```
#for the interface of the application
from tkinter import
from tkinter import ttk
#for graphing functions
import matplotlib.pyplot as plt
import numpy as np
#for database connectivity
import mysql.connector
#for organising data
import pandas as pd
#for DD code generation
from random import randint

mydb = mysql.connector.connect(
  host="localhost",
  user="root",
  password="avani2005",
  database="bank"
)

mycursor = mydb.cursor(buffered=True)

root = Tk()
root.geometry('600x400')

def ClearPrevFrame():
  #deleteing previous register frames
```

```
  for i in RegisterFrame.winfo_children():
    i.destroy()

  #Deleting previous loginframes
  for i in LoginFrame.winfo_children():
    i.destroy()

  #unpacking unnesscry frames
  RegisterFrame.pack_forget()
  LoginFrame.pack_forget()

def TransactionWindow():
  #Creating new window usint toplevel
  TrWindow = Toplevel(root)

  #declaring title and geometry
  TrWindow.title('Transaction History and New Transaction')
  TrWindow.geometry('300x300')

  #Cclear transaction frames clear the frames in teh transaction windows
  def ClearTransactionFrames():

    for i in MakeTrFrame.winfo_children():
      i.destroy()
```

```
    for i in TrHistoryFrame.winfo_children():
      i.destroy()

    MakeTrFrame.pack_forget()
    TrHistoryFrame.pack_forget()

  MakeTrFrame = Frame(TrWindow,bg = '#6BA058')

  def MakeTrPage():

    ClearTransactionFrames()

    MakeTrFrame.place(width=200,height=200)
    MakeTrFrame.pack(fill='both',expand=1)

trlabel = Label(MakeTrFrame,text = 'TRANSACTION MAKING PAGE',bg
 ='#6BA058')
    AUserAccNoLabel = Label(MakeTrFrame,text = 'Enter Your account Number
 -',bg ='#6BA058')
    BUserPhoneNoLabel = Label(MakeTrFrame,text='Phone Number of receiver
 -',bg ='#6BA058')
      AmmountLabel = Label(MakeTrFrame,text='Ammount to be sent-',bg
 ='#6BA058')

    AUserAccNoEntry = Entry(MakeTrFrame)
    BUserPhoneEntry = Entry(MakeTrFrame)
    AmmountEntry = Entry(MakeTrFrame)
```

```
    trlabel.place(x=50,y=150)
    AUserAccNoLabel.place(x=20,y=180)
    BUserPhoneNoLabel.place(x=20,y=200)
    AmmountLabel.place(x=20,y=220)
    AUserAccNoEntry.place(x=180,y=180)
    BUserPhoneEntry.place(x=180,y=200)
    AmmountEntry.place(x=180,y=220)

    def makeTrQuery():
                    Query1 = f'INSERT INTO transactions VALUES(\'{AUserAccNoEntry.get()}\',{BUserPhoneEntry.get()},{AmmountEntry.get()});'
        Query2 = f'UPDATE accounts SET BankBalance = BankBalance - {AmmountEntry.get()} WHERE AccountNumber = \'{AUserAccNoEntry.get()}\';'
        Query3 = f'UPDATE accounts SET BankBalance = BankBalance + {AmmountEntry.get()} WHERE PhoneNumber = {BUserPhoneEntry.get()};'

      mycursor.execute(Query1)
      mycursor.execute(Query2)
      mycursor.execute(Query3)

      label = Label(MakeTrFrame,text = 'transaction submitted sucessfuly')
      label.place(x=180,y=280)
```

```
        TransactButton = Button(MakeTrFrame,text = 'Submit',command=makeTrQuery)
    TransactButton.place(x=180,y=260)


  TrHistoryFrame = Frame(TrWindow,bg = '#304476')


  def TrHistoryPage():

    ClearTransactionFrames()


    TrHistoryFrame.place(width=200,height=200)
    TrHistoryFrame.pack(fill='both',expand=1)


    def GetTrData():

      #condition to make sure there is no cross checking of account transactions
        checkQuery = f'SELECT AccountNumber FROM accounts WHERE Password = {PasswordEntry.get()}'
      mycursor.execute(checkQuery)
      userdata = mycursor.fetchone()


      if userdata[0] == AccNoEntry.get():
```

```
Query = f'SELECT ReceiverPhone,Amount FROM transactions WHERE
SenderAccNo = {AccNoEntry.get()};'
        mycursor.execute(Query)
        data = mycursor.fetchall()

        Phonedata = []
        AmmountData = []
       for i in data:

         Phonedata.append(i[0])
        AmmountData.append(i[1])
        print(Phonedata,AmmountData)

        DictData = {'ReciverPhone':Phonedata,
                'Ammount':AmmountData
                }
        df = pd.DataFrame(DictData)

        DataLabel = Label(TrHistoryFrame,text = df,bg = '#304476')
        DataLabel.place(x=30,y=150)

hisLabel = Label(TrHistoryFrame,text = 'Check Your transaction History
here',bg='#304476',fg='#FFFFFF')
    AccNoLabel = Label(TrHistoryFrame,text ='Account Number',bg = '#304476')
    Passwordlabel = Label(TrHistoryFrame,text = 'Password',bg = '#304476'
```

```
AccNoEntry = Entry(TrHistoryFrame) PasswordEntry =
Entry(TrHistoryFrame,show = '*')


    hisLabel.place(x=30,y=70)
    AccNoLabel.place(x=40,y=90)
    Passwordlabel.place(x=40,y=110)
    AccNoEntry.place(x=120,y=90)
    PasswordEntry.place(x=90,y=110)


    ShowHistButton = Button(TrHistoryFrame,text = 'Show History',command =
GetTrData)
    ShowHistButton.place(x=100,y=130)
   makTrButton = Button(TrWindow,text = 'Make Transaction', command =
MakeTrPage)
  makTrButton.place(x=20,y=20)


  TrHistButton = Button(TrWindow,text= 'History',command = TrHistoryPage)
  TrHistButton.place(x=130,y=20)


def LoanWindow(
```

```
LoanWin = Toplevel(root)
   LoanWin.title('check credit scores, Loans and other relevant details here')
   LoanWin.geometry('300x320')
   LoanFrame = Frame(LoanWin,bg='#31C651')


   def LoanPage():
     LoanFrame.place(width=200,height=200)
     LoanFrame.pack(fill='both',expand=1)

def Loan():
       def ClearPrevFrame():
          for i in LoanFrame.winfo_children():
             i.destroy()
          LoanFrame.pack_forget()
       ClearPrevFrame()


       LoanFrame.pack(fill ='both',expand = 1)

       def GraphLoan():


          x1=[]
          x2=[]
          for j in range (1,D+1):
```

```
            x1.append(j)
            x2.append((L*i*j)+L)
        x=np.array(x1)
        y=np.array(x2)
        plt.plot(x,y)
        plt.show()
Cr=700
    if Cr>200 and Cr<400:

        Flag=1

    elif Cr>400 and Cr<600:

        Flag=2

    elif Cr>600 and Cr<850:

        Flag=3

    else:

         CreditScoreLabel=Label(LoanFrame,text="Enter your credit score in the
 login/register page")
        CreditScoreLabel.place(x=10,y=70)

   #IF FLAG=1 , INTEREST RATE = 4TIMES
```

```
    #IF FLAG=2 , INTEREST RATE = 2TIMES
    #IF FLAG=3 , INTEREST RATE= NORMAL RATE

        DuratioEntryLabel=Label(LoanFrame,text="Enter the duration of loan required")
        DurationEntry=Entry(LoanFrame)
        D=DurationEntry.get()
        D=7
        if D<5:

            InterestRateLabel=Label(LoanFrame,text="Interest Rate is 10%")
            MinLoanAmtLabel=Label(LoanFrame,text="Minimum Loan Amount: Rs.50,000")
            m=50000
            i=0.1

        elif D>5 and D<10:

            InterestRateLabel=Label(LoanFrame,text="Interest Rate is 6%")
            MinLoanAmtLabel=Label(LoanFrame,text="Minimum Loan Amount is: Rs.2,00,000")
            m=200000
            i=0.06

        elif D>10 and D<20:
```

```
        InterestRateLabel=Label(LoanFrame,text="Interest Rate is 3%")
        MinLoanAmtLabel=Label(LoanFrame,text="Minimum Loan Amount is: Rs.10,00,000")
        m=1000000
        i=0.03

    else:

                        DLabel=Label(LoanFrame,text="Loan duration not available").place(x=10,y=170)
        DLabel.place(x=10,y=170)

    LoanAmtEntryLabel=Label(LoanFrame,text="Enter the loan amount")
    LoanAmtEntry=Entry(LoanFrame)
    L=LoanAmtEntry.get()
    L=1000000
    if L>=m:
        if Flag==1:
            i=i*4
        if Flag==2:
            i=i*2
        S=(L*i*D)+L
        AmtToPayLabel=Label(LoanFrame,text=f"Amount to be paid at the end of term = {S}")
        AmtPerMonthLabel=Label(LoanFrame,text=f"Amount to be paid every month = {(S/(12*D))}")
    DuratioEntryLabel.place(x=10,y=100)
```

```
    InterestRateLabel.place(x=10,y=150)
    MinLoanAmtLabel.place(x=10,y=170)
    LoanAmtEntryLabel.place(x=10,y=200)
    AmtToPayLabel.place(x=10,y=250)
    AmtPerMonthLabel.place(x=10,y=270)
    DurationEntry.place(x=10,y=120)
    LoanAmtEntry.place(x=10,y=220)
                                        graphbutton=Button(LoanWin,text='CREATE
GRAPH',command=GraphLoan)
    graphbutton.place(x=20,y=300)

  Loanbutton=Button(LoanWin,text='LOANS',command=Loan)
  Loanbutton.place(x=10,y=40)
  LoanWin.mainloop()

def InvestWindow():
  def ClearInvestFrame():
    for i in mainframe.winfo_children():
      i.destroy()
    mainframe.pack_forget()
  InvestPage=Toplevel(root)
  InvestPage.geometry('500x500')
  mainframe=Frame(InvestPage,bg='blue')
  def fd():
    ClearInvestFrame()
```

```
mainframe.pack(fill ='both',expand = 1)
def GraphFD():
    dep=1000000
    y=5
    for i in range(1,y+1):
        fv=dep*((100+i)/(100))*n
        l1.append(i)
        l2.append(fv)
    x=np.array(l1)
    y=np.array(l2)
    plt.plot(x, y)
    plt.show()
def intrest(n,seniority):
    if n>=1 and n<=3:
        i=1.5
    elif n>3 and n<=5:
        i=2
    elif n>5 and n<=10:
        i=3
    else:
        i=5
    if seniority=='Y':
        i=i%2
    InterestLabel=Label(mainframe,text=f'Intrest Offered is {i}%' )
    return (i)
    InterestLabel.place(x=20,y=70)
```

```
def sen(age):
    if age>=60:
        return('Y')
    else:
        return('NO')


def premature(ynew,age):
    dep=1000000
      NoteLabel=Label(mainframe,text='PLEASE NOTE ON PREMATURE WITHDRAWAL , YOUR INTREST RATE WILL BE DEMOTED TO THE LOWER BRANCH')
    seniority=sen(age)
    i=intrest(ynew,seniority)


    fv=dep*((100+i)/(100))*ynew
    gain=(dep*i)/(100)
    NewInterestLabel=Label(mainframe,text=f'At the end of {ynew} years you will recive gain and final value as {gain} {fv}')
    NoteLabel.place(x=20,y=175)
    NewInterestLabel.place(x=20,y=225)
y=5
age=50
dep=1000000
n=y
        YearsLabel=Label(mainframe,text=f'FIXED DEPOSIT FOR YEARS={y}').place(x=20,y=50)
seniority=sen(age)
```

```
    i=intrest(n,seniority)
DepositLabel=Label(mainframe,text='Enter the deposit amount')
DepositEntry=Entry(mainframe,text='Enter Deposit') fv=dep*((100+i)/(100))*n
    gain=(dep*i)/(100)
     FinalValueLabel=Label(mainframe,text=f'At the end of {n} years you will recive gain and final value as {gain} {fv}')
    l1=[]
    l2=[]
       CheckEntryLabel=Label(mainframe,text='Would you like a premature Withdrawal (Yes/No)')
    CheckEntry=Entry(mainframe)
    if CheckEntry.get()=='Yes':
       NewDurationEntry=Entry(mainframe,text='Enter NEW DURATION')
       NewDurationEntry.place(x=20,y=200)
       premature(NewDurationEntry.get(),age)
    FinalValueLabel.place(x=20,y=350)
    DepositLabel.place(x=20,y=125)
    CheckEntryLabel.place(x=20,y=175)
    DepositEntry.place(x=20,y=150)
    CheckEntry.place(x=20,y=200)
                                    graphbutton=Button(mainframe,text='CREATE GRAPH',command=GraphFD)
```

```
    graphbutton.place(x=20,y=400)


  fdbutton=Button(InvestPage,text='FIXED DEPOSITS',command=fd)
  fdbutton.place(x=20,y=10)

RegisterFrame = Frame(root,bg = 'light blue')
def RegisterPage():
  ClearPrevFrame()
  RegisterFrame.place(width = 200,height = 300)
  RegisterFrame.pack(fill = 'both',expand=1)

  AccountNumber = Entry(RegisterFrame)
  Password = Entry(RegisterFrame,show = '*')
  Name = Entry(RegisterFrame)
  PhoneNumber = Entry(RegisterFrame)
  Age = Entry(RegisterFrame)
  Gender = Entry(RegisterFrame)
  Occupation = Entry(RegisterFrame)
  Address = Entry(RegisterFrame)
  BankBalance = Entry(RegisterFrame)

  AccountNumber.place(x = 160,y=40)
  Password.place(x = 150,y=60)
  Name.place(x = 150,y=80)
  PhoneNumber.place(x = 150,y=100)
```

```
Age.place(x = 150,y=120)
Gender.place(x = 150,y=140)
Occupation.place(x = 150,y=160)
Address.place(x = 150,y=180)
BankBalance.place(x = 170,y=200)

#LABELS
LAccountnumber=Label(RegisterFrame, text="Account Number - ",bg='light blue' )
LPassword=Label(RegisterFrame, text="Password - ",bg='light blue')
LName=Label(RegisterFrame, text="Name - ",bg='light blue')
LPhoneNumber=Label(RegisterFrame, text="Phone Number - ",bg='light blue')
Lage=Label(RegisterFrame, text="Age - ",bg='light blue')
LGender=Label(RegisterFrame, text="Gender - ",bg='light blue')
LOccupation=Label(RegisterFrame, text="Ocuupation - ",bg='light blue')
LAddress=Label(RegisterFrame, text="Address - ",bg='light blue')
LBankBalance= Label(RegisterFrame,text = 'Initial Bank deposit',bg='light blue')

LAccountnumber.place(x = 50,y=40)
LPassword.place(x = 50,y=60)
LName.place(x = 50,y=80)
LPhoneNumber.place(x = 50,y=100)
Lage.place(x = 50,y=120)
LGender.place(x = 50,y=140)
LOccupation.place(x = 50,y=160)
LAddress.place(x = 50,y=180)
```

```
  LBankBalance.place(x = 50,y=200)
  def DataDump():

                                                                nonlocal
AccountNumber,Password,Name,PhoneNumber,Age,Gender,Occupation,Addre
ss,BankBalance
                                                        data =
(AccountNumber.get(),Password.get(),Name.get(),PhoneNumber.get(),Age.get(
),Gender.get(),Occupation.get(),Address.get(),BankBalance.get())
    Query = 'INSERT INTO accounts VALUES(%s,%s,%s,%s,%s,%s,%s,%s,%s)'
     mycursor.execute(Query,data)
     mydb.commit()
      DataLabel = Label(RegisterFrame,text = 'Data submitted sucessfully',bg =
'light blue')
     DataLabel.place(x=60,y=250)

      SubmitData = Button(RegisterFrame,text = 'submit',command =
DataDump).place(x = 75,y = 220)

def IncomeTaxReturnPage():
  ITRWindow=Toplevel(root)
  ITRWindow.geometry('450x300')
  ITR= Frame(ITRWindow,bg='light green')
```

```
ITR.pack(fill='both',expand=1)

    gtr = Entry(ITR)
    dfc = Entry(ITR)
    toi = Entry(ITR)
    cyl = Entry(ITR)
    ntp = Entry(ITR)
    ifp = Entry(ITR)
    tfie = Entry(ITR)
    at = Entry(ITR)
    ei = Entry(ITR)

    gtr.place(x = 260,y=40)
    dfc.place(x = 250,y=60)
    toi.place(x = 250,y=80)
    cyl.place(x = 250,y=100)
    ntp.place(x = 250,y=120)
    ifp.place(x = 250,y=140)
    tfie.place(x = 250,y=160)
    at.place(x = 250,y=180)
    ei.place(x = 270,y=200)

    Lgtr=Label(ITR, text="Gross Total Income - ",bg='light green' )
    Ldfc=Label(ITR, text='Deductions from Chapter 6a-',bg='light green')
```

```
  Ltoi=Label(ITR, text='Total Income-',bg='light green')
  Lcyl=Label(ITR, text="Current Year Loss",bg='light green')
  Lntp=Label(ITR, text='Net Tax Payable',bg='light green')
  Lifp=Label(ITR, text='Intrest and FEE payable',bg='light green')
   Ltfie=Label(ITR, text='TOTAL TAX AND INTREST PAYABLE',bg='light green')
  Lat=Label(ITR, text='ADVANCED TAX',bg='light green')
  Lei=Label(ITR,text='EXempete Income',bg='light green')


  Lgtr.place(x = 50,y=40)
  Ldfc.place(x = 50,y=60)
  Ltoi.place(x = 50,y=80)
  Lcyl.place(x = 50,y=100)
  Lntp.place(x = 50,y=120)
  Lifp.place(x = 50,y=140)
  Ltfie.place(x = 50,y=160)
  Lat.place(x = 50,y=180)
  Lei.place(x = 50,y=200)


  def ITRData():
    a = gtr.get()
    b = dfc.get()
    c = toi.get()
    d = cyl.get()
    e = ntp.get()
    f = ifp.get()
```

```
    g = tfie.get()
    h = at.get()
    i = ei.get()

    #File handling with income tax returns
    fi = open('incometaxReturns.txt','w')
    fi.write(f'Gross total income is {a}\n')
    fi.write(f'Deductions from Chapter 6a is {b}\n')
    fi.write(f'total income is {c}\n')
    fi.write(f'Current year losses {d}\n')
    fi.write(f'Net tax payable {e}\n')
    fi.write(f'Intrest and Fee payable {f}\n')
    fi.write(f'Total tax and fee payable{g}\n')
    fi.write(f'Advance tax {h}\n')
    fi.write(f'Exempted Income (i.e agriculture and others){i}\n')

    fi.close()
fi = open('incometaxReturns.txt','r')arr
= fi.readlines()
    df =  pd.DataFrame(arr)
    print(df)
    fi.close()

  checkItrButton = Button(ITR,text='check ITR details', command = ITRData)
  checkItrButton.place(x=50,y=220)
```

```
def DemandDraftWithdrawal():

DemandDWin = Toplevel(root)
DemandDWin.geometry('350x300')
DemandDWin.title('')

   DDFrame = Frame(DemandDWin,bg = 'yellow')
   DDFrame.pack(fill = 'both',expand =1)


   #Labels
   Label1 = Label(DDFrame,text = 'Claim a Demand Draft here',bg='yellow')
   LSender = Label(DDFrame,text ='Your Name',bg='yellow')
   LReceiver= Label(DDFrame,text = 'Reciver',bg='yellow')
   Lammount = Label(DDFrame,text = 'Ammount',bg='yellow')
   LammountWords = Label(DDFrame,text = 'Ammount words',bg='yellow')


   #Entries
   Sender = Entry(DDFrame)
   Reciver = Entry(DDFrame)
   Ammount = Entry(DDFrame)
   AmmountWords = Entry(DDFrame)
```

```
def DDFiller():

  DDcode = ''

  for i in range(10):

    DDcode += str(randint(0,10))

  #entry Data
  a = Sender.get()
  b = Reciver.get()
  d = Ammount.get()
  e = AmmountWords.get()

  #Demand Draft file
  file = open('DemandDraft.txt','w')

  file.write('DEMAND DRAFT\n')
  file.write('_____\n')
  file.write(f'Issued By {a}')
  file.write('_____\n')
  file.write('_____\n')
  file.write(f'ON DEMAND PAY {b}\n')
  file.write(f'AMOUNT(in words) {e}||')
  file.write(f'AMOUNT={d}||\n')
  file.write('_____\n')
```

```
        file.write('_____\n')
        file.write('ISSUING BANK: AAS\n')
        file.write('ISSUING BRANCH: VELACHERY \n')
        file.write(f'Demand Draft Code :{DDcode}')
    file.close()

    #Placing Labels and ENtries
    Label1.place(x=90,y=10)
    LSender.place(x=40,y=30)
    LReceiver.place(x=40,y=60)
    Lammount.place(x=40,y=90)
    LammountWords.place(x=40,y=120)

    Sender.place(x=150,y=30)
    Reciver.place(x=150,y=60)
    Ammount.place(x=150,y=90)
    AmmountWords.place(x=150,y=120)

    #Buttons
    CreateDD = Button(DDFrame,text = 'Create DD',command = DDFiller)
    CreateDD.place(x=60,y=140)

  LoginFrame = Frame(root,bg = 'orange')
```

```
def Loginpage():
  def LoginData():


    nonlocal AccNoEntry, PasswordEntry
                                    query = f'SELECT Name,PhoneNumber,Age,Gender,Occupation,Address,BankBalance FROM accounts WHERE AccountNumber LIKE \'{AccNoEntry.get()}\' AND Password LIKE \'{PasswordEntry.get()}\';'
    mycursor.execute(query)
    data = mycursor.fetchall()


    #Destructuring the data
    Name,phno,age,sex,occu,addres,bal=data[0]


    #Identifier Labels
    NameLabel=Label(LoginFrame,text ='UserName',bg='orange')
    PhoneLabel = Label(LoginFrame,text='Phone Number',bg='orange')
    AgeLabel = Label(LoginFrame,text = 'Age',bg='orange')
    GenderLabel = Label(LoginFrame,text = 'Gender',bg = 'orange')
    OccupationLabel = Label(LoginFrame,text = 'Occupation',bg='orange')
    AddresLabel = Label(LoginFrame,text='Address',bg='orange')
    BalanceLabel = Label(LoginFrame,text = 'Bank Balance',bg='orange')


    #DataLabels
    DNamelabel = Label(LoginFrame,text = Name ,bg='orange')
    DPhoneLabel = Label(LoginFrame,text = phno ,bg='orange')
    DAgeLabel = Label(LoginFrame,text = age ,bg='orange')
```

```
    DGenderLabel = Label(LoginFrame,text = sex ,bg='orange')
    DOccupationLabel = Label(LoginFrame,text = occu ,bg='orange')
    DAddressLabel = Label(LoginFrame,text = addres ,bg='orange')
    DBalanceLabel = Label(LoginFrame,text = bal,bg='orange')

    #Label Positioning
    NameLabel.place(x=250,y=200)
    PhoneLabel.place(x=250,y=220)
    AgeLabel.place(x=250,y=240)
    GenderLabel.place(x=250,y=260)
    OccupationLabel.place(x=250,y=280)
    AddresLabel.place(x = 250,y=300)
    BalanceLabel.place(x=250,y=320)

    DNamelabel.place(x= 350,y=200)
    DPhoneLabel.place(x=350,y=220)
    DAgeLabel.place(x=350,y=240)
    DGenderLabel.place(x=350,y=260)
    DOccupationLabel.place(x=350,y=280)
    DAddressLabel.place(x=350,y=300)
    DBalanceLabel.place(x=350,y=320)

                DDcreation = Button(LoginFrame,text='Demand
Draft',bg='orange',command=DemandDraftWithdrawal)
    ITreturns = Button(LoginFrame,text = 'File IT-returns',bg = 'orange',command
= IncomeTaxReturnPage)
```

```
                                                    Transac =
Button(LoginFrame,text='Transactions',bg='orange',command=TransactionWin
dow)

    FixedD = Button(LoginFrame,text='Fixed Deposits',bg='orange',command =
InvestWindow)

          Loan = Button(LoginFrame,text='Loans',bg='orange',command =
LoanWindow)
    DDcreation.place(x=450,y=200)

  ITreturns.place(x=450,y=240)

  Transac.place(x=450,y=280)

  FixedD.place(x=450,y=320)

  Loan.place(x=450,y=360)

  ClearPrevFrame()


  LoginFrame.pack(fill='both',expand =1)

  LoginFrame.pack(expand = 1,fill = 'both')

    accNoLabel = Label(LoginFrame,text='Enter your accountnumber',bg =
'orange').place(x=50,y=40)

    PasswordLabel = Label(LoginFrame,text = 'Enter Your Password',bg =
'orange').place(x=50,y=80)

  AccNoEntry = Entry(LoginFrame)

  PasswordEntry = Entry(LoginFrame,show = '*')

  AccNoEntry.place(x=200,y=40)

  PasswordEntry.place(x=200,y=80)

     SubmitButton = Button(LoginFrame,text ='Sign In',command =
LoginData).place(x =80,y =150)
```

```
def AboutPage():

  AboutWin = Toplevel(root)

  AboutWin.geometry('650x500')

  AboutWin.title('About Us')


  AboutFrame = Frame(AboutWin,bg='#82EC99')

  AboutFrame.pack(fill = 'both',expand = 1)


  AboutLabel1 = Label(AboutFrame,text = '"This Bank database system was created by Avanindhra, Aswath and Srikar

  It demonstrates Various features of mobile banking and the functioning of an actual bank database with the help of

  MySQL, MySQL Connector module, Numpy Module, MatPlot Library, Tkinter Module and Pandas Dataframe.

  The Application comes with features like graphing the growth of loans, fixed deposits, tracking transactions using

  an SQL database and handles multiple accounts with a master accounts table.

  Other subsequent tables such as the transactions table

  are linked to the master table using Foreign Key relations and are managed by cursor executed queries by

  MySQL Connector module.

  Numpy Module and Matplot Library handle the nessescary graphs that are shown for demonstrating loans

  and fixed deposit schemes in the bank.

  The backend of the application is managed by MySQL and using MySQL connector module

  we are able to communicate to the
```

local database. The user interface of the application is built with the help Tkinter module.

```
'',bg = '#82EC99')
AboutLabel1.place(x=10,y=20)


#Root buttons
AboutPage = Button(root,text = 'About us', command = AboutPage).place(relheight = 0.06, relwidth =0.2,relx =0.66,rely=0.15 )
RegisterPage = Button(root, text = 'Register', command = RegisterPage).place(relheight=0.06,relwidth=0.2,relx = 0.56,rely = 0.1)
Loginpage = Button(root,text = 'Login', command = Loginpage ).place(relheight=0.06,relwidth=0.2,relx= 0.75,rely=0.1)


#declaring title and the loop of the applications
root.title('AAS bank')
root.mainloop()
```