A Project Report On

"BANK MANAGEMENT"

Submitted by:

M . PRANAV SAI
Roll no: 98 Class: XII

Under the Guidance of

Mr. Anoop
PGT (Computer Science)
Department of Computer science.
SAINIK SCHOOL KALIKIRI

A Project Report On

"BANK MANAGEMENT"

Submitted by:

G . SANDEEP KUMAR
Roll no: 46 Class: XII

Under the Guidance of

Mr. Anoop
PGT (Computer Science)
Department of Computer science.
SAINIK SCHOOL KALIKIRI

A Project Report On

" BANK MANAGEMENT "

Submitted by:

B . PRUDHVI RAJ

Roll no. 60 Class: XII

Under the Guidance of

Mr. Anoop

PGT (Computer Science)

Department of Computer science.

SAINIK SCHOOL KALIKIRI

Department of Computer Science.
SAINIK SCHOOL KALIKIRI

CERTIFICATE

This is to certify that **CDT M.PRANAV SAI,** Roll No. 98 of Class XII has prepared the report on the Project titled **" BANK MANAGEMENT "**. The report is the result of his efforts & endeavors. The report is found worthy of acceptance as final project report for the subject Computer Science of Class XII. He has prepared the report under my guidance.

Mr. Anoop
PGT (Computer Science)
Department of Computer Science.
SAINIK SCHOOL KALIKIRI

Department of Computer Science.
SAINIK SCHOOL KALIKIRI

CERTIFICATE

This is to certify that **CDT. G .SANDEEP KUMAR ,** Roll No.46 of Class XII has prepared the report on the Project titled **"BANK MANAGEMENT"**. The report is the result of his efforts & endeavors. The report is found worthy of acceptance as final project report for the subject Computer Science of Class XII. He has prepared the report under my guidance.

Mr. Anoop
PGT (Computer Science
Department of Computer Science.
SAINIK SCHOOL KALIKIRI

Department of Computer Science.
SAINIK SCHOOL KALIKIRI

CERTIFICATE

This is to certify that **CDT.B.PRUDHVI RAJ ,** Roll No. 60 of Class XII has prepared the report on the Project entitled **“BANK MANAGEMENT”**. The report is the result of his efforts & endeavors. The report is found worthy of acceptance as final project report for the subject Computer Science of Class XII. He has prepared the report under my guidance.

Mr. Anoop
PGT (Computer Science)
Department of Computer Science.
SAINIK SCHOOL KALIKIRI

DECLARATION

I hereby declare that the project work titled "**BANK MANAGEMENT**", submitted to Department of **Computer Science**, SAINIK SCHOOL KALIKIRI is **prepared** by **me**. All the **coding** is the result of my **personal efforts.**

CDT : M. PRANAV SAI
Roll No: 98
Class : XII

SAINIK SCHOOL KALIKIRI

DECLARATION

I hereby declare that the project work entitled "**BANK MANAGEMENT**", submitted to Department of **Computer Science**, SAINIK SCHOOL KALIKIRI is **prepared** by **me**. All the **coding** is the result of my **personal efforts.**

CDT: G. SANDEEP KUMAR
Roll No: 46
Class : XII

DECLARATION

I hereby declare that the project work entitled “**BANK MANAGEMENT**”, submitted to Department of **Computer Science**, SAINIK SCHOOL KALIKIRI is **prepared** by **me**. All the **coding** is the result of my **personal efforts.**

CDT : B . PRUDHVI RAJ
Roll No: 60
Class : XII

ACKNOWLEDGEMENT

I would like to express a deep sense of thanks & gratitude to my **project guide Mr. Anoop** Sir for guiding me immensely through the course of the project. He always evinced keen interest in my work. His constructive advice & **constant motivation** have been responsible for the **successful** completion of this project.

My sincere thanks goes to **Capt.(IN). Vikranth kishore,** our **principal** sir, for his co-ordination in extending every **possible support** for the completion of this project.

I also thanks to my **parents** for their **motivation & support**. I must thanks to my **classmates** for their timely help & support for **compilation** of this **project**.

Last but not the least; I would like to thank all those who had helped directly or indirectly towards the completion of this project.

CDT : M . PRANAV SAI
Roll No: 98 , Class XII

ACKNOWLEDGEMENT

I would like to express a deep sense of thanks & gratitude to my **project guide Mr. Anoop** Sir for guiding me immensely through the course of the project. He always evinced keen interest in my work. His constructive advice & **constant motivation** have been responsible for the **successful** completion of this project.

My sincere thanks goes to **Capt.(IN). Vikranth kishore,** our **principal** sir, for his co-ordination in extending every **possible support** for the completion of this project.

I also thanks to my **parents** for their **motivation & support**. I must thanks to my **classmates** for their timely help & support for **compilation** of this **project**.

Last but not the least; I would like to thank all those who had helped directly or indirectly towards the completion of this project

CDT : G . SANDEEP KUMAR
Roll No: 46 , Class XII

ACKNOWLEDGEMENT

I would like to express a deep sense of thanks & gratitude to my **project guide Mr. Anoop** Sir for guiding me immensely through the course of the project. He always evinced keen interest in my work. His constructive advice & **constant motivation** have been responsible for the **successful** completion of this project.

My sincere thanks goes to **Capt.(IN). Vikranth kishore,** our **principal** sir, for his co-ordination in extending every **possible support** for the completion of this project.

I also thanks to my **parents** for their **motivation & support**. I must thanks to my **classmates** for their timely help & support for **compilation** of this **project**.

Last but not the least; I would like to thank all those who had helped directly or indirectly towards the completion of this project.

CDT : B. PRUDHVI RAJ
Roll No: 60 , Class XII

CONTENTS

PREREQUISITIVES

- In order to execute the program you need to have the below mentioned modules installed on your system in addition to standard installation packages of python , MYSQL:

    1. gTTS
    2. sleep
    3. stdiomask
    4. pyglet
    5. datetime
    6. pytz
    7. mysql.connector

other module needs are a part of the standard python package.

- In order to have all the tables created on your MYSQL database which the program refers to during its execution , run the python file named “table creation.py”.

- You should have an active internet connection during the execution of the program for the gTTS module to function.

Working of the program

- You can use the program as an user as well as an employee.
- As an user you will be able to access the following functionalities :
    1. You'll be able to create a new account.
    2. If you're a pre-existing user you can :
        - Access your account info.
        - Transfer money to another Account holder of our bank.
        - View your transaction details.
        - Deposit or withdraw your money.
        - Update your account details.
        - Apply for a loan.

- Delete your account.

- If you run the program as an employee, after verifying your credentials you’ll get many admin privileges.
- As an employee you can:
    - Check user details and transactions.
    - Modify loan rates.
    - Sanction loans.

CODE OF THE PROJECT

```
#importing required modules and connecting to mysql
from gtts import gTTS
from time import sleep
import stdiomask
import os
import datetime
import pytz
import random
import pyglet
import mysql.connector as con

#accessing todays date
tdt = datetime.datetime.now(pytz.timezone('Asia/Kolkata'))
tdts = str(tdt)
d = tdts[0:11]

#connecting with mysql database
mycon = con.connect(host = 'localhost', user = 'root', password = 'password', database = 'rough')
cur = mycon.cursor()
```

```
#fetching list of usernames,passwords,names and account numbers
lu = []
lp = []
ln = []
lan = []
cur.execute('select * from user_password')
x = cur.fetchall()
for i in x:
    lu.append(i[0])
    lp.append(i[1])
    ln.append(i[2])
    lan.append(i[3])

#fetching balance of user with respect to their account number
lub = {}
cur.execute('select * from user_balance')
x = cur.fetchall()
for i in x:
    lub[i[0]] = i[1]

#fetching employee usernames,passwords and names
elu = []
elp = []
```

```
eln = []
cur.execute('select * from employee_password')
x = cur.fetchall()
for i in x:
    elu.append(i[0])
    elp.append(i[1])
    eln.append(i[2])

#fetching loan codes,rates and names
llc = []
llr = []
lln = []
cur.execute('select * from loan_code')
x = cur.fetchall()
for i in x:
    llc.append(i[0])
    llr.append(i[1])
    lln.append(i[2])

#creating a dictionary of rates with respect to loan name
lrln = {}
for i in range(4):
    lrln[lln[i]] = llr[i]
```

```
#fetching the account numbers applied for loan
lal = []
cur.execute('select account_num from loan_details')
x = cur.fetchall()
for i in x:
    i = list(i)
    lal.append(i[0])

#fetching account numbers whose loan status is pending
lpl = []
cur.execute("select account_num from loan_details where status = 'pending'")
x = cur.fetchall()
for i in x:
    i = list(i)
    lpl.append(i[0])


#fetching list of phone numbers and aadhar card numbers
lacn = []
lpn = []
cur.execute('select phone_num,aadhar_num from user_details')
```

```
x = cur.fetchall()
for i in x:
    lacn.append(i[1])
    lpn.append(i[0])



#creating a function for text to speech
def speech(x):
    tts = gTTS(text = x, lang = 'en')
    filename = 'welcome.mp3'
    tts.save(filename)
    music = pyglet.media.load(filename, streaming=False)
    music.play()
    sleep(music.duration)
    os.remove(filename)



#creating a function for returning only block letter string
def block(x):
    y = ''
    for i in x:
        y += i.capitalize()
    return y
```

```
#checking whether the connection to mysql is successfull
if mycon.is_connected():
    print('                              -\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\ /-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-')
    print('                              <-<-<-<-<-<-<-<-<-<-<-<-<-<-<-<-<-<-<-<-< THE ROYAL BANK OF INDIA >->->->->->->->->->->->->->->->->->->->->->->')
    print('                              -/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/ \-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-')
    print()
    print()
    #welcome note
    speech('WELCOME TO THE ROYAL BANK OF INDIA.')
    while(True):
        #asking to select the type of user
        print('1. USER')
        print('2. EMPLOYEE')
        print('3. EXIT')
        print()
        c1 = int(input('ENTER YOUR CHOICE: '))
        print()
        print()
        #if he/she is a user
        if c1 == 1:
            while(True):
                print('1. LOG IN')
                print('2. DON\'T HAVE AN ACCOUNT, CREATE NEW ACCOUNT')
                print('3. BACK')
                print()
                c0 = int(input('ENTER YOUR CHOICE: '))
                print()
                print()
                if c0 == 1:
                    while(True):
                        #asking the user to enter his/her username or b to go back
                        u = input('ENTER USERNAME, OR ENTER B GO BACK: ')
                        print()
                        #if the user enters b
                        if u == 'B' or u == 'b':
                            break
                        #checking whether the entered username is valid
                        if u in lu:
                            n = lu.index(u)
                            an = lan[n]
                            #asking the user to enter his/her password or b to go back
                            p = stdiomask.getpass(prompt = 'ENTER PASSWORD, OR ENTER B GO BACK: ')

                            print()

                            print()

                            #if the user enters b

                            if p == 'B' or p == 'b':

                                break

                            #checking whether the password is valid

                            if p == lp[n]:

                                #welcome note of the user

                                speech(f'WELCOME {ln[n]}')

                                print('WELCOME,',ln[n])

                                #giving a warning to the user if his/her balance is 0

                                if lub[an] == 0:

                                    speech('WARNING! . YOUR CURRENT BANK BALANCE IS 0. PLEASE DEPOSIT MINIMUM REQUIRED BALANCE TO USE ALL OF OUR SERVICES')

                                    print('WARNING! , YOUR CURRENT BANK BALANCE IS 0, PLEASE DEPOSIT MINIMUM REQUIRED BALANCE TO USE ALL OF OUR SERVICES.')

                                print()

                                while(True):

                                    print('1. ACCOUNT INFO')

                                    print('2. DEPOSIT MONEY')

                                    print('3. WITHDRAW MONEY')
```

```
print('4. VIEW TRANSACTIONS')
print('5. TRANSFER MONEY')
print('6. UPDATE ACCOUNT DETAILS')
print('7. APPLY LOAN')
print('8. DELETE MY ACCOUNT')
print('9. BACK')
print()
c2 = int(input('ENTER YOUR CHOICE: '))
print()
print()
#if the user chooses to view their details
if c2 == 1:
    while(True):
        tl = []
        #collecting the details from mysql
        st = f'select * from user_details where account_num = {an}'
        cur.execute(st)
        x = cur.fetchall()
        for i in x:
            for j in i:
                tl.append(j)
        tl.append(lub[an])
```

```
                    print('ACCOUNT NUMBER | USERNAME     | NAME               | DATE OPENED | AADHAR NUMBER | PAN CARD ID | PHONE NUMBER | BALANCE | ADDRESS',' ' * (len(str(tl[7])) - 8),'|')
                    print('-' * (132 + ((len(str(tl[7])) - 6))))
                    print(tl[0],'     |',tl[1],'  |',tl[2],' ' * (19 - len(str(tl[2]))),'|',tl[3],' |',tl[4],' |',tl[5],' |',tl[6],'  |',tl[-1],' ' * (6 - len(str(tl[-1]))),'|',tl[7],'|')
                    print()
                    b = input('ENTER ANY KEY TO GO BACK: ')
                    print()
                    print()
                    if b == 'b':
                        break
                    else:
                        break
            #if the user wants to deposit money
            elif c2 == 2:
                while(True):
                    a = int(input('ENTER THE AMOUNT: '))
                    print()
                    #updating the users balance
                    st = f'update user_balance set balance = balance + {a} where account_num = {an}'
                    cur.execute(st)
                    mycon.commit()
                    lub[an] += a
                    #giving a conformation to the user
                    speech(f'THE  AMOUNT OF RUPEES {a} HAS BEEN SUCCESSFULLY DEPOSITED TO YOUR ACCOUNT.')
                    print('THE  AMOUNT HAS BEEN SUCCESSFULLY DEPOSITED TO YOUR ACCOUNT.')
                    print()
                    #updating the transactions table
                    tdt = datetime.datetime.now(pytz.timezone('Asia/Kolkata'))
                    tdts = str(tdt)
                    t = tdts[11:19]
                    st = f"insert into {an}_trans values('{d}','{t}',{a},'CREDITED TO YOURSELF')"
                    cur.execute(st)
                    mycon.commit()
                    b = input('ENTER N TO DEPOSIT AGAIN, OR ANY KEY TO GO BACK: ')
                    print()
                    print()
                    if b == 'N' or b == 'n':
                        continue
                    else:
                        break
            #if the user wants to withdraw money
```

```
elif c2 == 3:
    while(True):
        a = int(input('ENTER THE AMOUNT: '))
        print()
        #checking whether the entered amount is available in the users account
        if a <= lub[an]:
            #updating the users balance
            st = f'update user_balance set balance = balance - {a} where account_num = {an}'
            cur.execute(st)
            mycon.commit()
            lub[an] -= a
            #giving a conformation to the user
            speech(f'THE  AMOUNT OF RUPEES {a} HAS BEEN SUCCESSFULLY DEBITED FROM YOUR ACCOUNT')
            print('THE  AMOUNT HAS BEEN SUCCESSFULLY WITHDRAWED FROM YOUR ACCOUNT.')
            print()
            #updating the transactions table
            tdt = datetime.datetime.now(pytz.timezone('Asia/Kolkata'))
            tdts = str(tdt)
            t = tdts[11:19]
```

```
                     st = f"insert into {an}_trans values('{d}','{t}',{a},'DEBITED BY YOURSELF')"
                     cur.execute(st)
                     mycon.commit()
                  else:
                     #if the entered amount is greater than the balance intimating it to the user
                     speech('THE AMOUNT OF RUPEES {a} IS GREATER THAN YOUR BALANCE. PLEASE ENTER AN APPROPRIATE AMOUNT')
                     print('THE AMOUNT ENTERED IS GREATER THAN YOUR BALANCE, PLEASE ENTER AN APPROPRIATE AMOUNT.')
                     print()
                     continue
                  b = input('ENTER N TO WITHDRAW AGAIN, OR ANY KEY TO GO BACK: ')
                  print()
                  print()
                  if b == 'N' or b == 'n':
                     continue
                  else:
                     break
            #if the user wants to check his/her transactions
            elif c2 == 4:
              while(True):
                  #printing all the transactions made by the user
                  print('DATE(Y-M-D) | TIME(H-M-S) | AMOUNT    | DESCRIPTION                ')
                  print('--------------------------------------------------------------------------------')
                  st = f'select * from {an}_trans'
                  cur.execute(st)
                  x = cur.fetchall()
                  for i in x:
                     tl = []
                     for j in i:
                         tl.append(j)
                     print(tl[0],' |',tl[1],'   |',tl[2],' ' * (8 - len(str(tl[2]))),'|',tl[3])
                  print()
                  speech('THESE ARE THE TRANSACTIONS YOU HAVE MADE THROUGH OUR BANK')
                  b = input('ENTER ANY KEY TO GO BACK: ')
                  print()
                  print()
                  if b == 'b':
                     break
                  else:
                     break
            #if the user wants to transfer money to another user
```

```
elif c2 == 5:
    while(True):
        tua = int(input('ENTER THE ACCOUNT NUMBER OF THE PERSON YOU WANT TO TRANSFER: '))
        print()
        #checking whether the entered account number is valid
        if tua in lan:
            a = int(input('ENTER THE AMOUNT: '))
            print()
            #checking whether the entered amount is available in the user account
            if a <= lub[an]:
                #adding the amount to the specified account number
                st = f'update user_balance set balance = balance + {a} where account_num = {tua}'
                cur.execute(st)
                mycon.commit()
                #deducting the amount from the user account
                st = f'update user_balance set balance = balance - {a} where account_num = {an}'
                cur.execute(st)
                mycon.commit()
                lub[an] -= a
                tn = lan.index(tua)
                tun = ln[tn]
                #giving the user a conformation
                speech(f'THE AMOUNT ENTERED HAS SUCCESSFULLY TRANSFERED TO THE ACCOUNT OF {tun}')
                print('THE AMOUNT ENTERED HAS BEEN SUCCESSFULLY TRANSFERED.')
                print()
                #updating the user transactions table
                tdt = datetime.datetime.now(pytz.timezone('Asia/Kolkata'))
                tdts = str(tdt)
                t = tdts[11:19]
                st = f"insert into {an}_trans values('{d}','{t}',{a},'TRANSFERED FROM YOUR ACCOUNT TO {tun}')"
                cur.execute(st)
                mycon.commit()
                st = f"insert into {tua}_trans values('{d}','{t}',{a},'TRANSFERED TO YOUR ACCOUNT BY {ln[n]}')"
                cur.execute(st)
                mycon.commit()
                #giving a warning to the user if the balance is 0
                if lub[an] == 0:
                    speech('WARNING!. YOUR CURRENT BANK BALANCE IS 0. PLEASE DEPOSIT MINIMUM REQUIRED BALANCE TO FURTHER USE OUR SERVICES')
```

```
            print('WARNING!, YOUR CURRENT BANK BALANCE IS 0, PLEASE DEPOSIT MINIMUM REQUIRED BALANCE TO FURTHER USE OUR SERVICES.')
            print()
    else:
        #intimating the user if the entered amount is greater than the balance
        speech('THE AMOUNT ENTERED IS GREATER THAN YOUR BALANCE. PLEASE ENTER AN APPROPRIATE AMOUNT')
        print('THE AMOUNT ENTERED IS GREATER THAN YOUR BALANCE, PLEASE ENTER AN APPROPRIATE AMOUNT.')
        print()
        continue
else:
    #intimating the user if the account number does not exist
    speech('THIS ACCOUNT DOES NOT EXIST. PLEASE ENTER AN VALID ACCOUNT NUMBER')
    print('THIS ACCOUNT DOES NOT EXIST, PLEASE ENTER AN VALID ACCOUNT NUMBER.')
    print()
    continue
b = input('ENTER N TO TRANSFER AGAIN, OR ANY KEY TO GO BACK: ')
print()
```

```
                        print()
                        if b == 'N' or b == 'n':
                            continue
                        else:
                            break
                #if the user wants to update his/her details
                elif c2 == 6:
                    while(True):
                        print('PHONE NUMBER -- P')
                        print('ADDRESS      -- A')
                        print('PASSWORD     -- S')
                        print('USERNAME     -- U')
                        print()
                        f = input('ENTER THE FIELD CODE AS MENTIONED ABOVE, THAT YOU WANT TO UPDATE: ')
                        print()
                        #if the user wants to update his/her address
                        if f == 'a' or f == 'A':
                            ua = input('ENTER YOUR NEW ADDRESS: ')
                            print()
                            #updating the address in mysql
                            st = f"update user_details set address = '{ua}' where account_num = {an}"
                            cur.execute(st)
                            mycon.commit()
                            speech('YOUR ADDRESS HAS BEEN SUCCESSFULLY UPDATED')
                            print('YOUR ADDRESS HAS BEEN SUCCESSFULLY UPDATED.')
                            print()
                        #if the user wants to update his/her phone number
                        elif f == 'P' or f == 'p':
                            up = input('ENTER YOUR NEW PHONE NUMBER: ')
                            print()
                            #updating the phone number in mysql
                            st = f"update user_details set phone_num = '{up}' where account_num = {an}"
                            cur.execute(st)
                            mycon.commit()
                            speech('YOUR PHONE NUMBER HAS BEEN SUCCESSFULLY UPDATED')
                            print('YOUR PHONE NUMBER HAS BEEN SUCCESSFULLY UPDATED.')
                            print()
                        #if the user wants to change his/her password
                        elif f == 'S' or f == 's':
                            #asking the user to enter the current password
                            tp = stdiomask.getpass(prompt = 'ENTER YOUR CURRENT PASSWORD: ')
```

```
print()
if tp == p:
    #asking the user to create new password
    p1 = stdiomask.getpass(prompt = 'ENTER YOUR NEW PASSWORD: ')
    print()
    p2 = stdiomask.getpass(prompt = 'REENTER YOUR NEW PASSWORD: ')
    print()
    #checking whether the user has entered the passwords correctly
    if p1 == p2:
        #updating the password in mysql
        st = f"update user_password set password = '{p1}' where account_num = {an}"
        cur.execute(st)
        mycon.commit()
        speech('YOUR PASSWORD HAS BEEN SUCCESSFULLY UPDATED')
        print('YOUR PASSWORD HAS BEEN SUCCESSFULLY UPDATED.')
        print()
    #if the entered passwords do not match
    else:
        speech('THE ENTERED PASSWORDS DO NOT MATCH')
```

```
                                                        print('THE ENTERED PASSWORDS DO
NOT MATCH.')
                                                        print()
                                                #if the user enters incorrect password
                                                else:
                                                    speech('INCORRECT PASSWORD')
                                                    print('INCORRECT PASSWORD.')
                                                    print()
                                            #if the user wants to change his/her username
                                            elif f == 'U' or f == 'u':
                                                while(True):
                                                    print('THE USERNAME MUST CONTAIN
ONLY 10 CHARACTERS.')
                                                    print()
                                                    nu = input('ENTER YOUR NEW
USERNAME: ')
                                                    print()
                                                    #checking whether the entered username is
valid or not
                                                    if nu not in lu and len(nu) == 10:
                                                        #updating the username in mysql
                                                        st = f"update user_details set username =
'{nu}' where account_num = {an}"
                                                        cur.execute(st)
                                                        mycon.commit()
```

```
                        st = f"update user_password set username =
'{nu}' where account_num = {an}"
                        cur.execute(st)
                        mycon.commit()
                        speech('YOUR USERNAME HAS BEEN
SUCCESSFULLY UPDATED')
                        print('YOUR USERNAME HAS BEEN
SUCCESSFULLY UPDATED.')
                        break
                    #if the entered username is invalid
                    else:
                        speech('THE ENTERED USERNAME
ALREADY EXISTS. or THE ENTERED USERNAME IS INVALID.
PLEASE ENTER A VALID USERNAME')
                        print('THE ENTERED USERNAME
ALREADY EXISTS, OR THE ENTERED USERNAME IS INVALID.
PLEASE ENTER A VALID USERNAME.')
                        print()
                        continue
            else:
                speech('THE ENTERED FIELD CODE IS
INVALID. PLEASE ENTER A VALID FIELD CODE')
                print('THE ENTERED FIELD CODE IS
INVALID, PLEASE ENTER A VALID FIELD CODE.')
            b = input('ENTER N TO FURTHER CHANGE,
OR ENTER ANY KEY TO GO BACK: ')
```

```
        print()
        print()
        if b == 'N' or b == 'n':
            continue
        else:
            break
#if the user wants to aplly for a loan
elif c2 == 7:
    while(True):
        if an not in lal:
            speech('THESE ARE THE TYPES OF LOANS OFFERED BY OUR BANK')
            print('THESE ARE THE TYPE OF LOANS OFFERED BY OUR BANK:')
            print()
            print('| LOAN NAME      | LOAN RATE | LOAN CODE |')
            print('-----------------------------------------')
            print('| BUSSINESS LOAN | 15 %      | bs15 |')
            print('| HOME LOAN      | 10 %      | hm07 |')
            print('| PERSONAL  LOAN | 11 %      | pl11 |')
            print('| VEHICLE LOAN   | 09 %      | vc09 |')
```

```
                    print()
                    s = input('ENTER THE LOAN CODE OF THE LOAN YOU WANT TO APPLY: ')
                    tln = llc.index(s)
                    print()
                    a = int(input('ENTER THE AMOUNT: '))
                    print()
                    t = int(input('ENTER THE TIME PERIOD: '))
                    print()
                    st = f"insert into loan_details values({an},'{ln[n]}',{a},{t},'{lln[tln]}','PENDING')"
                    cur.execute(st)
                    mycon.commit()
                    lal.append(an)
                    speech('YOUR LOAN REQUEST HAS BEEN SUCCESSFULLY UPDATED. WE WILL VERIFY AND SANCTION YOUR LOAN')
                    print('YOUR LOAN REQUEST HAS BEEN SUCCESSFULLY UPDATED, WE WILL VERIFY AND SANCTION YOUR LOAN.')
                    print()
                else:
                    speech('YOU HAVE ALREADY APPLIED FOR A LOAN. YOU CAN NOT APPLY FOR ANOTHER LOAN UNTIL THE CURRENT LOAN PERIOD is OVER')
```

```
            print('YOU HAVE ALREADY APPLIED FOR A LOAN, YOU CAN NOT APPLY FOR ANOTHER LOAN UNTIL THE CURRENT LOAN PERIOD IS OVER.')
            print()
        b = input('ENTER ANY KEY TO GO BACK: ')
        print()
        print()
        if b == 'b':
            break
        else:
            break
#if the user wants to delete his/her account
elif c2 == 8:
    if an in lal:
        speech('YOU HAVE CURRENTLY APPLIED FOR A LOAN. YOU CANNOT DELETE YOUR ACCOUNT NOW')
        print('YOU HAVE CURRENTLY APPLIED FOR A LOAN, YOU CANNOT DELETE YOUR ACCOUNT NOW.')
        print()
    else:
        #confirming from the user whether he/she wants to delete their account
        speech('ARE YOU SURE. DO YOU WANT TO DELETE YOUR ACCOUNT')
        l = block(input('ENTER Y TO DELETE YOUR ACCOUNT OR N TO GO BACK: '))
```

```
                        print()
                        #if the user wants to delete their account
                        if l == 'Y':
                            #asking the user to enter his/her password to
furhter proceed to deletion of the account
                            speech('TO CONFIRM DELETION OF THE
ACCOUNT ENTER YOUR PASSWORD')
                            print('TO CONFIRM DELETION OF THE
ACCOUNT ENTER YOUR PASSWORD.')
                            print()
                            tp = stdiomask.getpass(prompt = 'ENTER
YOUR PASSWORD: ')
                            print()
                            #if the entered password is correct
                            if tp == p:
                                #deleting all the users records and details from
mysql
                                st = f'delete from user_details where
account_num = {an}'
                                cur.execute(st)
                                mycon.commit()
                                st = f'delete from user_password where
account_num = {an}'
                                cur.execute(st)
                                mycon.commit()
```

```
                    st = f'delete from user_balance where account_num = {an}'
                    cur.execute(st)
                    mycon.commit()
                    st = f'drop table {an}_trans'
                    cur.execute(st)
                    mycon.commit()
                    speech('SORRY FOR YOUR INCONVENIENCE. HOPE YOU WILL JOIN us AGAIN')
                    print('SORRY FOR YOUR INCONVENIENCE, HOPE YOU WILL JOIN US AGAIN.')
                    print()
                    speech('YOUR ACCOUNT HAS BEEN SUCCESSFULLY DELETED')
                    print('YOUR ACCOUNT HAS BEEN SUCCESSFULLY DELETED.')
                    print()
                #if the user enters incorrect password
                else:
                    speech('INCORRECT PASSWORD')
                    print('INCORRECT PASSWORD.')
                    print()
            if l == 'N':
                print()
            else:
```

```
                                print()
                        b = input('ENTER ANY KEY TO GO BACK: ')
                        print()
                        print()
                        if b == 'b':
                            break
                        else:
                            break
                    #if the user wants to go back
                    elif c2 == 9:
                        break
            #if the entered password is invalid
            else:
                print('INVALID PASSWORD, TRY AGAIN')
                print()
                continue
        #if the entered username is invalid
        else:
            print('INVALID USERNAME, TRY AGAIN')
            print()
            continue
#if the user wants to create a new account
elif c0 == 2:
        #asking the user to enter the required details
        speech('THANK YOU. FOR CHOOSING OUR BANK.
PLEASE ENTER THE REQUIRED DETAILS TO OPEN AN ACCOUNT
IN OUR BANK')
        sn = block(input('ENTER YOUR SURNAME: '))
        print()
        mn = block(input('ENTER YOUR MIDDLE NAME: '))
        print()
        lsn = block(input('ENTER YOUR LAST NAME: '))
        print()
        while(True):
            pn = input('ENTER YOUR PHONE NUMBER: ')
            print()
            #checking whether the entered phone number is valid or not
            if pn not in lpn and len(pn) == 10:
                break
            else:
                speech('AN ACCOUNT ALREADY EXISTS
CORRESPONDING TO THIS PHONE NUMBER. or YOU HAVE
ENTERED AN INVALID PHONE NUMBER. PLEASE ENTER A VALID
PHONE NUMBER')
                print('AN ACCOUNT ALREADY EXISTS
CORRESPONDING TO THIS PHONE NUMBER, OR YOU HAVE
ENTERED AN INVALID PHONE NUMBER, PLEASE ENTER A VALID
PHONE NUMBER.')
                print()
                continue
        while(True):
            an = input('ENTER YOUR AADHAR CARD NUMBER: ')
            print()
            #checking whether entered aadhar card number is valid or not
            if an not in lacn and len(an) == 12:
                break
            else:
                speech('AN ACCOUNT ALREADY EXISTS CORRESPONDING TO THIS AADHAR CARD NUMBER. or YOU HAVE ENTERED AN INVALID AADHAR NUMBER. PLEASE ENTER A VALID AADHAR CARD NUMBER')
                print('AN ACCOUNT ALREADY EXISTS CORRESPONDING TO THIS AADHAR CARD NUMBER, OR YOU HAVE ENTERED AN INVALID AADHAR NUMBER, PLEASE ENTER A VALID AADHAR CARD NUMBER.')
                print()
                continue
        while(True):
            pan = block(input('ENTER YOUR PAN CARD ID: '))
            print()
            #checking whether the entered pan card number is valid or not
            if pan[0:5].isalpha() and pan[5:9].isnumeric() and pan[-1].isalpha():
                break
            else:
                speech('YOU HAVE ENTERED AN INVALID PAN CARD ID. PLEASE ENTER A VALID PAN CARD ID')
                print('YOU HAVE ENTERED AN INVALID PAN CARD ID, PLEASE ENTER A VALID PAN CARD ID.')
                print()
                continue
        print()
        ad = block(input('ENTER YOUR PERMANENT ADDRESS: '))
        print()
        while(True):
            mb = int(input('DEPOSIT MONEY GREATER THAN OR EQUAL TO 3000 RUPEES: '))
            if mb >= 3000:
                break
            else:
                speech('THE AMOUNT DEPOSITED IS LESS THAN 3000 RUPEES. DEPOSITH AN AMOUNT GREATER THAN or EQUAL TO 3000 RUPEES')
                print('THE AMOUNT DEPOSITED IS LESS THAN 3000 RUPEES, DEPOSITH AN AMOUNT GREATER THAN OR EQUAL TO 3000 RUPEES.')
                print()
        print()
        while(True):
            us = input('CREATE YOUR USERNAME [IT MUST CONSIST ONLY 10 CHARACTERS]: ')
            print()
            #checking whether the entered username is valid or not
            if us not in lu and len(us) == 10:
                print('PLEASE REMEMBER YOUR USERNAME.')
                print()
                break
            else:
                speech('THE USERNAME YOU HAVE ENTERED ALREADY EXISTS OR IS IVALID. PLEASE ENTER A VALID USERNAME')
                print('THE USERNAME YOU HAVE ENTERED ALREADY EXISTS OR IS IVALID, PLEASE ENTER A VALID USERNAME.')
                print()
                continue
        while(True):
            p1 = stdiomask.getpass(prompt = 'CREATE YOUR PASSWORD: ')
            print()
            p2 = stdiomask.getpass(prompt = 'CONFIRM YOUR PASSWORD: ')
            print()
            #checking whether the entered passwords match or not
            if p1 == p2:
                print('PLEASE REMEMBER YOUR PASSWORD.')
                print()
                break
            else:
                speech('THE ENTERED PASSWORDS DO NOT MATCH PLEASE ENTER THEM CORRECTLY')
                print('THE ENTERED PASSWORDS DO NOT MATCH PLEASE ENTER THEM CORRECTLY.')
                print()
                continue
        rn = random.randint(100,500)
        speech('TO CONFIRM YOU ARE NOT A ROBOT PLEASE ENTER THE NUMBER SPOKEN BY me')
        print('TO CONFIRM YOU ARE NOT A ROBOT PLEASE ENTER THE NUMBER SPOKEN BY THE COMPUTER.')
        print()
        q = input('ENTER ANY KEY TO HEAR THE VOICE: ')
        if q == '1':
            speech(f'{rn}')
        else:
            speech(f'{rn}')
        uen = int(input('ENTER THE NUMBER SPOKEN BY THE COMPUTER: '))
        print()
        if uen == rn:
            while(True):
```

```
                            #creating a new account number for the user
                            nan = random.randint(100000000,999999999)
                            if nan not in lan:
                                break
                            else:
                                continue
                        print('THIS IS YOUR ACCOUNT NUMBER:',nan)
                        print()
                        print('PLEASE REMEMBER IT.')
                        print()
                        #creating all the required tables and entering the data to mysql
                        st = f"insert into user_details
values({nan},'{us}','{mn}','{d}','{an}','{pan}','{pn}','{ad}','{sn}','{lsn}')"
                        cur.execute(st)
                        mycon.commit()
                        st = f"insert into user_password
values('{us}','{p1}','{mn}',{nan})"
                        cur.execute(st)
                        mycon.commit()
                        st = f"insert into user_balance values({nan},{mb})"
                        cur.execute(st)
                        mycon.commit()
                        st = f"create table {nan}_trans(date varchar(10),time
varchar(10),amount int(9),description varchar(100))"
```

```
            cur.execute(st)
            mycon.commit()
            speech('THESE ARE YOUR ACCOUNT DETAILS. PLEASE
CHECK AND NOTE THEM')
            print('THESE ARE YOUR ACCOUNT DETAILS. PLEASE
CHECK AND NOTE THEM.')
            print()
            #printing all the details of the user to verify
            print('ACCOUNT NUMBER | USERNAME     | NAME
| DATE OPENED | AADHAR NUMBER | PAN CARD ID | PHONE
NUMBER | BALANCE | ADDRESS',' ' * (len(str(ad)) - 8),'|')
            print('-' * (132 + ((len(str(ad)) - 6))))
            print(nan,'     |',us,'  |',mn,' ' * (19 - len(str(mn))),'|',d,'|',an,'
|',pan,' |',pn,'  |',mb,' ' * (6 - len(str(mb))),'|',ad,'|')
            print()
            lu.append(us)
            lp.append(p1)
            ln.append(mn)
            lan.append(nan)
            lub[nan] = mb
        else:
            speech('SORRY. THE ENTERED NUMBER DOES NOT
MATCH')
            print('SORRY. THE ENTERED NUMBER DOES NOT
MATCH.')
            print()
            b = input('ENTER ANY KEY TO GO BACK: ')
            print()
            print()
            if b == 'b':
                break
            else:
                break
        #if the user wants to go back
        else:
            break
#if the user is an employee
elif c1 == 2:
    while(True):
        #asking the employee to enter the username
        u = input('ENTER USERNAME, OR B TO GO BACK: ')
        print()
        #if the employee enters b
        if u == 'B' or u == 'b':
            break
        #checking whether the entered username is valid
        if u in elu:
            n = elu.index(u)
            #asking the employee to enter the password
            p = stdiomask.getpass(prompt = 'ENTER PASSWORD, OR ENTER B GO BACK: ')
            print()
            print()
            #if the employee enters b
            if p == 'b' or p == 'B':
                break
            #checking whether the entered password is valid or not
            if p == elp[n]:
                #welcome note to the employee
                speech(f'WELCOME {eln[n]}')
                print('WELCOME,',eln[n])
                print()
                while(True):
                    print('1. CHECK USER DETAILS')
                    print('2. CHAECK USER TRANSACTIONS')
                    print('3. MODIFY RATES')
                    print('4. SANCTION LOANS')
                    print('5. DEDUCT LOAN DUES')
                    print('6. BACK')
                    print()
                    print()
                    c2 = int(input('ENTER YOUR CHOICE: '))
                    print()
```

```
print()
#if the employee wants to check the user details
if c2 == 1:
    while(True):
        #asking the employee to enter the account number
        tan = int(input('ENTER THE ACCOUNT NUMBER OF THE USER: '))
        print()
        #checking whether the entered account number is valid
        if tan in lan:
            tl = []
            st = f'select * from user_details where account_num = {tan}'
            cur.execute(st)
            x = cur.fetchall()
            for i in x:
                for j in i:
                    tl.append(j)
            tl.append(lub[tan])
            n = lan.index(tan)
            print('ACCOUNT NUMBER | USERNAME     | NAME               | DATE OPENED | AADHAR NUMBER | PAN CARD ID | PHONE NUMBER | BALANCE | ADDRESS',' ' * (len(str(tl[7])) - 8),'|')
            print('-' * (132 + ((len(str(tl[7])) - 6))))
```

```
                                        print(tl[0],'     |',tl[1],'  |',tl[2],' ' * (19 - len(str(tl[2]))),'|',tl[3],' |',tl[4],' |',tl[5],' |',tl[6],'  |',tl[-1],' ' * (6 - len(str(tl[-1]))),'|',tl[7],'|')
                                        print()
                                        speech(f'THESE ARE THE ACCOUNT DETAILS OF {ln[n]}')
                                    #if the entered account number is invalid
                                    else:
                                        speech('THE ENTERED ACCOUNT NUMBER IS INVALID. PLEASE ENTER A VALID ACCOUNT NUMBER')
                                        print('THE ENTERED ACOOUNT NUMBER IS INVALID, PLEASE ENTER A VALID ACCOUNT NUMBER.')
                                        print()
                                        continue
                                    b = input('ENTER N TO FURTHER CHECK DETAILS, OR ANY KEY TO GO BACK: ')
                                    print()
                                    print()
                                    if b == 'n' or b == 'N':
                                        continue
                                    else:
                                        break
                            #if the employee wants to check users transactions
                            elif c2 == 2:
                                while(True):
```

```
#asking the employee to enter the account number
tan = int(input('ENTER THE ACCOUNT NUMBER OF THE USER: '))
print()
#checking whether the entered account number is valid
if tan in lan:
    print('DATE(Y-M-D) | TIME(H-M-S) | AMOUNT | DESCRIPTION                    ')
    print('------------------------------------------------------------------------------------')
    st = f'select * from {an}_trans'
    cur.execute(st)
    x = cur.fetchall()
    for i in x:
        tl = []
        for j in i:
            tl.append(j)
        print(tl[0],' |',tl[1],'   |',tl[2],' ' * (8 - len(str(tl[2]))),'|',tl[3])
    print()
    n = lan.index(tan)
    speech(f'THESE ARE THE ACCOUNT TRANSACTIONS OF {ln[n]}')
#if the entered account number is invalid
else:
```

```
            speech('THE ENTERED ACCOUNT NUMBER IS INVALID. PLEASE ENTER A VALID ACCOUNT NUMBER')
            print('THE ENTERED ACOOUNT NUMBER IS INVALID, PLEASE ENTER A VALID ACCOUNT NUMBER.')
            print()
            continue
        b = input('ENTER N TO FURTHER CHECK USER TRANSACTIONS, OR ANY KEY TO GO BACK: ')
        print()
        print()
        if b == 'N' or b == 'n':
            continue
        else:
            break
#if the empolyee wants to modify the loan rates
elif c2 == 3:
    while(True):
        #asking the employee to enter the loan code
        lc = input('ENTER THE LOAN CODE: ')
        print()
        #checking whether the entered loan code is valid
        if lc in llc:
            n = llc.index(lc)
            r = int(input('ENTER THE NEW RATE: '))
```

```
                    print()
                    st = f"update loan_code set rate = {r} where loan_code = '{lc}'"
                    cur.execute(st)
                    mycon.commit()
                    speech(f'THE LOAN RATE OF {lln[n]} HAS BEEN SUCCESSFULLY CHANGED TO {r} PERCENT')
                    print('THE LOAN RATE OF',lln[n],'HAS BEEN SUCCESSFULLY CHANGED TO',r,'PERCENT.')
                    print()
                #if the entered loan code is invalid
                else:
                    speech('THE CODE ENTERED DOES NOT EXIST. PLEASE ENTER A VALID LOAN CODE')
                    print('THE CODE ENTERED DOES NOT EXIST, PLEASE ENTER A VALID LOAN CODE')
                    print()
                    continue
                b = input('ENTER N TO MODIFY RATES FURTHER, OR ANY KEY TO GO BACK: ')
                print()
                print()
                if b == 'N' or b == 'n':
                    continue
                else:
```

```
            break
    #if the employee wants to sanction loans
    elif c2 == 4:
        while(True):
            speech('THESE ARE THE PENDING LOAN REQUESTS YOU HAVE TO SANCTION')
            print('THESE ARE THE PENDING LOAN REQUESTS YOU HAVE TO SANCTION.')
            print()
            cur.execute("select * from loan_details where status = 'pending'")
            x = cur.fetchall()
            print('ACCOUNT NUMBER | NAME                | AMOUNT    | TIME(IN YEARS) | LOAN NAME      | STATUS')
            print('---------------------------------------------------------------------------------------------')
            for i in x:
                tl = []
                for j in i:
                    tl.append(j)
                print(tl[0],'     |',tl[1],' ' * (19 - len(str(tl[1]))),'|',tl[2],' ' * (8 - len(str(tl[2]))),'|',tl[3],' ' * 12,'|',tl[4],' ' * (13 - len(tl[4])),'|',tl[5])
            print()
            while(True):
                san = int(input('ENTER THE ACCOUNT NUMBER OF THE USER YOU WANT TO SANCTION THE LOAN: '))
```

```
print()
if san not in lpl:
    speech('INVALID ACCOUNT NUMBER. PLEASE ENTER A VALID ACCOUNT NUMBER WHO HAS APPLIED FOR A LOAN')
    print('INVALID ACCOUNT NUMBER, PLEASE ENTER A VALID ACCOUNT NUMBER WHO HAS APPLIED FOR A LOAN.')
    print()
    continue
else:
    st = f"update loan_details set status = 'APPROVED' where account_num = {san}"
    cur.execute(st)
    mycon.commit()
    lptr = []
    cur.execute(f"select amount,time,loan_name from loan_details where account_num = {san}")
    x = cur.fetchall()
    a = 0
    t = 0
    ln = ''
    for i in x:
        i = list(i)
        a = i[0]
        t = i[1]
        ln = i[2]
    r = lrln[ln]
    i = (a * t * r) / 100
    di = int((i + (a * (r / 100))) / 12)
    ld = []
    pd = d[8:10]
    if pd == '31' or pd == '30' or pd == '29':
        pd = '28'
    ld.append(d)
    for i in range(t * 12):
        if ld[-1][5:7] == '12':
            y = str((int(ld[-1][0:4]) + 1))
            ts = y + '-01' + '-' + pd
            ld.append(ts)
        else:
            m = str((int(ld[-1][5:7]) + 1))
            if len(m) == 1:
                m = '0' + m
            ts = ld[-1][0:4] + '-' + m + '-' + pd
            ld.append(ts)
    ld.remove(d)
    st = f'create table {san}_loan(date varchar(11),amount int(9),status varchar(20))'
```

```
                        cur.execute(st)
                        mycon.commit()
                        for i in ld:
                            st = f"insert into {san}_loan
values('{i}',{di},'NOT YET DEDUCTED')"
                            cur.execute(st)
                            mycon.commit()
                        st = f"update user_balance set balance = balance +
{a} where account_num = {san}"
                        cur.execute(st)
                        mycon.commit()
                        tdt =
datetime.datetime.now(pytz.timezone('Asia/Kolkata'))
                        tdts = str(tdt)
                        t = tdts[11:19]
                        st = f"insert into {san}_trans
values('{d}','{t}',{a},'CREDITED BY BANK ON LOAN')"
                        cur.execute(st)
                        mycon.commit()
                        speech('THE LOAN HAS BEEN
SUCCESSFULLY SANCTIONED')
                        print('THE LOAN HAS BEEN SUCCESSFULLY
SANCTIONED.')
                        print()
                        break
```

```
                b = input('ENTER N TO SANCTION OTHER LOANS, OR ANY KEY TO GO BACK: ')
                print()
                print()
                if b == 'N' or b == 'n':
                    continue
                else:
                    break
        #if the employee wants to deduct loan dues
        elif c2 == 5:
            while(True):
                st = "select account_num from loan_details where status = 'APPROVED'"
                cur.execute(st)
                x = cur.fetchall()
                print('THESE ARE THE ACCOUNT NUMBERS YOU NEED TO DEDUCT DUES FROM: ')
                print()
                tl = []
                for i in x:
                    for j in i:
                        tl.append(j)
                print('| ACCOUNT NUMBERS |')
                print('-------------------')
```

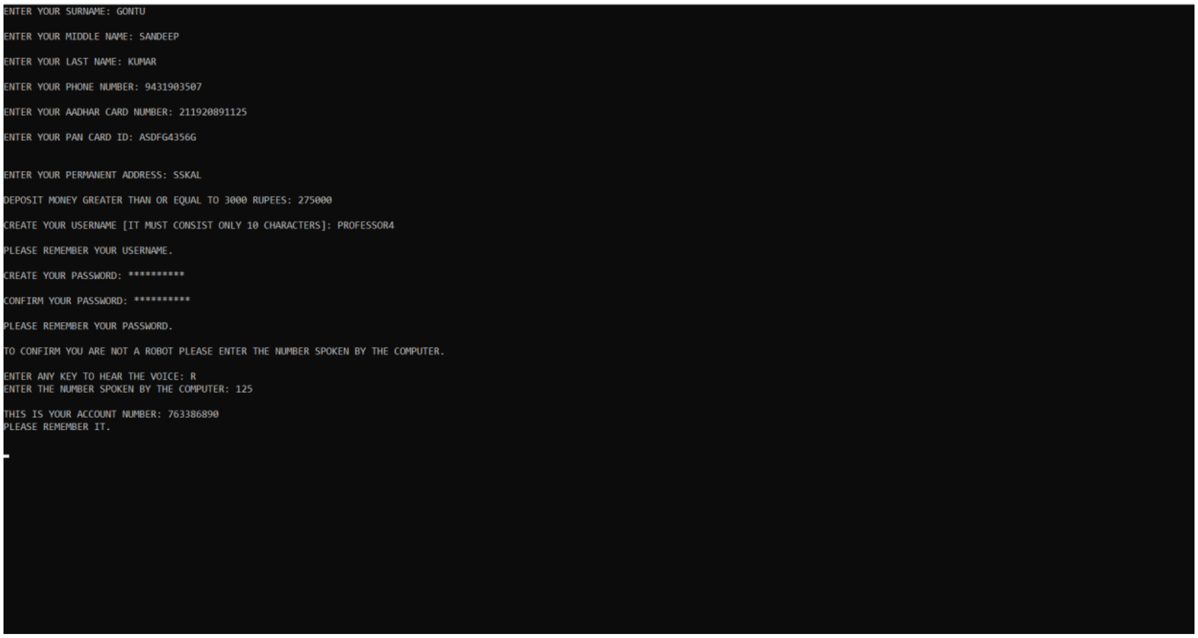
```
for i in tl:
    print('|',i,'      |')
print()
tan = int(input('ENTER THE ACCOUNT NUMBER: '))
print()
if tan in tl:
    st = f"select * from {tan}_loan"
    cur.execute(st)
    x = cur.fetchall()
    print('DATE        | AMOUNT | STATUS')
    print('----------------------------------')
    for i in x:
        tl = []
        for j in i:
            tl.append(j)
        print(tl[0],'|',tl[1],' |',tl[2])
    print()
    a = int(input('ENTER THE AMOUNT: '))
    print()
    st = f"update {tan}_loan set status = 'DEDUCTED' where date = {d}"
    cur.execute(st)
    mycon.commit()
```

```
                        st = f"update user_balance set balance = balance - {a}
where account_num = {tan}"

                        cur.execute(st)

                        mycon.commit()

                        tdt =
datetime.datetime.now(pytz.timezone('Asia/Kolkata'))

                        tdts = str(tdt)

                        t = tdts[11:19]

                        st = f"insert into {tan}_trans
values('{d}','{t}',{a},'DEDUCTED BY BANK, LOAN DUE')"

                        cur.execute(st)

                        mycon.commit()

                        speech('THE AMOUNT HAS BEEN
SUCCESSFULLY DEDUCTED')

                        print('THE AMOUNT HAS BEEN
SUCCESSFULLY DEDUCTED.')

                        print()

                        print('DELETE RECORD OF LOAN DUE [IF THE
USER HAS COMPLETED HIS LOAN DUES ONLY THEN DELETE
THEIR RECORD]')

                        print()

                        o = input('ENTER D TO DELETE, OR ANY KEY
TO GO BACK: ')

                        print()

                        if o == 'D' or o == 'd':

                            st = f"drop table {tan}_loan"
```

```
                                            cur.execute(st)
                                            mycon.commit()
                                            st = f"delete from loan_details where account_num
= {tan}"
                                            cur.execute()
                                            mycon.commit()
                                        else:
                                            print('REMEMBER TO DELETE THE RECORD
AFTER THE USER COMPLETES THEIR LOAN DUES.')
                                            print()
                                    b = input('ENTER N TO DEDUCT LOAN DUES
AGAIN, OR ANY KEY TO GO BACK: ')
                                    print()
                                    print()
                                    if b == 'N' or b == 'n':
                                        continue
                                    else:
                                        break
                            #if the wants to go back
                            else:
                                break
                    #if the entered password is incorrect
                    else:
                        print('INVALID PASSWORD, TRY AGAIN')
                        print()
                        continue
                #if the entered username is invalid
                else:
                    print('INVALID USERNAME, TRY AGAIN')
                    print()
                    continue
        #if the employee wants to go back
        else:
            break

#if connection fails
else:
    speech('SORRY. AT THIS MOMENT OUR SERVICE IS NOT WORKING PROPERLY. PLEASE TRY LATER')
    print('SORRY, AT THIS MOMENT OUR SERVICE IS NOT WORKING PROPERLY, PLEASE TRY LATER.')
```

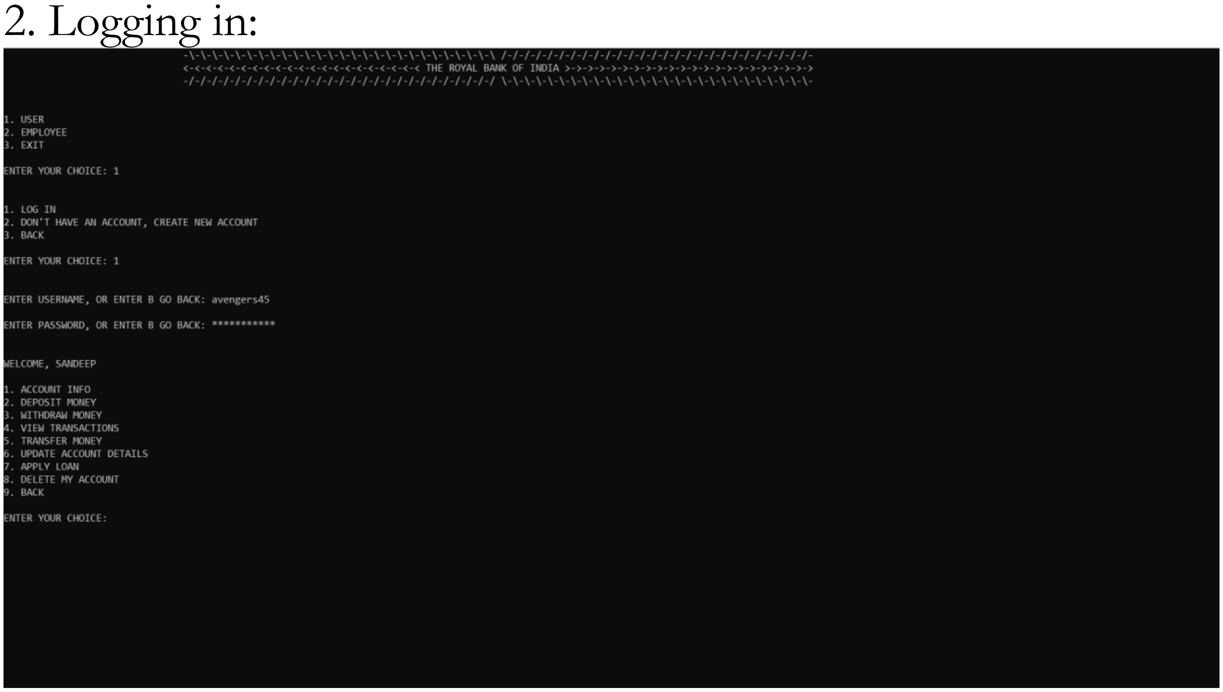
OUTPUT OF THE PROGRAM

1. Creating a new account :



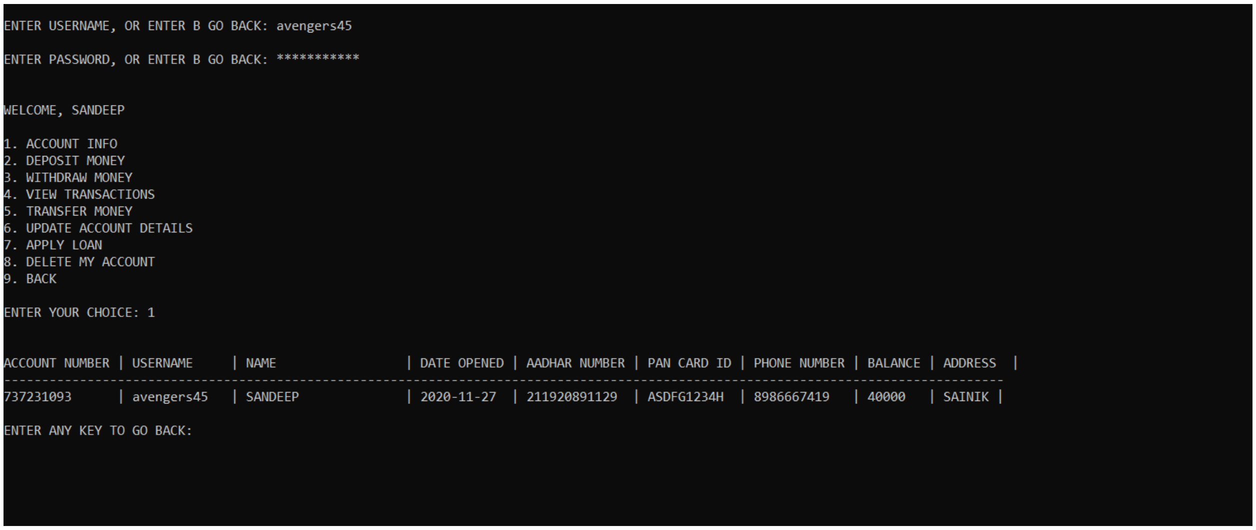
```
ENTER YOUR SURNAME: GONTU

ENTER YOUR MIDDLE NAME: SANDEEP

ENTER YOUR LAST NAME: KUMAR

ENTER YOUR PHONE NUMBER: 9431903507

ENTER YOUR AADHAR CARD NUMBER: 211920891125

ENTER YOUR PAN CARD ID: ASDFG4356G


ENTER YOUR PERMANENT ADDRESS: SSKAL

DEPOSIT MONEY GREATER THAN OR EQUAL TO 3000 RUPEES: 275000

CREATE YOUR USERNAME [IT MUST CONSIST ONLY 10 CHARACTERS]: PROFESSOR4

PLEASE REMEMBER YOUR USERNAME.

CREATE YOUR PASSWORD: **********

CONFIRM YOUR PASSWORD: **********

PLEASE REMEMBER YOUR PASSWORD.

TO CONFIRM YOU ARE NOT A ROBOT PLEASE ENTER THE NUMBER SPOKEN BY THE COMPUTER.

ENTER ANY KEY TO HEAR THE VOICE: R
ENTER THE NUMBER SPOKEN BY THE COMPUTER: 125

THIS IS YOUR ACCOUNT NUMBER: 763386890
PLEASE REMEMBER IT.
```

2. Logging in:



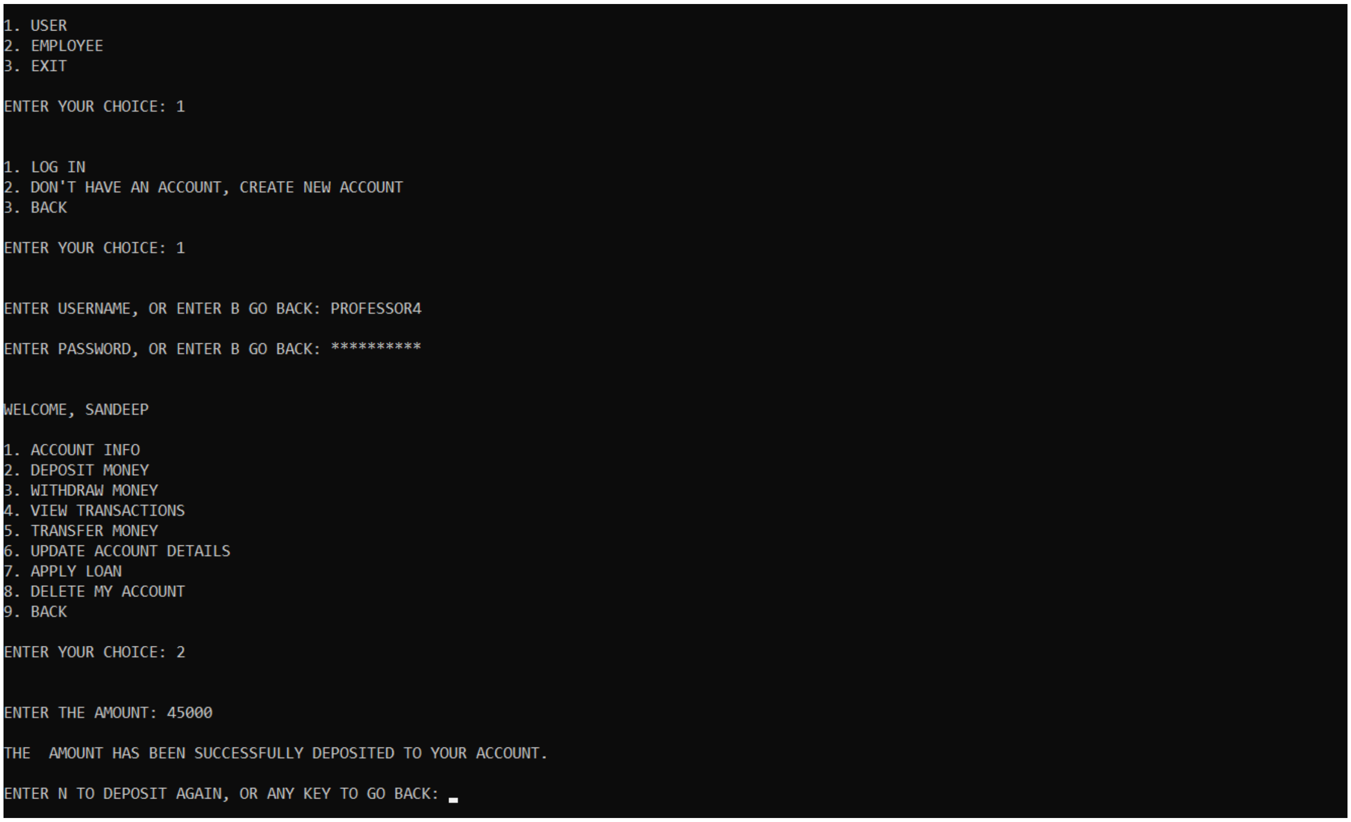
```
                                  -\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\ /-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-
                                  <-<-<-<-<-<-<-<-<-<-<-<-<-<-<-<-<-<-<-<-< THE ROYAL BANK OF INDIA >->->->->->->->->->->->->->->->->->->->->->->
                                  -/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/-/ \-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-\-


1. USER
2. EMPLOYEE
3. EXIT

ENTER YOUR CHOICE: 1


1. LOG IN
2. DON'T HAVE AN ACCOUNT, CREATE NEW ACCOUNT
3. BACK

ENTER YOUR CHOICE: 1


ENTER USERNAME, OR ENTER B GO BACK: avengers45

ENTER PASSWORD, OR ENTER B GO BACK: ***********


WELCOME, SANDEEP

1. ACCOUNT INFO
2. DEPOSIT MONEY
3. WITHDRAW MONEY
4. VIEW TRANSACTIONS
5. TRANSFER MONEY
6. UPDATE ACCOUNT DETAILS
7. APPLY LOAN
8. DELETE MY ACCOUNT
9. BACK

ENTER YOUR CHOICE:
```

3.Getting your account info :



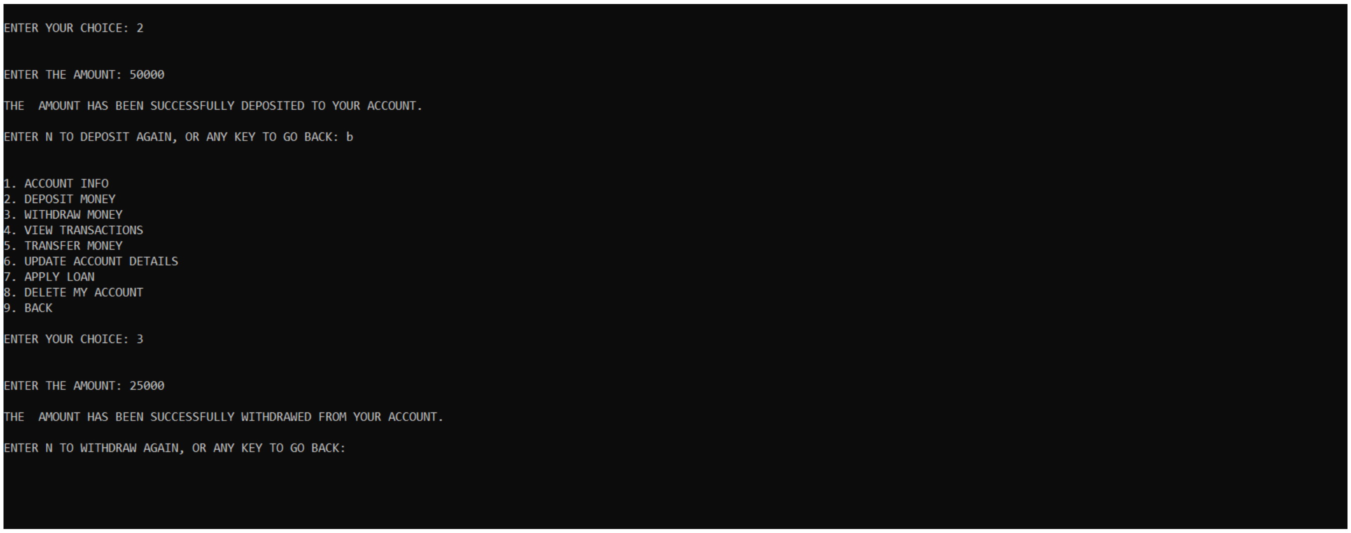
```
ENTER USERNAME, OR ENTER B GO BACK: avengers45

ENTER PASSWORD, OR ENTER B GO BACK: ***********


WELCOME, SANDEEP

1. ACCOUNT INFO
2. DEPOSIT MONEY
3. WITHDRAW MONEY
4. VIEW TRANSACTIONS
5. TRANSFER MONEY
6. UPDATE ACCOUNT DETAILS
7. APPLY LOAN
8. DELETE MY ACCOUNT
9. BACK

ENTER YOUR CHOICE: 1


ACCOUNT NUMBER | USERNAME     | NAME                 | DATE OPENED | AADHAR NUMBER | PAN CARD ID | PHONE NUMBER | BALANCE | ADDRESS  |
-----------------------------------------------------------------------------------------------------------------------------------
737231093      | avengers45   | SANDEEP              | 2020-11-27  | 211920891129  | ASDFG1234H  | 8986667419   | 40000   | SAINIK |

ENTER ANY KEY TO GO BACK:
```

4.Depositting money:



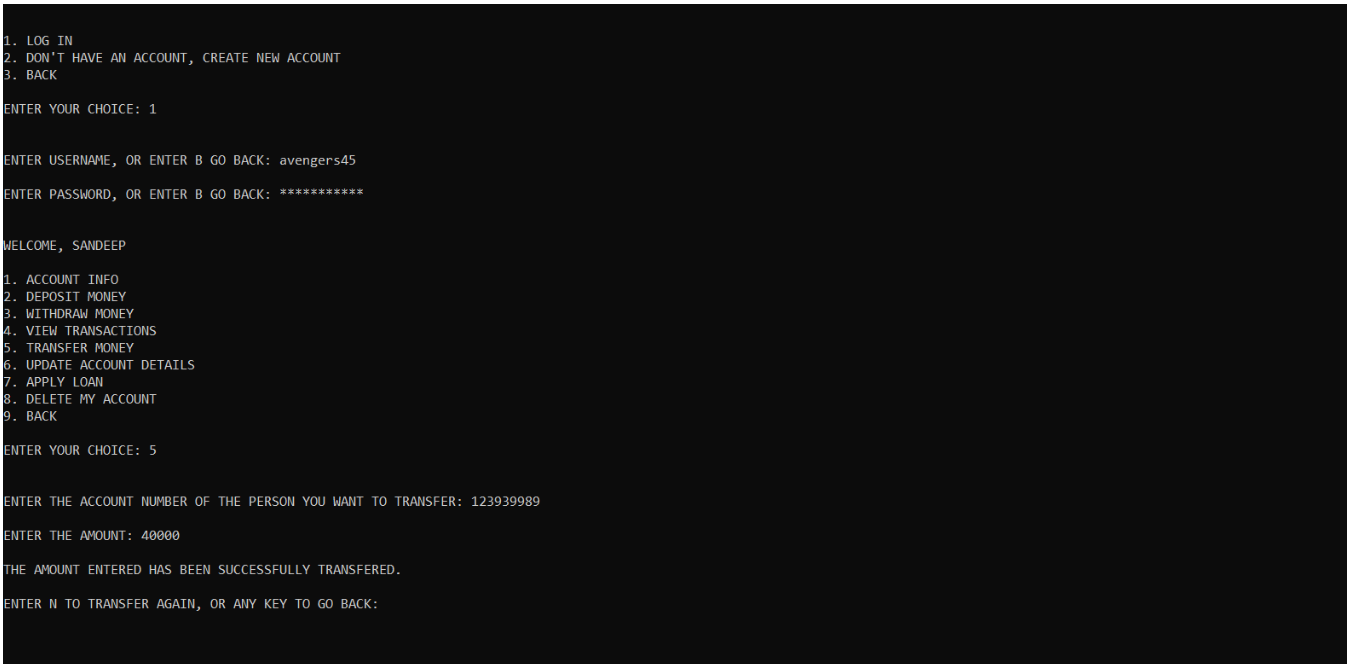
```
1. USER
2. EMPLOYEE
3. EXIT

ENTER YOUR CHOICE: 1


1. LOG IN
2. DON'T HAVE AN ACCOUNT, CREATE NEW ACCOUNT
3. BACK

ENTER YOUR CHOICE: 1


ENTER USERNAME, OR ENTER B GO BACK: PROFESSOR4

ENTER PASSWORD, OR ENTER B GO BACK: **********


WELCOME, SANDEEP

1. ACCOUNT INFO
2. DEPOSIT MONEY
3. WITHDRAW MONEY
4. VIEW TRANSACTIONS
5. TRANSFER MONEY
6. UPDATE ACCOUNT DETAILS
7. APPLY LOAN
8. DELETE MY ACCOUNT
9. BACK

ENTER YOUR CHOICE: 2


ENTER THE AMOUNT: 45000

THE  AMOUNT HAS BEEN SUCCESSFULLY DEPOSITED TO YOUR ACCOUNT.

ENTER N TO DEPOSIT AGAIN, OR ANY KEY TO GO BACK: _
```

5. Withdrawing money:



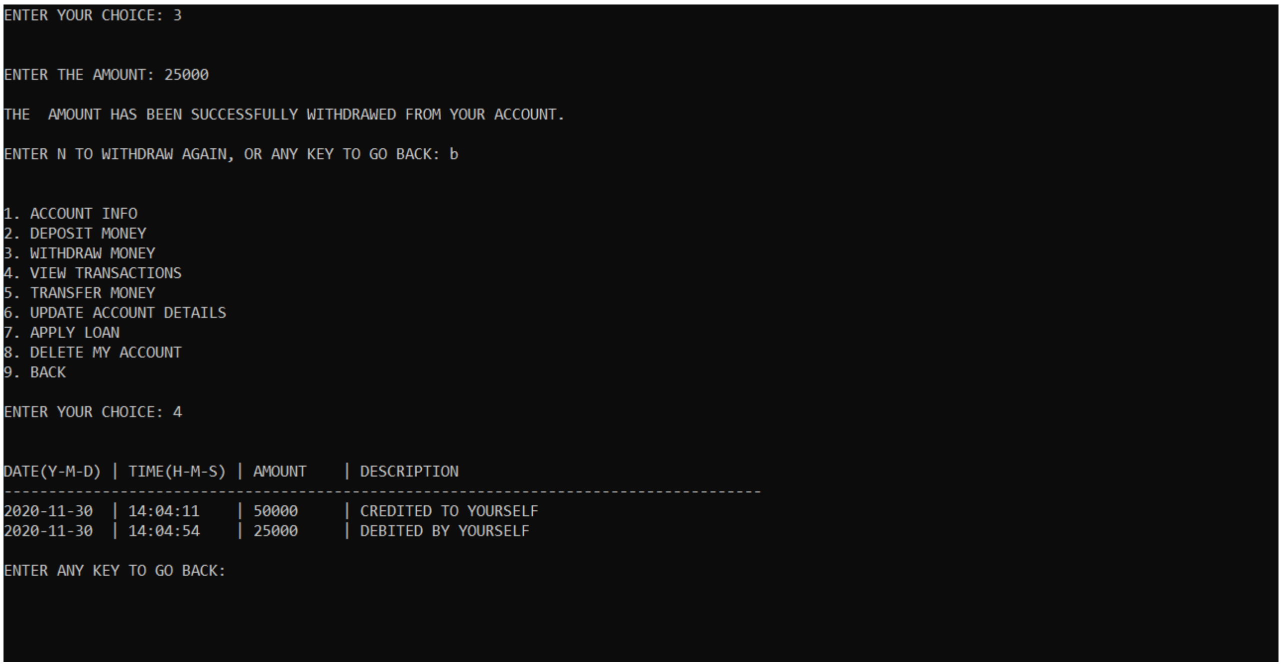
```
ENTER YOUR CHOICE: 2


ENTER THE AMOUNT: 50000

THE  AMOUNT HAS BEEN SUCCESSFULLY DEPOSITED TO YOUR ACCOUNT.

ENTER N TO DEPOSIT AGAIN, OR ANY KEY TO GO BACK: b


1. ACCOUNT INFO
2. DEPOSIT MONEY
3. WITHDRAW MONEY
4. VIEW TRANSACTIONS
5. TRANSFER MONEY
6. UPDATE ACCOUNT DETAILS
7. APPLY LOAN
8. DELETE MY ACCOUNT
9. BACK

ENTER YOUR CHOICE: 3


ENTER THE AMOUNT: 25000

THE  AMOUNT HAS BEEN SUCCESSFULLY WITHDRAWED FROM YOUR ACCOUNT.

ENTER N TO WITHDRAW AGAIN, OR ANY KEY TO GO BACK:
```

6.Transferring money:



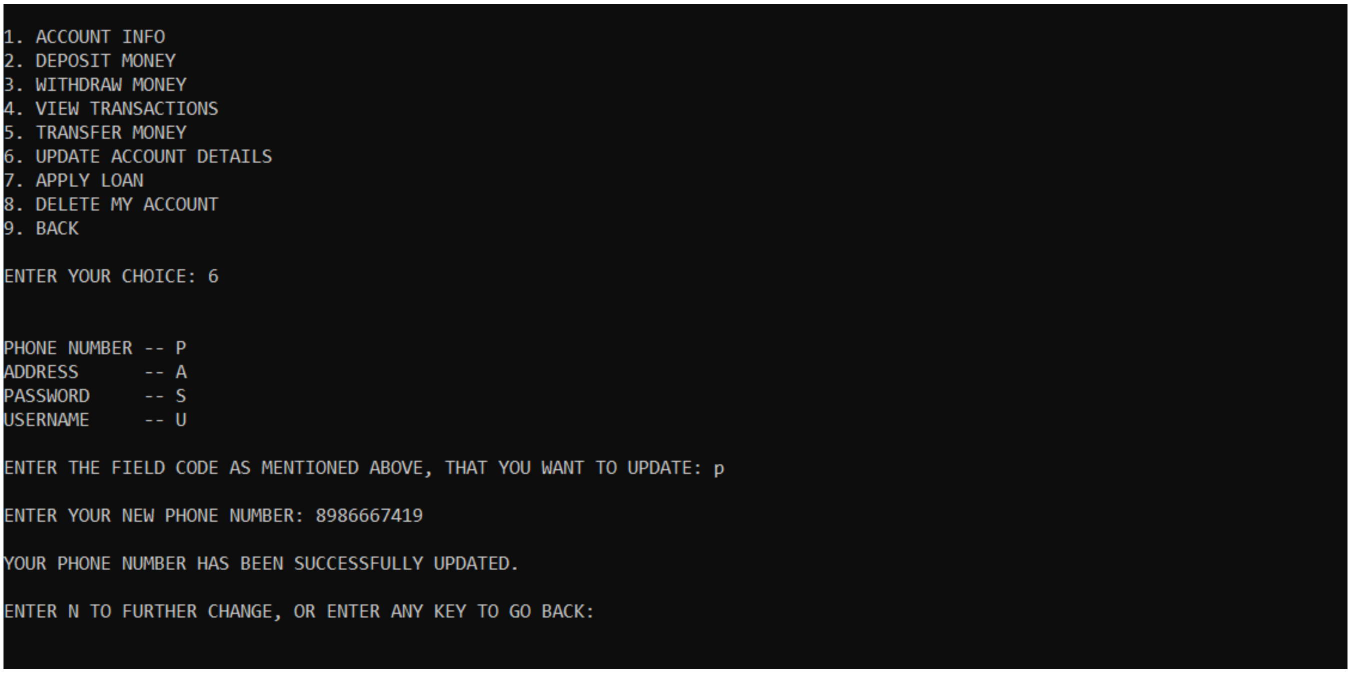
```
1. LOG IN
2. DON'T HAVE AN ACCOUNT, CREATE NEW ACCOUNT
3. BACK

ENTER YOUR CHOICE: 1


ENTER USERNAME, OR ENTER B GO BACK: avengers45

ENTER PASSWORD, OR ENTER B GO BACK: ***********


WELCOME, SANDEEP

1. ACCOUNT INFO
2. DEPOSIT MONEY
3. WITHDRAW MONEY
4. VIEW TRANSACTIONS
5. TRANSFER MONEY
6. UPDATE ACCOUNT DETAILS
7. APPLY LOAN
8. DELETE MY ACCOUNT
9. BACK

ENTER YOUR CHOICE: 5


ENTER THE ACCOUNT NUMBER OF THE PERSON YOU WANT TO TRANSFER: 123939989

ENTER THE AMOUNT: 40000

THE AMOUNT ENTERED HAS BEEN SUCCESSFULLY TRANSFERED.

ENTER N TO TRANSFER AGAIN, OR ANY KEY TO GO BACK:
```

7.View transaction details:



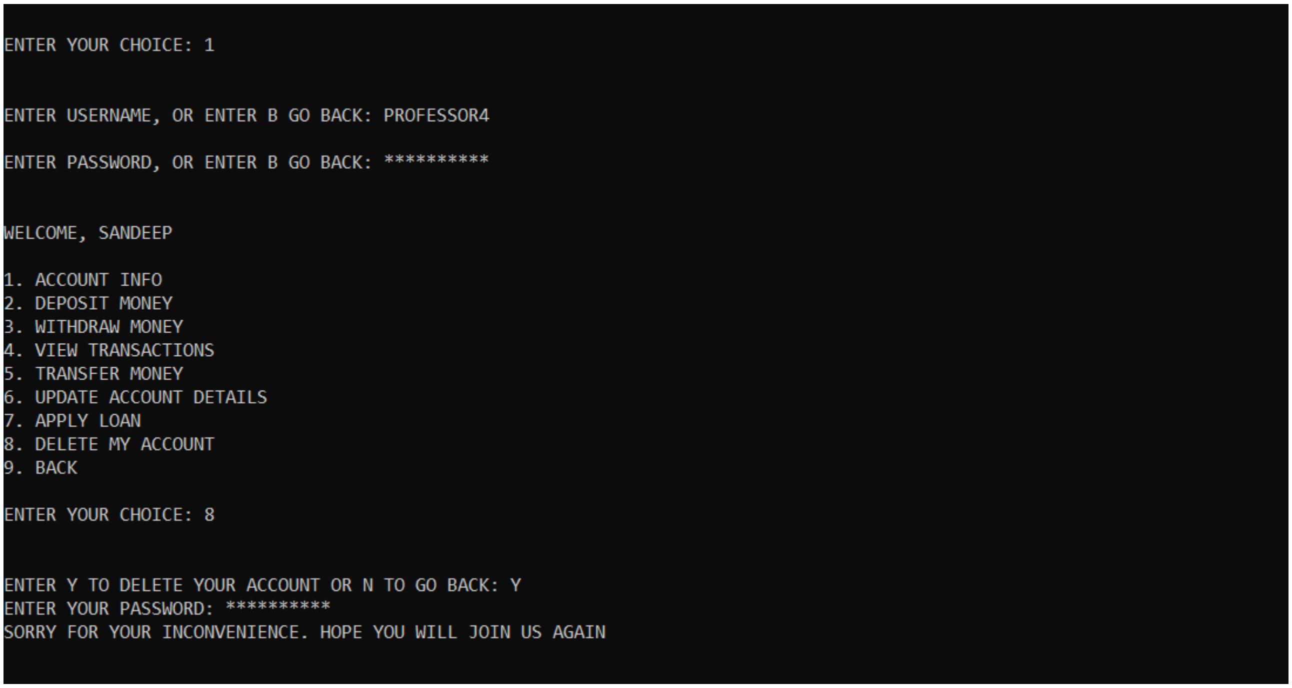
```
ENTER YOUR CHOICE: 3


ENTER THE AMOUNT: 25000

THE  AMOUNT HAS BEEN SUCCESSFULLY WITHDRAWED FROM YOUR ACCOUNT.

ENTER N TO WITHDRAW AGAIN, OR ANY KEY TO GO BACK: b


1. ACCOUNT INFO
2. DEPOSIT MONEY
3. WITHDRAW MONEY
4. VIEW TRANSACTIONS
5. TRANSFER MONEY
6. UPDATE ACCOUNT DETAILS
7. APPLY LOAN
8. DELETE MY ACCOUNT
9. BACK

ENTER YOUR CHOICE: 4


DATE(Y-M-D) | TIME(H-M-S) | AMOUNT    | DESCRIPTION
-------------------------------------------------------------------------------------
2020-11-30  | 14:04:11    | 50000     | CREDITED TO YOURSELF
2020-11-30  | 14:04:54    | 25000     | DEBITED BY YOURSELF

ENTER ANY KEY TO GO BACK:
```

8 .Updating account details:



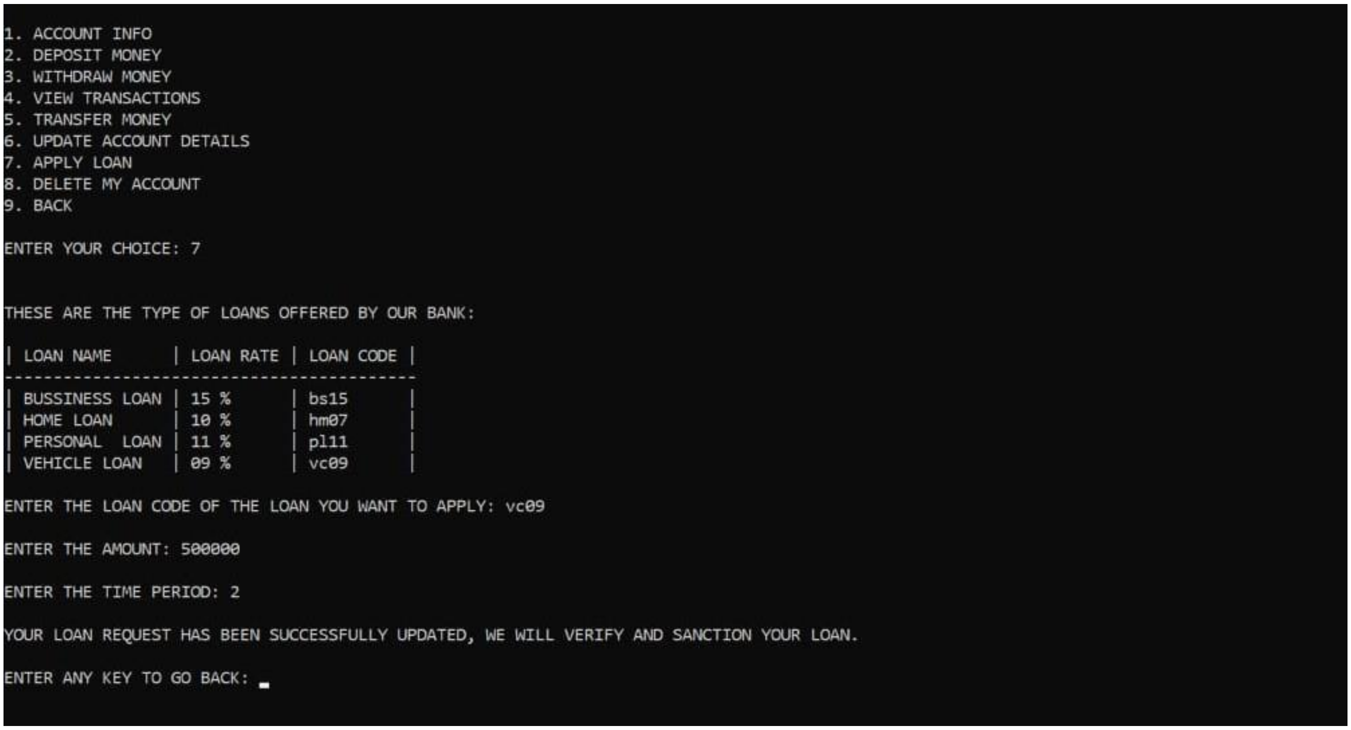
```
1. ACCOUNT INFO
2. DEPOSIT MONEY
3. WITHDRAW MONEY
4. VIEW TRANSACTIONS
5. TRANSFER MONEY
6. UPDATE ACCOUNT DETAILS
7. APPLY LOAN
8. DELETE MY ACCOUNT
9. BACK

ENTER YOUR CHOICE: 6


PHONE NUMBER -- P
ADDRESS      -- A
PASSWORD     -- S
USERNAME     -- U

ENTER THE FIELD CODE AS MENTIONED ABOVE, THAT YOU WANT TO UPDATE: p

ENTER YOUR NEW PHONE NUMBER: 8986667419

YOUR PHONE NUMBER HAS BEEN SUCCESSFULLY UPDATED.

ENTER N TO FURTHER CHANGE, OR ENTER ANY KEY TO GO BACK:
```

9 . Deleting your account:



```
ENTER YOUR CHOICE: 1


ENTER USERNAME, OR ENTER B GO BACK: PROFESSOR4

ENTER PASSWORD, OR ENTER B GO BACK: **********


WELCOME, SANDEEP

1. ACCOUNT INFO
2. DEPOSIT MONEY
3. WITHDRAW MONEY
4. VIEW TRANSACTIONS
5. TRANSFER MONEY
6. UPDATE ACCOUNT DETAILS
7. APPLY LOAN
8. DELETE MY ACCOUNT
9. BACK

ENTER YOUR CHOICE: 8


ENTER Y TO DELETE YOUR ACCOUNT OR N TO GO BACK: Y
ENTER YOUR PASSWORD: **********
SORRY FOR YOUR INCONVENIENCE. HOPE YOU WILL JOIN US AGAIN
```

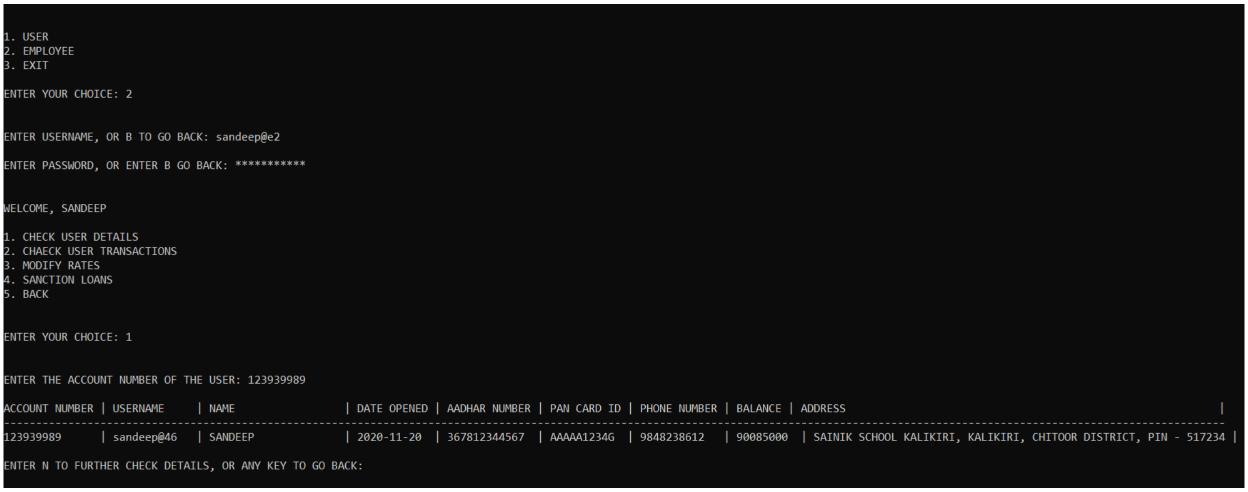
10 . Applying for loan :



```
1. ACCOUNT INFO
2. DEPOSIT MONEY
3. WITHDRAW MONEY
4. VIEW TRANSACTIONS
5. TRANSFER MONEY
6. UPDATE ACCOUNT DETAILS
7. APPLY LOAN
8. DELETE MY ACCOUNT
9. BACK

ENTER YOUR CHOICE: 7


THESE ARE THE TYPE OF LOANS OFFERED BY OUR BANK:

| LOAN NAME      | LOAN RATE | LOAN CODE |
------------------------------------------
| BUSSINESS LOAN | 15 %      | bs15      |
| HOME LOAN      | 10 %      | hm07      |
| PERSONAL  LOAN | 11 %      | pl11      |
| VEHICLE LOAN   | 09 %      | vc09      |

ENTER THE LOAN CODE OF THE LOAN YOU WANT TO APPLY: vc09

ENTER THE AMOUNT: 500000

ENTER THE TIME PERIOD: 2

YOUR LOAN REQUEST HAS BEEN SUCCESSFULLY UPDATED, WE WILL VERIFY AND SANCTION YOUR LOAN.

ENTER ANY KEY TO GO BACK: _
```

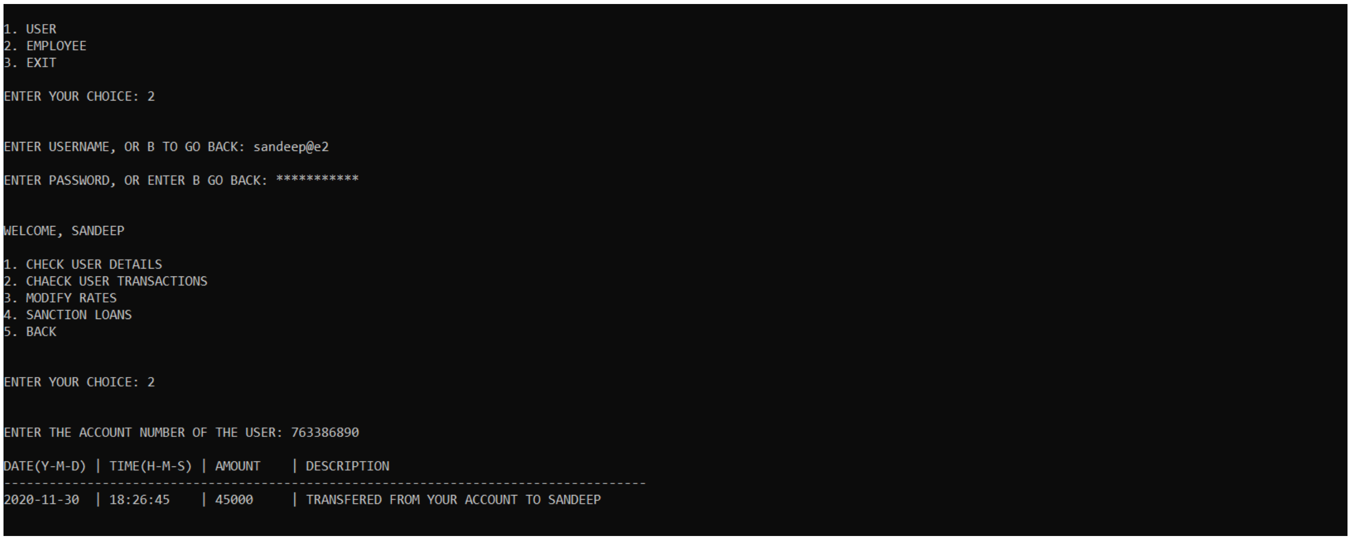
- Accessing as an employee:

1. Checking user details:



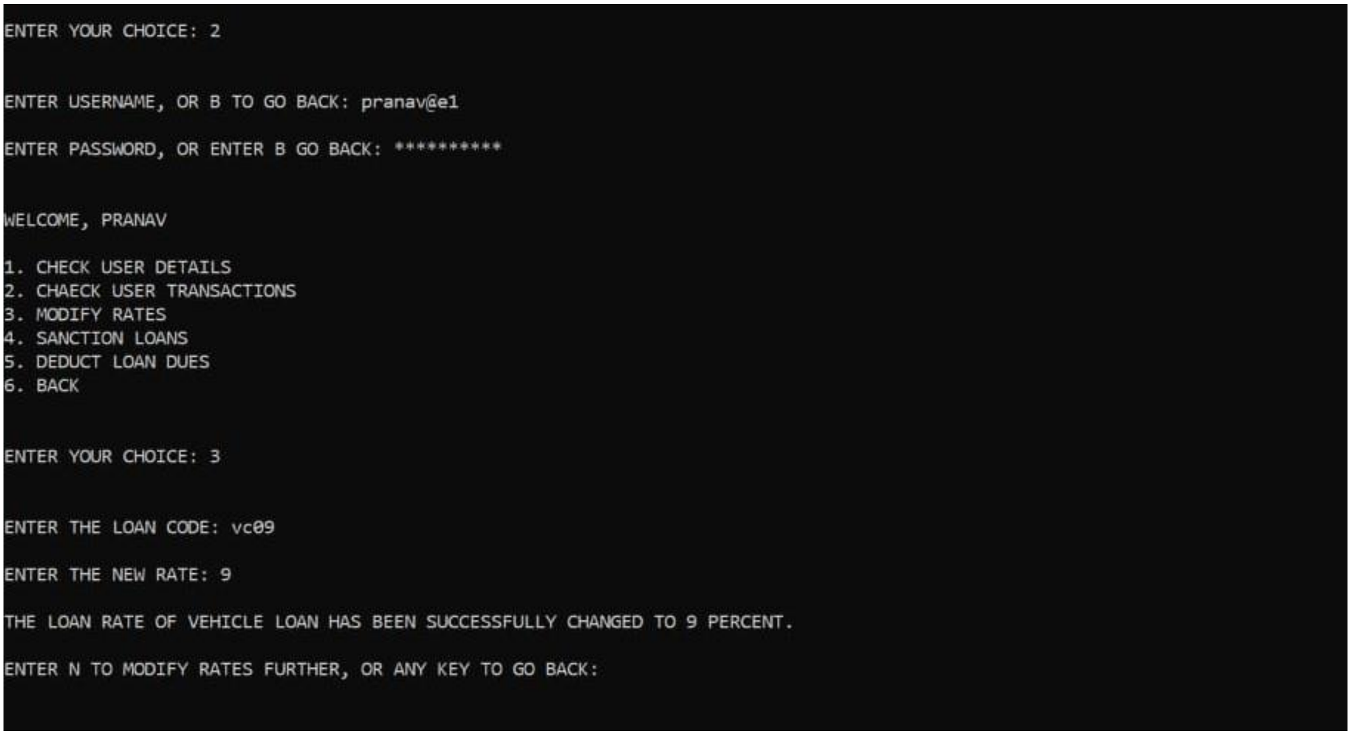
```
1. USER
2. EMPLOYEE
3. EXIT

ENTER YOUR CHOICE: 2


ENTER USERNAME, OR B TO GO BACK: sandeep@e2

ENTER PASSWORD, OR ENTER B GO BACK: ***********


WELCOME, SANDEEP

1. CHECK USER DETAILS
2. CHAECK USER TRANSACTIONS
3. MODIFY RATES
4. SANCTION LOANS
5. BACK


ENTER YOUR CHOICE: 1


ENTER THE ACCOUNT NUMBER OF THE USER: 123939989

ACCOUNT NUMBER | USERNAME     | NAME                 | DATE OPENED | AADHAR NUMBER | PAN CARD ID | PHONE NUMBER | BALANCE | ADDRESS                                                             |
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
123939989      | sandeep@46   | SANDEEP              | 2020-11-20  | 367812344567  | AAAAA1234G  | 9848238612   | 90085000  | SAINIK SCHOOL KALIKIRI, KALIKIRI, CHITOOR DISTRICT, PIN - 517234 |

ENTER N TO FURTHER CHECK DETAILS, OR ANY KEY TO GO BACK:
```

2. Checking transaction details:



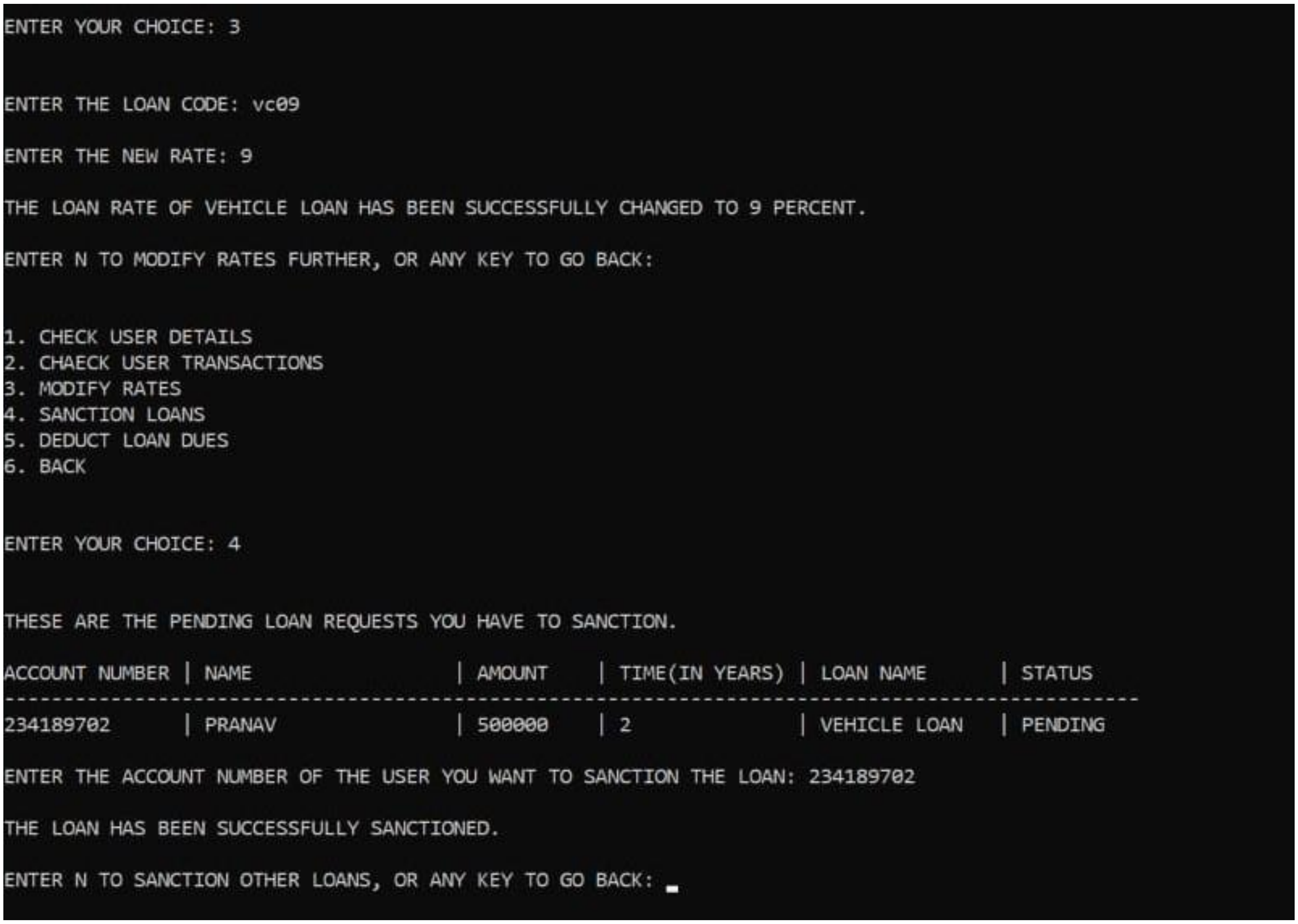
```
1. USER
2. EMPLOYEE
3. EXIT

ENTER YOUR CHOICE: 2


ENTER USERNAME, OR B TO GO BACK: sandeep@e2

ENTER PASSWORD, OR ENTER B GO BACK: ***********


WELCOME, SANDEEP

1. CHECK USER DETAILS
2. CHAECK USER TRANSACTIONS
3. MODIFY RATES
4. SANCTION LOANS
5. BACK


ENTER YOUR CHOICE: 2


ENTER THE ACCOUNT NUMBER OF THE USER: 763386890

DATE(Y-M-D) | TIME(H-M-S) | AMOUNT    | DESCRIPTION
--------------------------------------------------------------------------------------
2020-11-30  | 18:26:45    | 45000     | TRANSFERED FROM YOUR ACCOUNT TO SANDEEP
```

3 . Modifying loan interest rates :



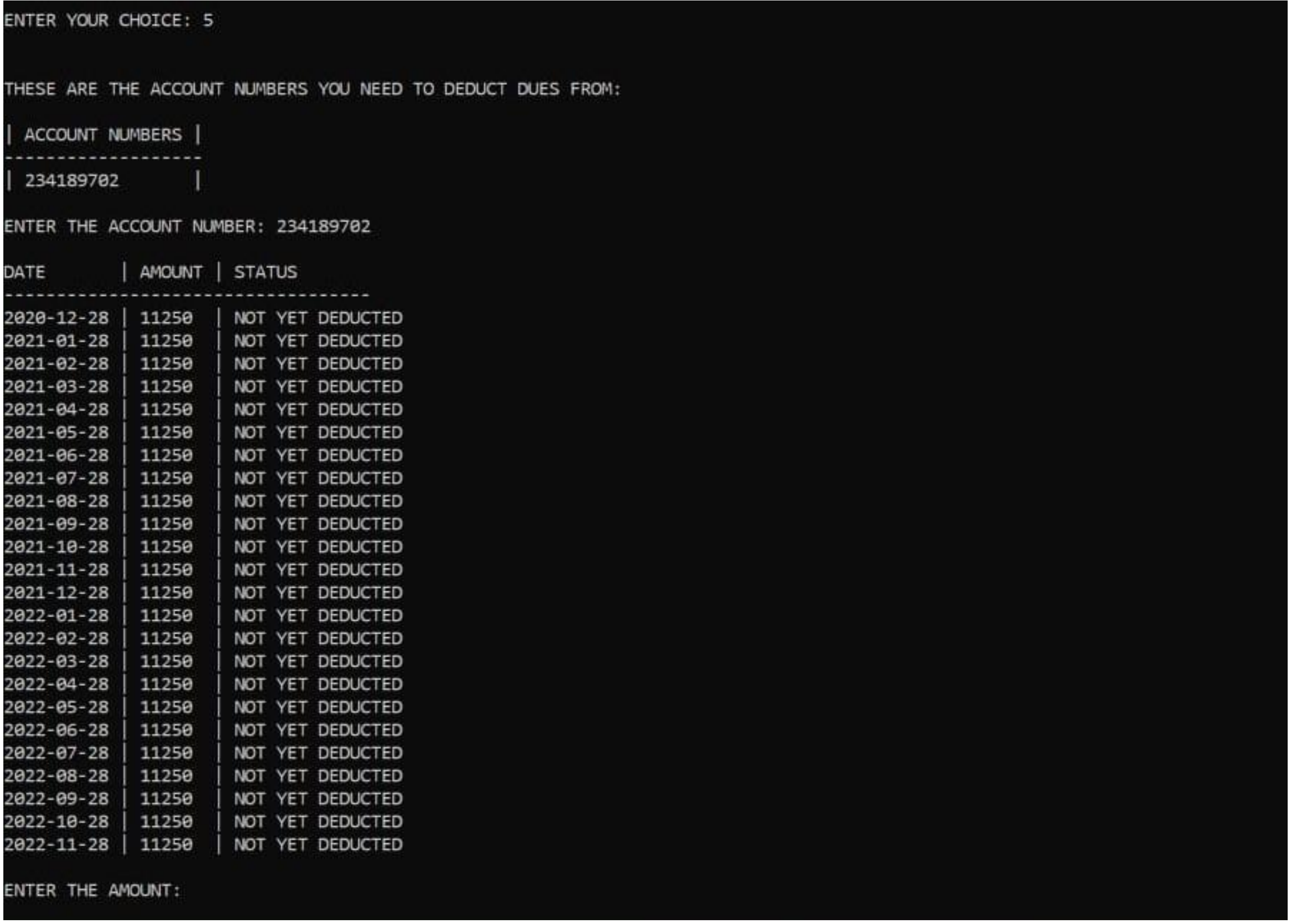
```
ENTER YOUR CHOICE: 2


ENTER USERNAME, OR B TO GO BACK: pranav@e1

ENTER PASSWORD, OR ENTER B GO BACK: **********


WELCOME, PRANAV

1. CHECK USER DETAILS
2. CHAECK USER TRANSACTIONS
3. MODIFY RATES
4. SANCTION LOANS
5. DEDUCT LOAN DUES
6. BACK


ENTER YOUR CHOICE: 3


ENTER THE LOAN CODE: vc09

ENTER THE NEW RATE: 9

THE LOAN RATE OF VEHICLE LOAN HAS BEEN SUCCESSFULLY CHANGED TO 9 PERCENT.

ENTER N TO MODIFY RATES FURTHER, OR ANY KEY TO GO BACK:
```

4 . Sanctioning loans:



```
ENTER YOUR CHOICE: 3


ENTER THE LOAN CODE: vc09

ENTER THE NEW RATE: 9

THE LOAN RATE OF VEHICLE LOAN HAS BEEN SUCCESSFULLY CHANGED TO 9 PERCENT.

ENTER N TO MODIFY RATES FURTHER, OR ANY KEY TO GO BACK:


1. CHECK USER DETAILS
2. CHAECK USER TRANSACTIONS
3. MODIFY RATES
4. SANCTION LOANS
5. DEDUCT LOAN DUES
6. BACK


ENTER YOUR CHOICE: 4


THESE ARE THE PENDING LOAN REQUESTS YOU HAVE TO SANCTION.

ACCOUNT NUMBER | NAME                 | AMOUNT     | TIME(IN YEARS) | LOAN NAME      | STATUS
------------------------------------------------------------------------------------------------
234189702      | PRANAV               | 500000     | 2              | VEHICLE LOAN   | PENDING

ENTER THE ACCOUNT NUMBER OF THE USER YOU WANT TO SANCTION THE LOAN: 234189702

THE LOAN HAS BEEN SUCCESSFULLY SANCTIONED.

ENTER N TO SANCTION OTHER LOANS, OR ANY KEY TO GO BACK: _
```

5. Deducting amount from accounts of the users having loan:



```
ENTER YOUR CHOICE: 5


THESE ARE THE ACCOUNT NUMBERS YOU NEED TO DEDUCT DUES FROM:

| ACCOUNT NUMBERS |
-------------------
| 234189702       |

ENTER THE ACCOUNT NUMBER: 234189702

DATE       | AMOUNT | STATUS
-----------------------------------
2020-12-28 | 11250  | NOT YET DEDUCTED
2021-01-28 | 11250  | NOT YET DEDUCTED
2021-02-28 | 11250  | NOT YET DEDUCTED
2021-03-28 | 11250  | NOT YET DEDUCTED
2021-04-28 | 11250  | NOT YET DEDUCTED
2021-05-28 | 11250  | NOT YET DEDUCTED
2021-06-28 | 11250  | NOT YET DEDUCTED
2021-07-28 | 11250  | NOT YET DEDUCTED
2021-08-28 | 11250  | NOT YET DEDUCTED
2021-09-28 | 11250  | NOT YET DEDUCTED
2021-10-28 | 11250  | NOT YET DEDUCTED
2021-11-28 | 11250  | NOT YET DEDUCTED
2021-12-28 | 11250  | NOT YET DEDUCTED
2022-01-28 | 11250  | NOT YET DEDUCTED
2022-02-28 | 11250  | NOT YET DEDUCTED
2022-03-28 | 11250  | NOT YET DEDUCTED
2022-04-28 | 11250  | NOT YET DEDUCTED
2022-05-28 | 11250  | NOT YET DEDUCTED
2022-06-28 | 11250  | NOT YET DEDUCTED
2022-07-28 | 11250  | NOT YET DEDUCTED
2022-08-28 | 11250  | NOT YET DEDUCTED
2022-09-28 | 11250  | NOT YET DEDUCTED
2022-10-28 | 11250  | NOT YET DEDUCTED
2022-11-28 | 11250  | NOT YET DEDUCTED

ENTER THE AMOUNT:
```

LIMITATIONS

- DUE TO THE USE OF gTTS MODULE THE PROGRAM REQUIRES THE ACCESS TO INTERNET DURING IT'S EXECUTION.

- THOUGH HAVING ALL THE FUNCTION -ALITIES , IT CAN'T BE READILY USED FOR A BANK BECAUSE IT CAN'T CHECK IF THE PROVIDED INFORMATION LIKE ADHAAR NUMBER,PAN NU MBER ,TEL - EPHONE NUMBERS ARE AUTHENTIC .