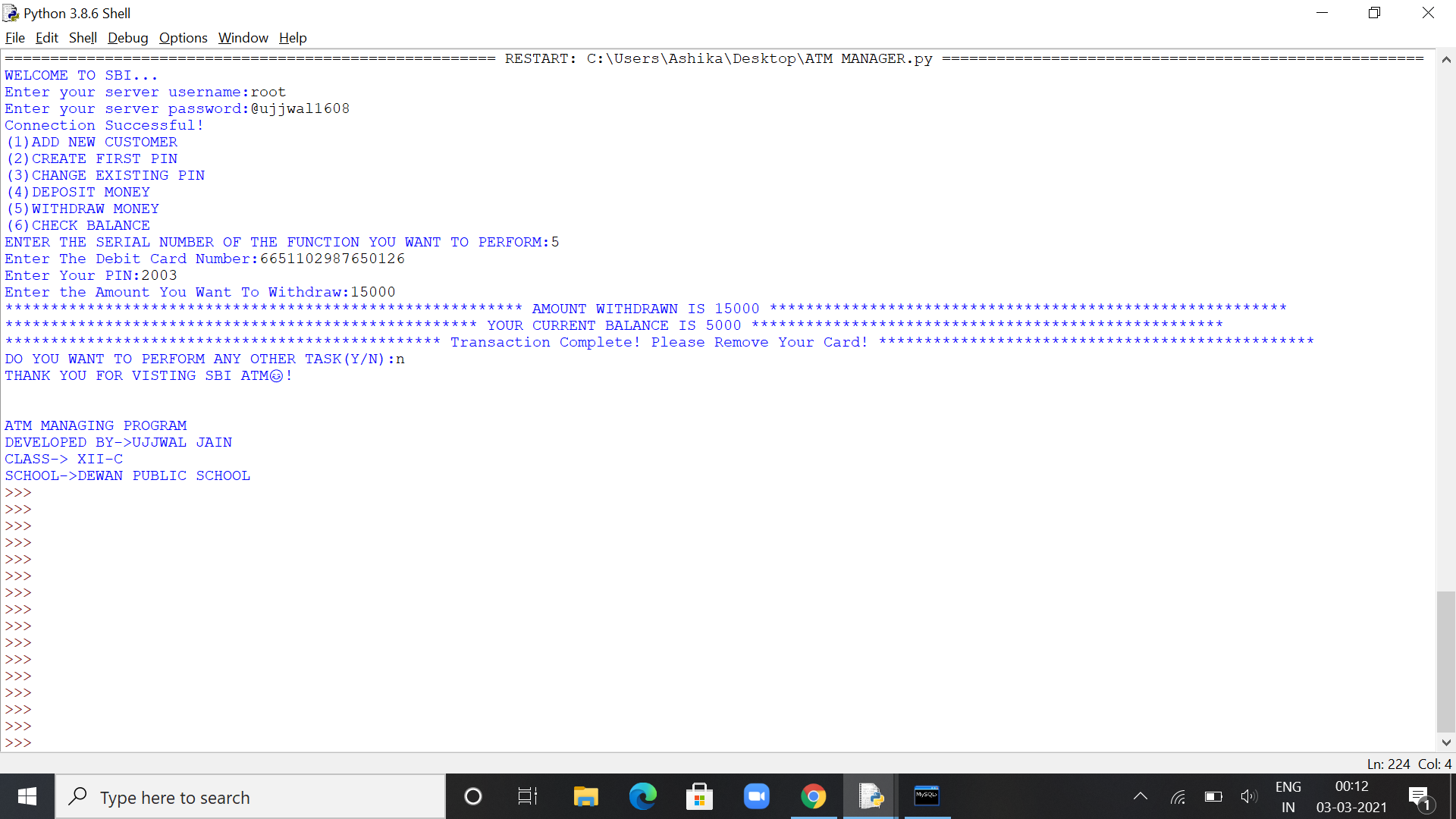


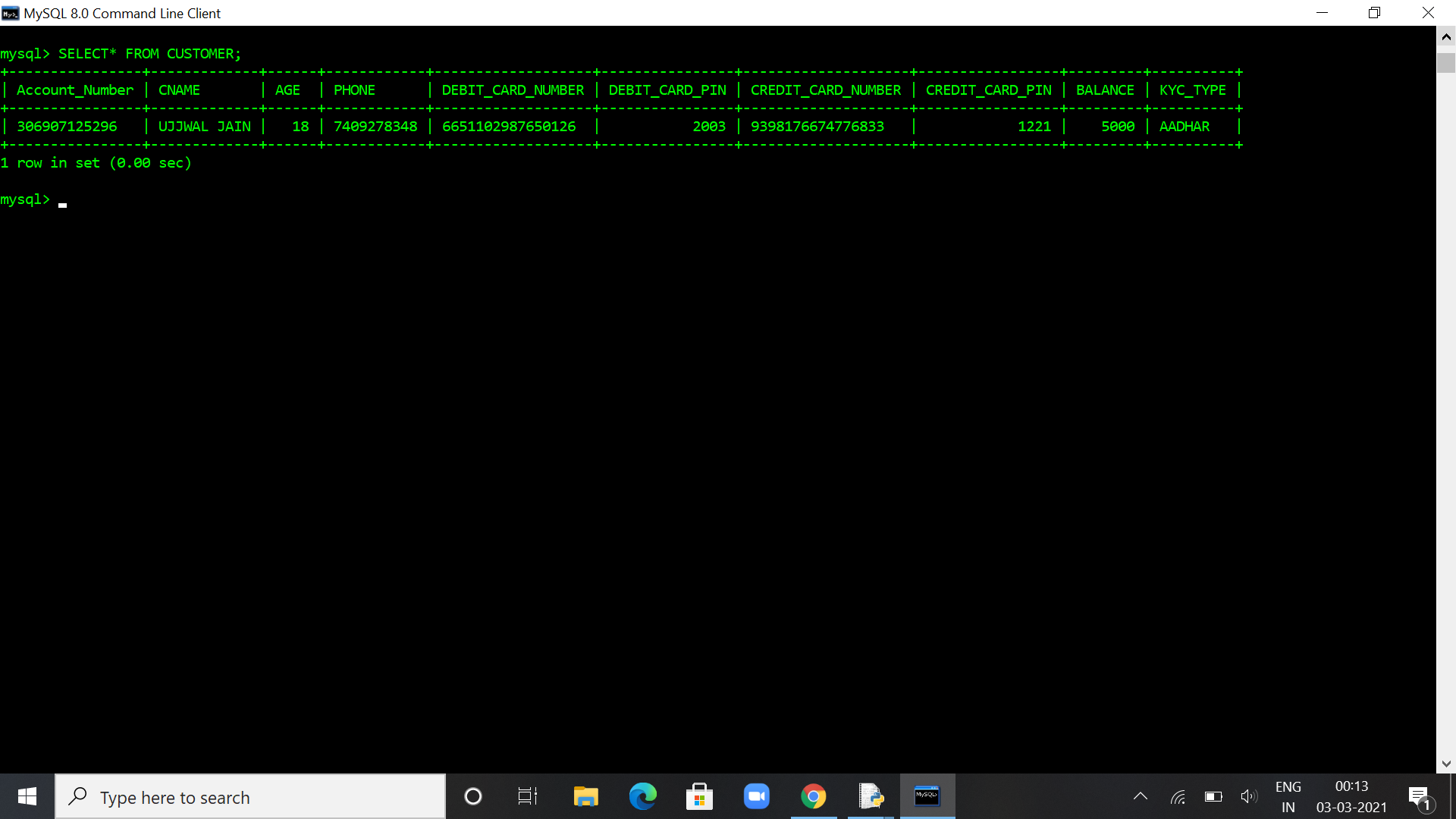
* Fourth function



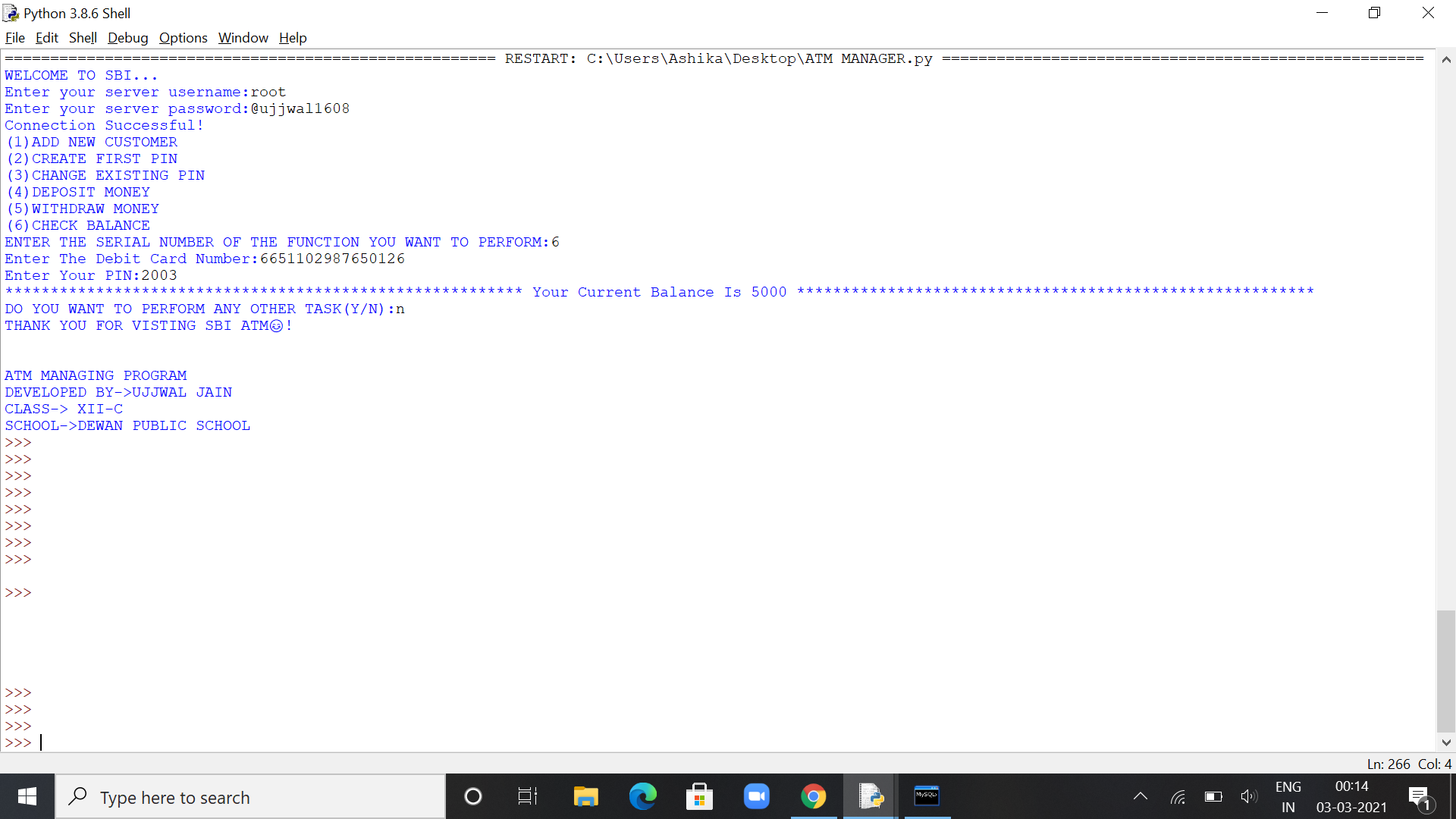


* Fifth function





* Sixth function

****

**10. future scope of PROJECT**

The future will see huge ATM software popularity, which will provide personalized features and a user friendly interface. This project will have ample of scope in ATM usage. A large number of people are moving towards using ATMs in place of conventionally banking which will be greatly helpful for the better future of this project. It can also be made using GUI for a better user experience.

**11. Bibliography**

* <https://www.youtube.com/>
* <https://www.w3schools.com/python/python_mysql_getstarted.asp>
* <https://dev.mysql.com/doc/connector-python/en/connector-python-examples.html>

**12. TEACHER’S REMARKS**

**13. Software Disk**