

Please specify all the below titles in separate page. Remove this line.

Update the header with your project title.

Each line must have spacing of 1.5

Use Times New Roman font for every text content

All below headings must have font size:16 and font style:Bold

Other text must have font size of 12 and justify this text in the report.

The margins should be: Left and Right– 2.00 cm, Top and Bottom – 2.00 cm

**Abstract** – Maximum 100 words about the project

## **Table of Contents**

**Introduction** (scope of the work and its importance)

**Design/Implementation : Code**

**Testing**

**Result and Analysis:** Output snapshots with proper captions

**Conclusions & future enhancements**

**References-** in standard IEEE format ([click here for the sample format](#))

## TABLE OF CONTENTS:

Sl No.	TOPIC	PAGE No.
1.	INTRODUCTION & ABSTRACT	3-4
2.	DESIGN & IMPLEMENTATION	5 - 25
3.	RESULTS AND ANALYSIS	26 – 30
4.	TESTING	31 - 32
5.	CONCLUSIONS & FUTURE ENHANCEMENTS	33
6.	REFERENCES	34

## Banking Management System

### INTRODUCTION :

"BANK MANAGEMENT SYSTEM"

The emergence of digital systems has made information to be available on finger tips. By automating the transactions, one can view the details as and when required and in no time. This project emphasizes on creation of new customer accounts, managing the existing account holders in the bank, etc. by digitalizing the system.



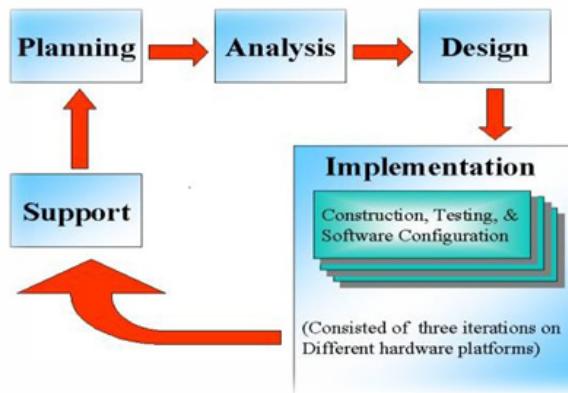
### OBJECTIVES OF THE PROJECT :

- ✓ The main objective of bank management is to build an organic and optimal interaction system between the bank and the customer.
- ✓ Creating a fully functional and interactive Graphic User Interface using tkinter.
- ✓ Implementation of various topics such as Iterables, loops, functions, real time updation of variables, concepts of global and local variables, etc.
- ✓ Creating options that allow one to create a bank account, deposit or withdraw money from it, check account balance, apply for a loan and check loan's status. In view of security, we have also added an Account PIN system without which one cannot access his/her account.
- ✓ Creating a secure system for the customers to prevent frauds.

### LEARNINGS FROM THIS PROJECT :

1. The main learning from this project was "How does a Bank function?"

2. In the technical leanings point of view we learnt and implemented several GUI features such as buttons, entries, background colours, multiple windows and error message boxes, which go beyond our current scope of syllabus.
3. We could test our skills on flow statements like if, elif, etc...



### **MENU IS DISPLAYED**

1. CREATE BANK ACCOUNT
2. MONEY WITHDRAWAL
3. MONEY DEPOSIT
4. ACCOUNT BALANCE DETAILS
5. LOAN ACOUNT CREATION
6. LOAN STATUS
7. QUIT

## DESIGN / IMPLEMENTATION:

```
from tkinter import *
from tkinter import messagebox
#data
clientName = ['Dhiru Ram', 'Bharadwaj Pranav', 'Samar Pratap', 'Rodri Martinez', 'Kevin de Bruyne', 'Younis Khan', 'Idris Alba']
clientPins = ['0001','0002', '0003','0004','0005','0006','0007']
clientBalances = [7000,9000,10000,20000,15000,2000,25000]
clientDeposition = 0
clientWithdrawal = 0
clientBalance=0
global loanAct
loanAct=['Younis Khan','Idris Alba']
global loanActpin
loanActpin = ['0006','0007']
global loanAmtlist
loanAmtlist=[10000, 5000]
global loanAmt
loanAmt=0
global interestAmtlist
interest_amt=[]
global interest
interest=0
#data
```

```
root = Tk()  
root.geometry='700x700'  
root.configure(bg = "#8ee4af")
```

```
#new account window
```

```
def clients():
```

```
    new2 = Toplevel(root)  
    new2.geometry("750x250")  
    new2.title("Client Details")
```

```
def account():
```

```
    global u  
    u = 0  
    global disk1  
    disk1 = 1  
    global disk2  
    disk2 = len(clientName)
```

```
new = Toplevel(root)
new.geometry("750x500")
new.title("New Account")
new.configure(bg = "#5cdb95")

label1 = Label(new, text="New client account", font=('Helvetica 17 bold'),
background = "#5cdb95")

label2 = Label(new, text="Please enter the name of client",background =
"#5cdb95" )

name_entry = Entry(new, font=('calibre', 10, 'normal'), width = 60)
#name=name_entry.get()

label3 = Label(new, text="Please enter a PIN to secure your account",
background = "#5cdb95")

pin_entry = Entry(new, font=('calibre', 10, 'normal'), width = 60)
#pin=pin_entry.get()

label4 = Label(new, text="Please enter the amount of Money you wish to
Deposit to start an Account", background = "#5cdb95")

amount_entry = Entry(new, font=('calibre', 10, 'normal'), width = 60)
amount_entry.insert(0, 0)
#amount=int(amount_entry.get())

def value(a, b, c):
    x = a.get()
    y = b.get()
    z = int(c.get())
    clientName.append(x)
    clientPins.append(y)
    clientBalances.append(z)
```

```
def passer():
    n = value(name_entry, pin_entry, amount_entry)

def msg1():
    messagebox.showinfo("alert", "Your account details have been successfully
entered")

    submitbutton= Button(new, text="Submit", command=lambda:[passer(),
msg1()])

label1.pack(padx = 30)
label2.pack(pady=30)
name_entry.pack()
```

```
label3.pack(pady=30)
pin_entry.pack()
label4.pack(pady=30)
amount_entry.pack()
submitbutton.pack(pady = "20")
```

```
#withdraw money window
```

```
def withdraw():
    new2 = Toplevel(root)
    new2.geometry("750x500")
    new2.title("Withdraw money")
    new2.configure(bg="#5cdb95")
```

```
    label1 = Label(new2, text="Withdraw money from your account",
font=('Helvetica 17 bold'), background = "#5cdb95")
    label1.pack(padx = 30)
```

```
#####
#####
```

```
wlabel1 = Label(new2, text="Please enter the name of client", background =
"#5cdb95")

wname_entry = Entry(new2, font=('calibre', 10, 'normal'), width=60)

wlabel2 = Label(new2, text="Please enter your PIN here", background =
"#5cdb95")

wpin_entry = Entry(new2, font=('calibre', 10, 'normal'), width=60)

def wsubmit(a, b):
    wcounter = 0
    for i in range(len(clientName)):
        if clientName[i]==a and clientPins[i]==b:
            global withdrawcount
            withdrawcount = i

            withdrawmoney()
            wcounter +=1
        if wcounter < 1:
            messagebox.showerror("warning", "the client details don't match! please
ensure that both the name and pin are correctly entered")
            wsubmitbutton= Button(new2, text = 'submit',
            command=lambda:wsubmit(wname_entry.get(), wpin_entry.get()))

def withdrawmoney():
    new3 = Toplevel(root)
    new3.geometry("750x500")
```

```
new3.title("Withdraw money")
new3.configure(bg = "#5cdb95")
wlabel03 = Label(new3, text="Your current balance is $"
"+str(clientBalances[withdrawcount]), background = "#5cdb95")
wlabel3 = Label(new3, text="Please enter the amount you want to withdraw
from your account", background = "#5cdb95")
wamount_entry = Entry(new3, font=('calibre', 10, 'normal'), width=60)
wamount_entry.insert(0, 0)

def wpasser():
    if int(wamount_entry.get()) < clientBalances[withdrawcount]:
        clientBalances[withdrawcount] -= int(wamount_entry.get())
        messagebox.showinfo(title='success', message=f'Your withdrawal was
successful. You now have ${clientBalances[withdrawcount]} in your account')
    else:
        messagebox.showerror(title="Alert", message="You don't have enough
money in your account to complete this transaction")

wamount_submit = Button(new3, text = 'submit', command = wpasser)

wlabel03.pack()
wlabel3.pack()
wamount_entry.pack()
wamount_submit.pack()
```

```
wlabel1.pack(pady = 30)
wname_entry.pack()
wlabel2.pack(pady = 30)
wpin_entry.pack()
wsubmitbutton.pack(pady= 30)
```

```
#deposit money window
```

```
def deposit():

    new3 = Toplevel(root)
    new3.geometry("750x500")
    new3.title("Deposit money")
    new3.configure(bg = "#5cdb95")

    label1 = Label(new3, text="Deposit money into your account",
    font=('Helvetica 17 bold'), background = "#5cdb95")
    label1.pack(padx=30)

    #####
    #####
    #####
```

```
dlabel1 = Label(new3, text="Please enter the name of client", background =
"#5cdb95")
dname_entry = Entry(new3, font=('calibre', 10, 'normal'), width=60)
dlabel2 = Label(new3, text="Please enter your PIN here", background =
"#5cdb95")
dpin_entry = Entry(new3, font=('calibre', 10, 'normal'), width=60)
```

```
def dsubmit(a, b):
    dcounter = 0
    for i in range(len(clientName)):
        if clientName[i] == a and clientPins[i] == b:
            global depositcount
```

```
depositcount = i
```

```
depositmoney()
dcounter += 1
if dcounter < 1:
    messagebox.showerror("warning",
        "the client details don't match! please ensure that both the
name and pin are correctly entered")
```

```
dsubmitbutton = Button(new3, text='submit', command=lambda:
dsubmit(dname_entry.get(), dpin_entry.get()))
```

```
def depositmoney():
    new4 = Toplevel(root)
    new4.geometry("750x500")
    new4.title("Deposit money")
    new4.configure(bg = "#5cdb95")
    dlabel01= Label(new4, text="Your current balance is $"
"+str(clientBalances[depositcount]), background = "#5cdb95")
    dlabel3 = Label(new4, text="Please enter the amount you want to deposit
into your account", background = "#5cdb95")
    damount_entry = Entry(new4, font=('calibre', 10, 'normal'), width=60)
    damount_entry.insert(0, 0)

def dpasser():
    clientBalances[depositcount] += int(damount_entry.get())
```

```
messagebox.showinfo(title='success', message=f"Your deposit was
successful. You now have $ {clientBalances[depositcount]} in your account")

#print(clientBalances[depositcount])

# else:

#     messagebox.showerror(title="Alert",
#                         message="You don't have enough money in your
# account to complete this transaction")

damount_submit = Button(new4, text='submit', command=dpasser)

dlabel01.pack()
dlabel3.pack()
damount_entry.pack()
damount_submit.pack()

dlabel1.pack(pady=30)
dname_entry.pack()
dlabel2.pack(pady=30)
dpin_entry.pack()
dsubmitbutton.pack(pady=30)

#check balance window
```

```
def balance():
    new5 = Toplevel(root)
    new5.geometry("750x500")
    new5.title("Check Balance")
    new5.configure(bg = "#5cdb95")

    label1 = Label(new5, text="Check your account balance", font=('Helvetica 17 bold'), background = "#5cdb95")
    label1.pack(padx = 30)

    blabel1 = Label(new5, text="Please enter the name of client", background = "#5cdb95")
    bname_entry = Entry(new5, font=('calibre', 10, 'normal'), width=60)
    blabel2 = Label(new5, text="Please enter your PIN here", background = "#5cdb95")
    bpin_entry = Entry(new5, font=('calibre', 10, 'normal'), width=60)

def bsubmit(a, b):
    bcounter = 0
    for i in range(len(clientName)):
        if clientName[i] == a and clientPins[i] == b:
            global balancecount
            balancecount = i

messagebox.showinfo(title="Balance", message=f"Your account balance is ${clientBalances[i]}")
```

```
bcounter += 1
if bcounter < 1:
    messagebox.showerror("warning",
        "the client details don't match! please ensure that both the
name and pin are correctly entered")

bsubmitbutton = Button(new5, text='submit', command=lambda:
bsubmit(bname_entry.get(), bpin_entry.get()))

#blabel1.pack()
#bamount_entry.pack()
#bamount_submit.pack()

blabel1.pack(pady=30)
bname_entry.pack()
blabel2.pack(pady=30)
bpin_entry.pack()
bsubmitbutton.pack(pady=30)
```

```
#loan window
```

```
def loan():
```

```
    new6 = Toplevel(root)
    new6.geometry("750x500")
    new6.title("Loan window")
    new6.configure(bg = "#5cdb95")
```

```
    label1 = Label(new6, text="Apply for Loan", font=('Helvetica 17 bold'),
background = "#5cdb95")
    label1.pack(padx=30)
```

```
#####
#####
```

```
    llabel1 = Label(new6, text="Please enter the name of client", background =
"#5cdb95")
    lname_entry = Entry(new6, font='calibre', 10, 'normal'), width=60)
    llabel2 = Label(new6, text="Please enter your PIN here", background =
"#5cdb95")
    lpin_entry = Entry(new6, font='calibre', 10, 'normal'), width=60)
```

```
def lsubmit(a, b):
    lcounter = 0
    for i in range(len(clientName)):
        if clientName[i] == a and clientPins[i] == b:
            global loancount
            loancount = i

    if clientName[loancount] in loanAct:
        messagebox.showerror("alert", "You have already applied for loan!
Check loan status!")
        lcounter+=1

    else:
        loanmoney()
        lcounter += 1

    if lcounter < 1:
        messagebox.showerror("warning",
                            "the client details don't match! please ensure that both the
name and pin are correctly entered")

lsubmitbutton = Button(new6, text='submit', command=lambda:
lsubmit(lname_entry.get(), lpin_entry.get()))

def loanmoney():
```

```

new7 = Toplevel(root)
new7.geometry("750x500")
new7.title("Loan Window")
new7.configure(bg = "#5cbd95")

llabel01= Label(new7, text="Your current balance is $ "+str(clientBalances[loancount]), background = "#5cdb95")

llabel02 = Label(new7, text="Our bank offers these interest rates:", background = "#5cdb95")

llabel03 = Label(new7, text="Loan Amount: 0-1000      ==> to be repaid at 2% interest rate in 1 year", background = "#5cdb95")

llabel04 = Label(new7, text="Loan Amount: 1001-10000      ==> to be repaid at 3% interest rate in 1.5 years", background = "#5cdb95")

llabel05 = Label(new7, text="Loan Amount: 10001-50000      ==> to be repaid at 4% interest rate in 2.5 years", background = "#5cdb95")

llabel06 = Label(new7, text="Loan Amount: 50001-100000      ==> to be repaid at 6% interest rate in 3.5 years", background = "#5cdb95")

llabel07 = Label(new7, text="Loan Amount: 100001 and above ==> to be repaid at 10% interest rate in 5.5 years", background = "#5cdb95")

llabel08 = Label(new7, text="", background = "#5cdb95")

llabel3 = Label(new7, text="Please enter the amount you want to apply for loan", background = "#5cdb95")

lamount_entry = Entry(new7, font=('calibre', 10, 'normal'), width=60)
lamount_entry.insert(0, 0)
#loanAmt = int(lamount_entry.get())

def lpasser():

```

```
if int(lamount_entry.get())>20*clientBalances[loancount]:  
    messagebox.showerror("Warning", "Loan amount exceeds limit")  
  
elif int(lamount_entry.get())<0:  
    messagebox.showerror("Alert", "Invalid loan amount")  
  
else:  
    clientBalances[loancount] = clientBalances[loancount] +  
        int(lamount_entry.get())  
    loanAmt=int(lamount_entry.get())  
    loanAct.append(clientName[loancount])  
    loanActpin.append(clientPins[loancount])  
    loanAmtlist.append(loanAmt)  
    messagebox.showinfo(title='success', message=f'Your loan was  
approved! You now have $ {clientBalances[loancount]} in your account')  
  
lamount_submit = Button(new7, text='submit', command=lpasser)  
  
llabel01.pack()  
llabel02.pack()  
llabel03.pack()  
llabel04.pack()
```

```
llabel05.pack()  
llabel06.pack()  
llabel07.pack()  
llabel08.pack()  
llabel3.pack()  
lamount_entry.pack()  
lamount_submit.pack()
```

```
llabel1.pack(pady=30)  
lname_entry.pack()  
llabel2.pack(pady=30)  
lpin_entry.pack()  
lsubmitbutton.pack(pady=30)
```

```
#label1 = Label(new, text="Hey, Howdy?", font=('Helvetica 17 bold'))
```

```
#loan status window
```

```
def loans():

    new8 = Toplevel(root)
    new8.geometry("750x500")
    new8.title("Loan Status")
    new8.configure(bg = "#5cdb95")

    label1 = Label(new8, text="Loan Status", font=('Helvetica 17 bold'),
background = "#5cdb95")
    label1.pack(padx = 30)

    slabel1 = Label(new8, text="Please enter the name of client", background =
"#5cdb95")
    sname_entry = Entry(new8, font=('calibre', 10, 'normal'), width=60)
    slabel2 = Label(new8, text="Please enter your PIN here", background =
"#5cdb95")
    spin_entry = Entry(new8, font=('calibre', 10, 'normal'), width=60)

def ssubmit(a, b):
    scounter = 0

    for i in range(len(clientName)):
        if clientName[i] == a and clientPins[i] == b:
            global scount1
            scount1 = i
```

```
if a not in loanAct:  
    messagebox.showerror("Warning","You have not applied for a loan")  
    scounter=1  
  
else:  
    for i in range(len(loanAct)):  
        if loanAct[i] == a:  
            global scount2  
            scount2=i  
            lstatus()  
            scounter += 1  
    if scounter < 1:  
        messagebox.showerror("warning",  
                            "the client details don't match! please ensure that both the  
                            name and pin are correctly entered")  
  
ssubmitbutton = Button(new8, text='submit', command=lambda:  
ssubmit(sname_entry.get(), spin_entry.get()))  
  
def lstatus():  
    new9 = Toplevel(root)
```

```
new9.geometry("750x500")
new9.title("Loan Status")
new9.configure(bg = "#5cdb95")
```

```
label1=Label(new9, text="Your Loan Status", font=('Helvetica 17 bold'),
background = "#5cdb95")
label1.pack(padx = 30)
```

```
label2=Label(new9, text="Your account balance is $ "
"+str(clientBalances[scount1]), background = "#5cdb95")
label2.pack()
```

```
label3=Label(new9, text="You have taken a loan of $ "
"+str(loanAmtlist[scount2]), background = "#5cdb95" )
label3.pack(padx = 30)
```

```
if loanAmtlist[scount2]>0 and loanAmtlist[scount2]<=1000:
    interest = loanAmtlist[scount2]*(0.02)
    slabel01=Label(new9, text="You are repaying your loan with 2% interes
in 1 year", background = "#5cdb95")
    slabel02=Label(new9, text="Your interest amount is $ "+str(interest),
background = "#5cdb95")
```

```
elif loanAmtlist[scount2]>1000 and loanAmtlist[scount2]<=10000:
```

```
interest = loanAmtlist[scount2]*(0.03)*1.5
slabel01=Label(new9, text="You have to repay your loan in 1.5 years, at
3% interest rate", background = "#5cdb95")
slabel02=Label(new9, text="Your interest amount is $ "+str(interest),
background = "#5cdb95")

elif loanAmtlist[scount2]>10000 and loanAmtlist[scount2]<=100000:
    interest = loanAmtlist[scount2]*(0.04)*2.5
    slabel01=Label(new9, text="You have to repay your loan in 2.5 years, at
4% interest rate", background = "#5cdb95")
    slabel02=Label(new9, text="Your interest amount is $ "+str(interest),
background = "#5cdb95")

elif loanAmtlist[scount2]>100000 and loanAmtlist[scount2]<=1000000:
    interest = loanAmtlist[scount2]*(0.06)*3.5
    slabel01=Label(new9, text="You have to repay your loan in 3.5 years, at
6% interest rate", background = "#5cdb95")
    slabel02=Label(new9, text="Your interest amount is $ "+str(interest),
background = "#5cdb95")

else:
    interest = loanAmtlist[scount2]*(0.1)*5.5
```

```
    slabel01=Label(new9, text="You have to repay your loan in 5.5 years, at  
10% interest rate", background = "#5cdb95")
```

```
    slabel02=Label(new9, text="Your interest amount is $ "+str(interest),  
background = "#5cdb95")
```

```
slabel01.pack()
```

```
slabel02.pack()
```

```
slabel1.pack(pady=30)
```

```
sname_entry.pack()
```

```
slabel2.pack(pady=30)
```

```
spin_entry.pack()
```

```
ssubmitbutton.pack(pady=30)
```

```
#main buttons here
```

```
titlelabel = Label(root, text = "Welcome to PES banking system", font =  
("Helvetica 17 bold", 30), background = "#8ee4af")  
titlelabel.grid(row = 0, column = 1, pady = 20, padx = 10)  
  
#button1  
  
newaccount = Button(root, text = "Open a new client account", font =  
("Helvetica 17 bold", 10), height = 5, command = account, background =  
"#003366", fg = "white")  
newaccount.grid(row = 1, column = 0, pady = 20)  
  
#button2  
  
withdraw = Button(root, text = "Withdraw money from your account", font =  
("Helvetica 17 bold", 10), height = 5, command = withdraw, background =  
"#003366", fg = "white")  
withdraw.grid(row = 1, column = 1, pady = 20)  
  
#button3  
  
deposit = Button(root, text = "Deposit money in your account", font =  
("Helvetica 17 bold", 10), height = 5, command = deposit, background =  
"#003366", fg = "white")  
deposit.grid(row = 1, column = 2, pady = 20)  
  
#button4  
  
checkbalance = Button(root, text = "Check your account balance", font =  
("Helvetica 17 bold", 10), height = 5, width = 30, command = balance,  
background = "#003366", fg = "white")  
checkbalance.grid(row = 2, column = 0, padx = 70, pady = 20)  
  
#button5  
  
loan = Button(root, text = "loan window", font = ("Helvetica 17 bold", 10),  
height = 5, width = 30, command = loan, background = "#003366", fg =  
"white")  
loan.grid(row = 2, column = 1, padx = 70, pady = 20)  
  
#button6
```

```
loanstatus = Button(root, text = "Check your loan status", font = ("Helvetica 17 bold", 10), height = 5, width = 30, command = loans, background = "#003366", fg = "white")

loanstatus.grid(row = 2, column = 2, pady = 20)

#button7

def quitter():

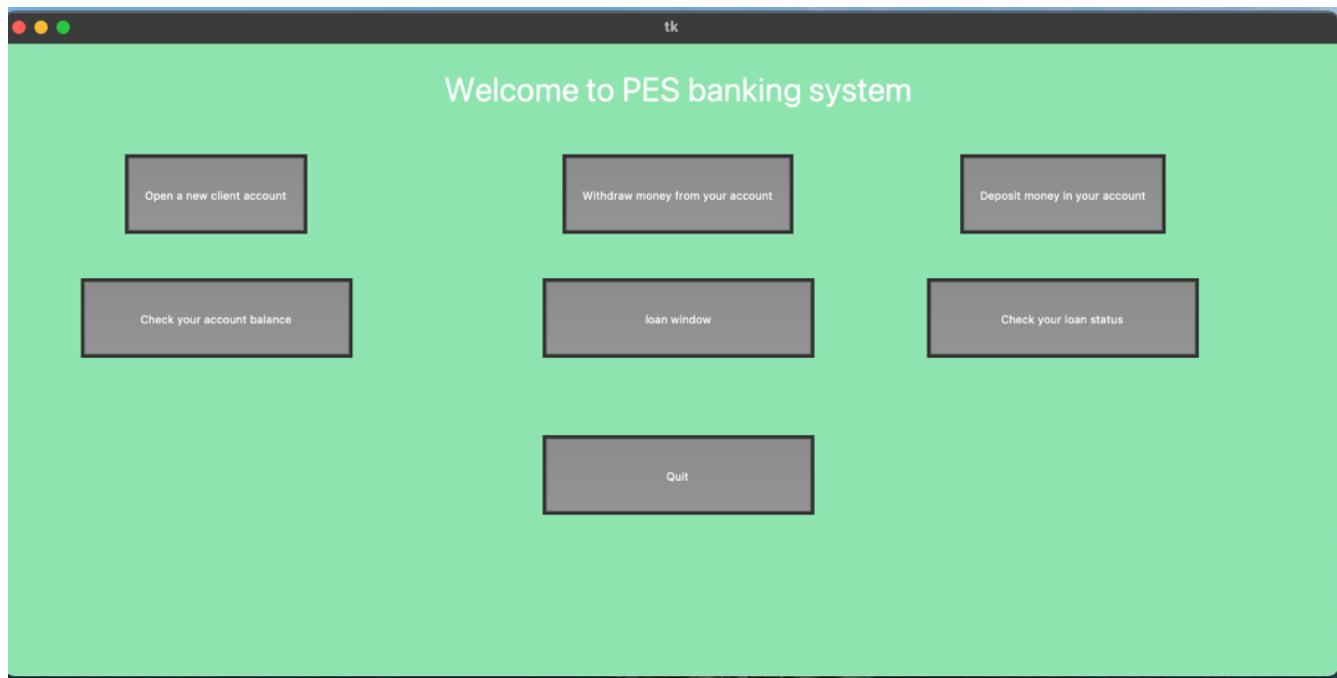
    root.destroy()

quitbank = Button(root, text = "Quit", font = ("Helvetica 17 bold", 10), height = 5, width = 30, command = quitter, background = "#003366", fg = "white")

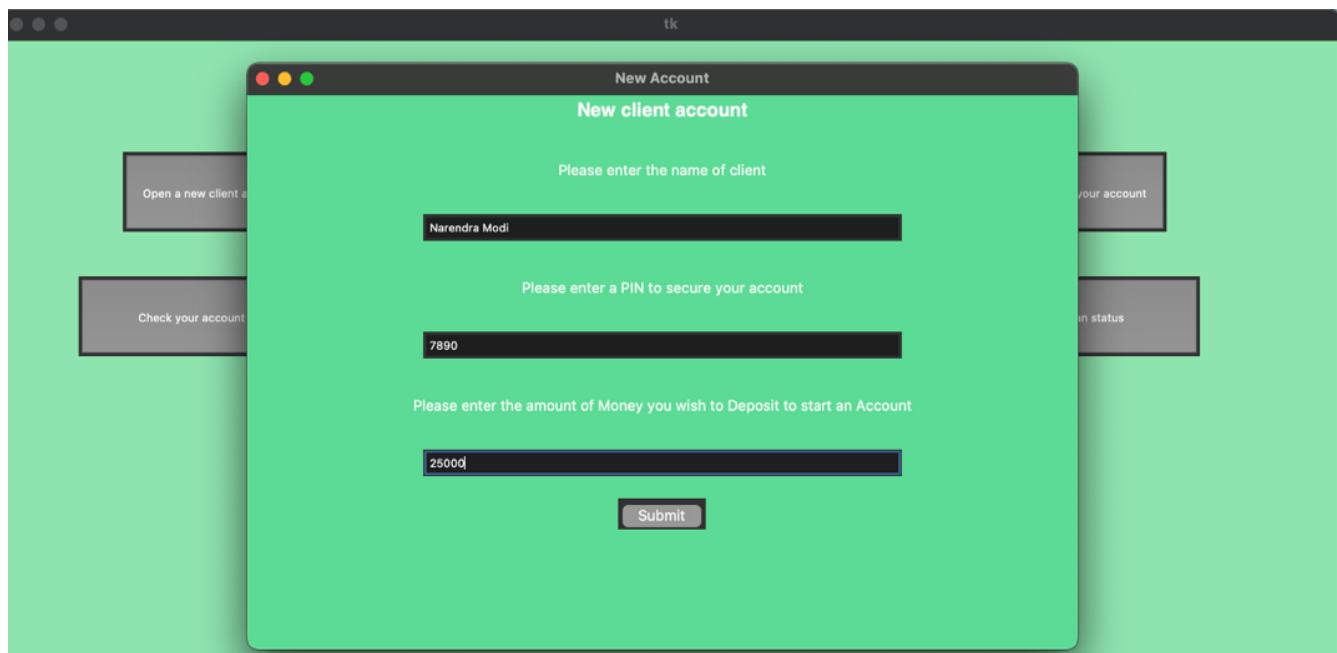
quitbank.grid(row = 3, column = 1, padx = 70, pady = 50)

root.mainloop()
```

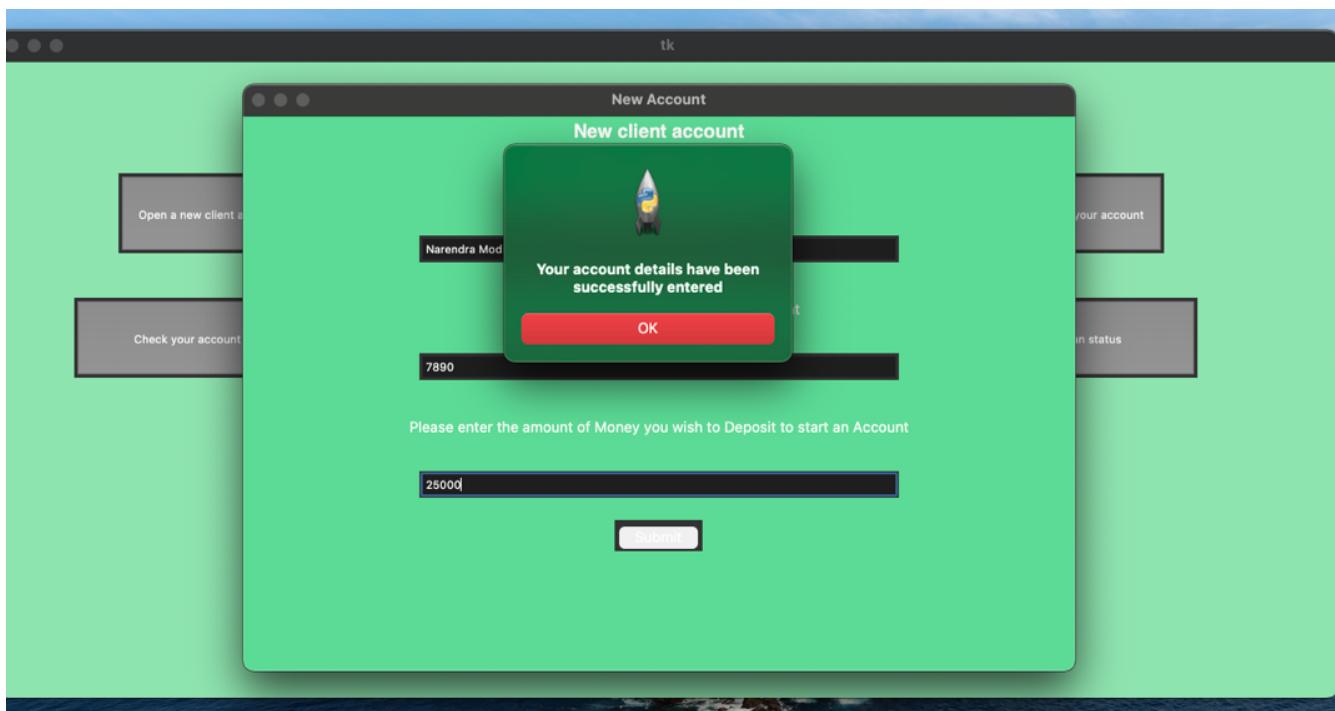
## **RESULTS AND ANALYSIS:**



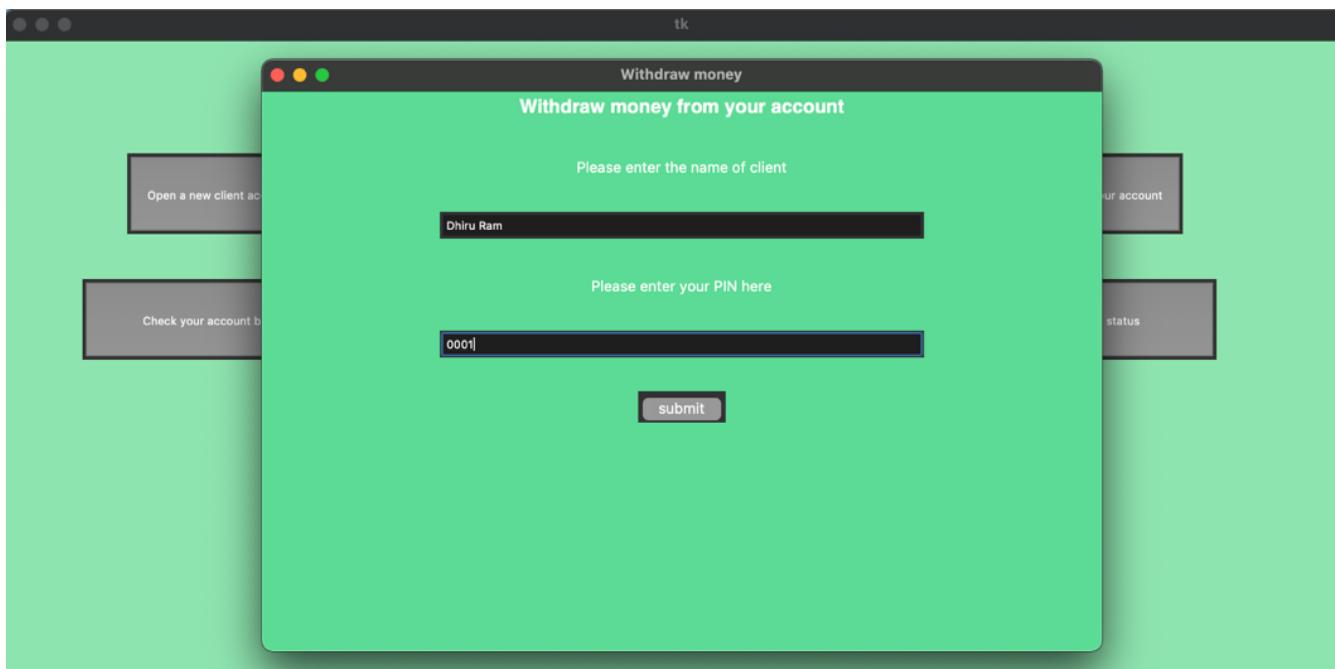
## HOME



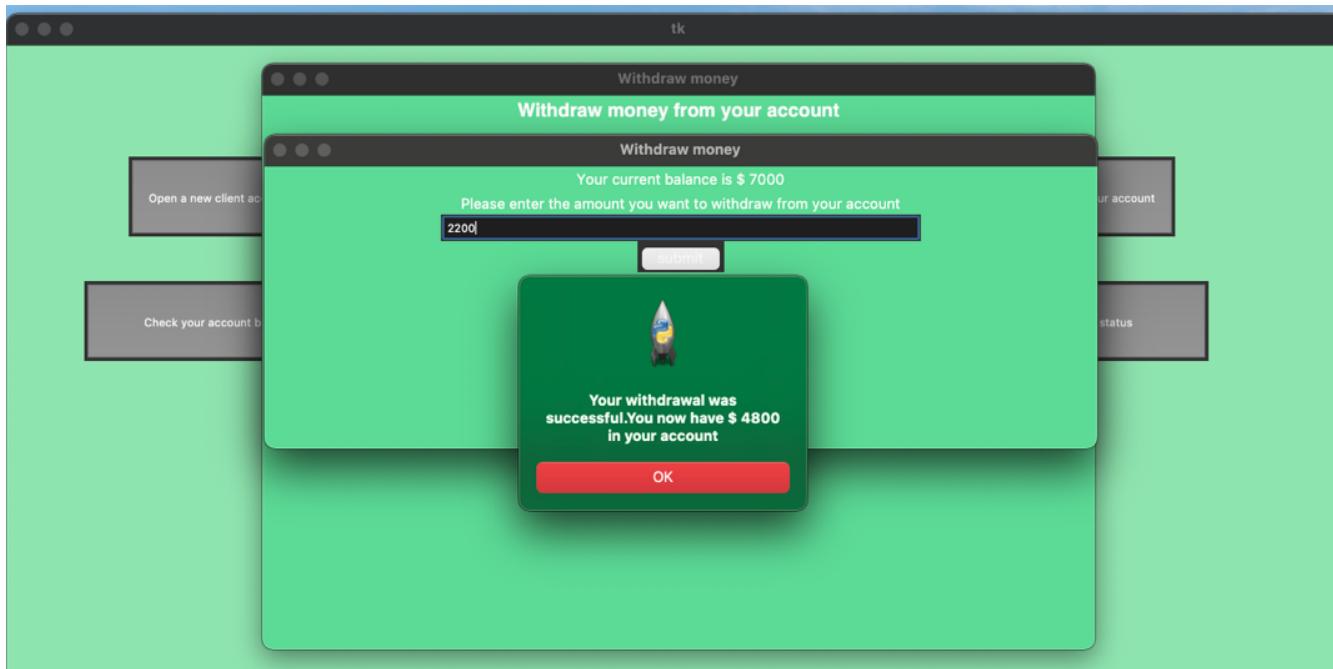
New Client Account Window



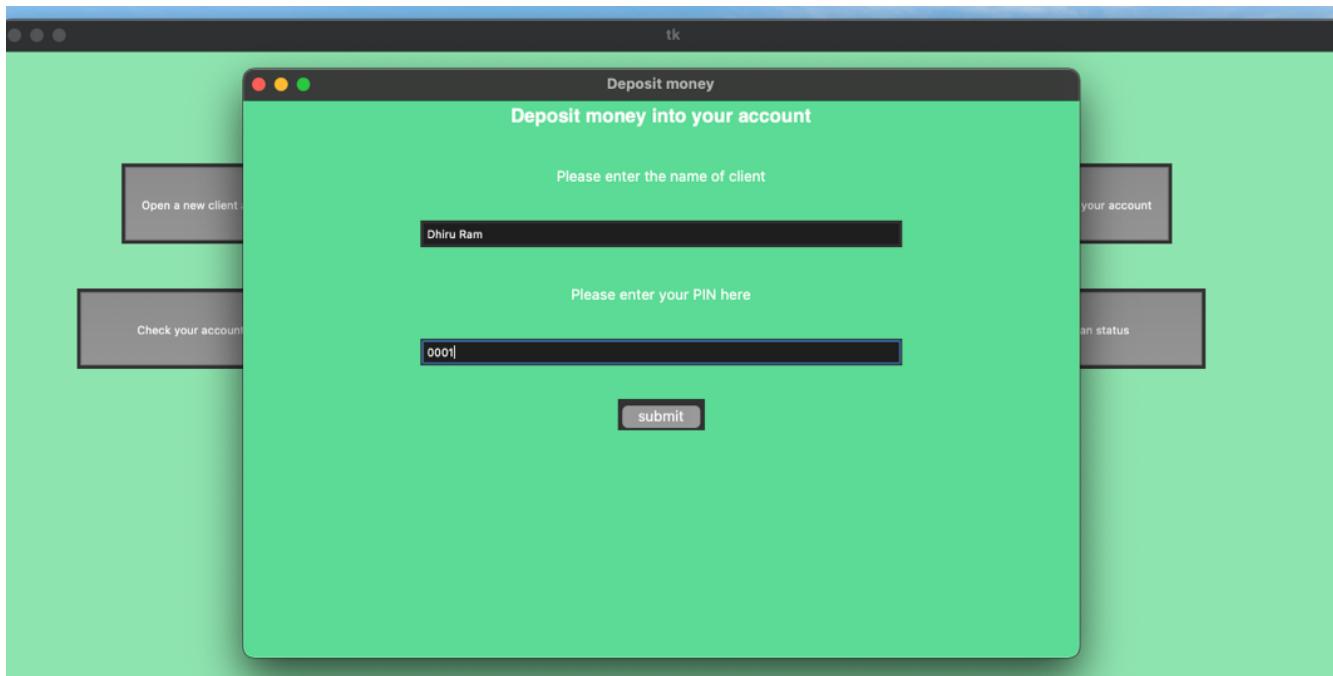
Successfully created New Client Account



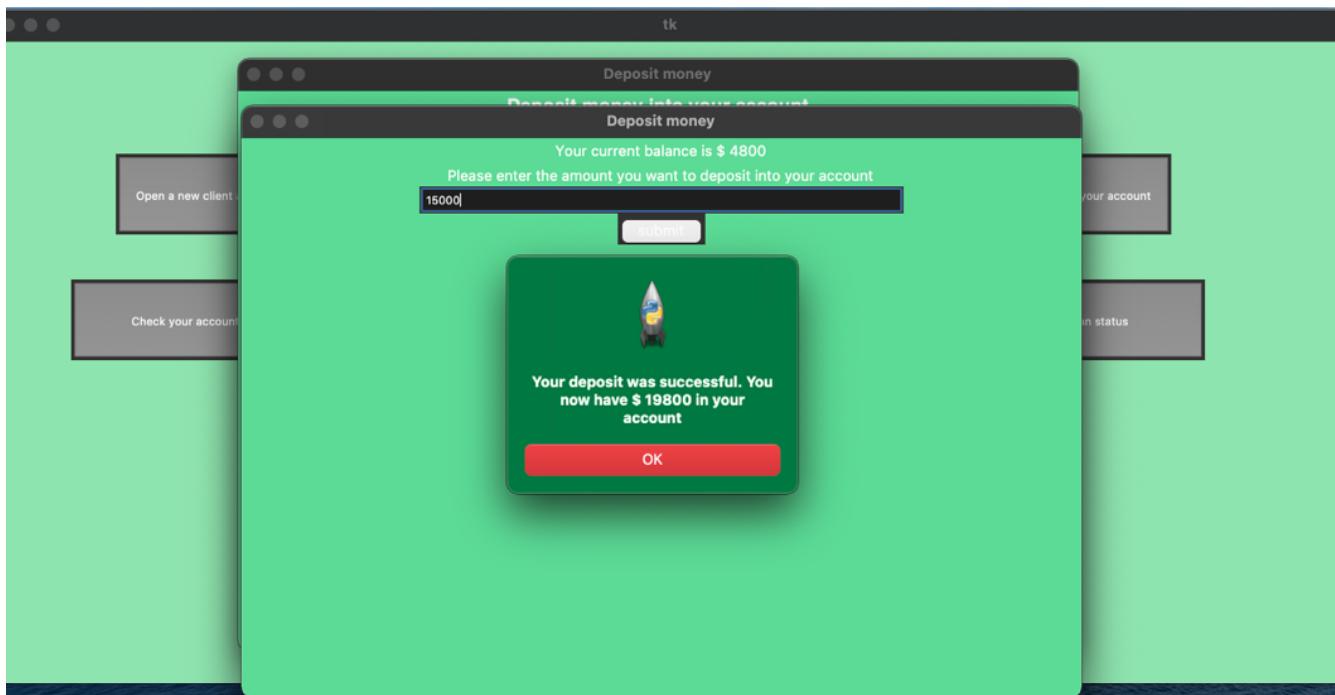
Money Withdrawal Window



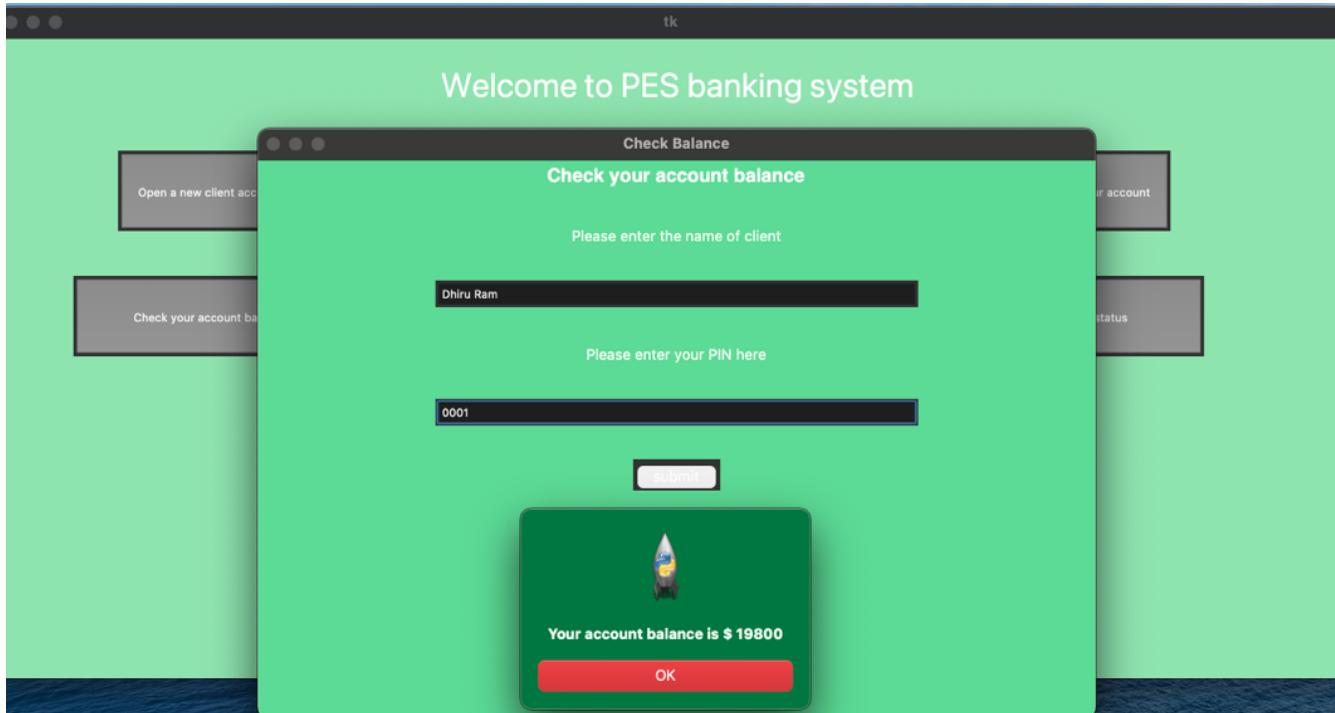
Successful Money Withdrawal with immediate client balance updation



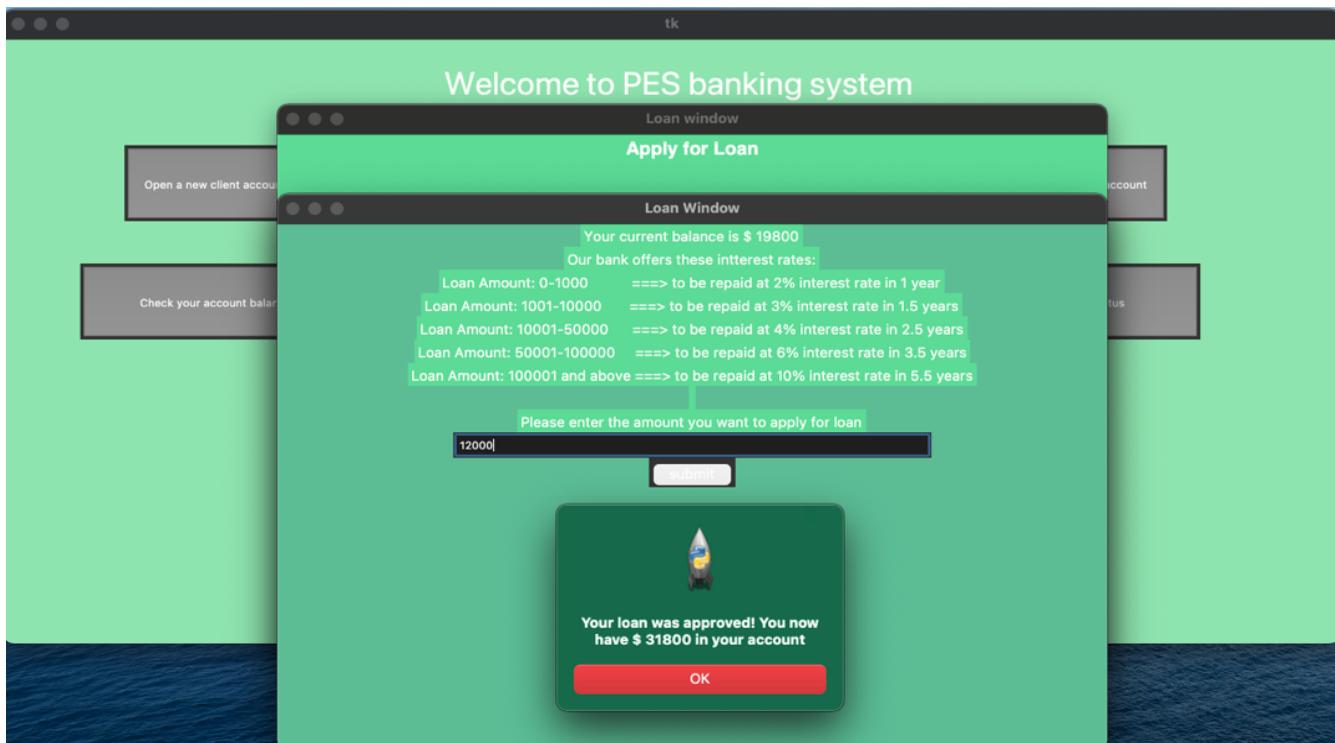
Deposit Money Window



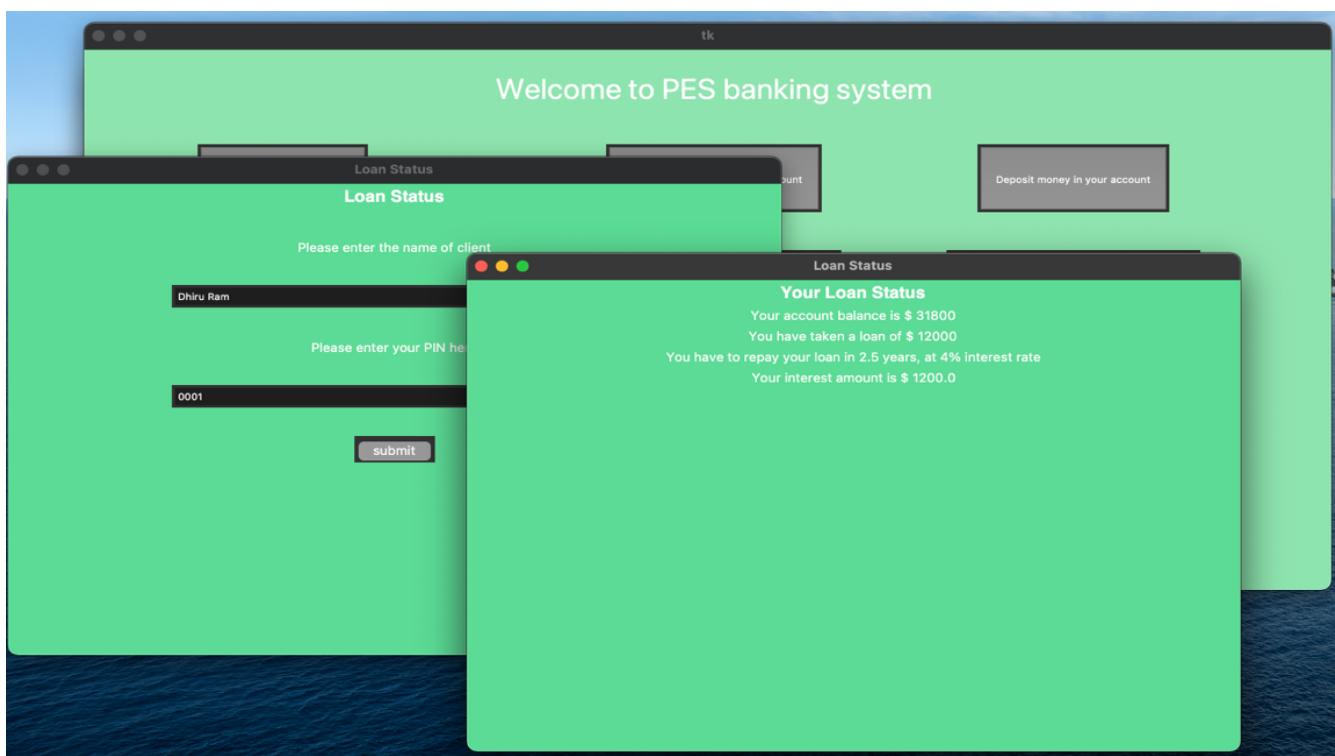
Successful Money Deposition with immediate client balance updation



Check Client Balance Window

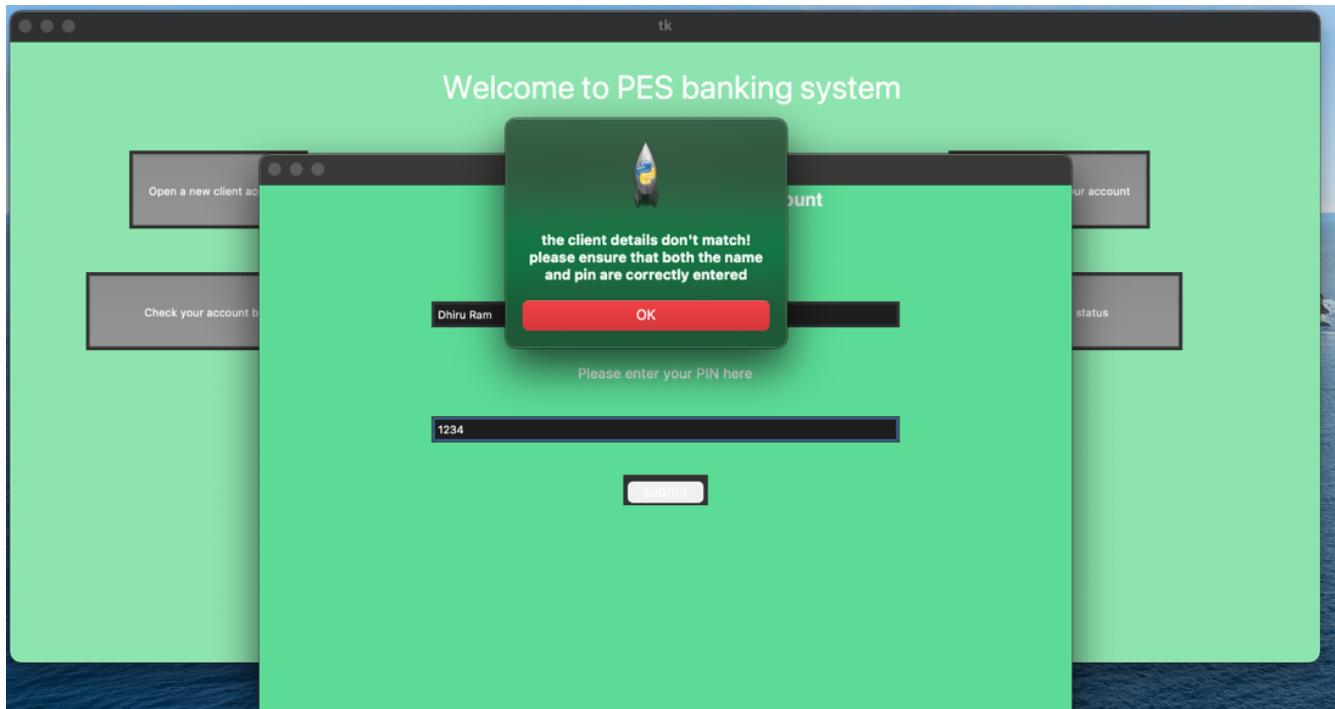


### Applying for Loan Window

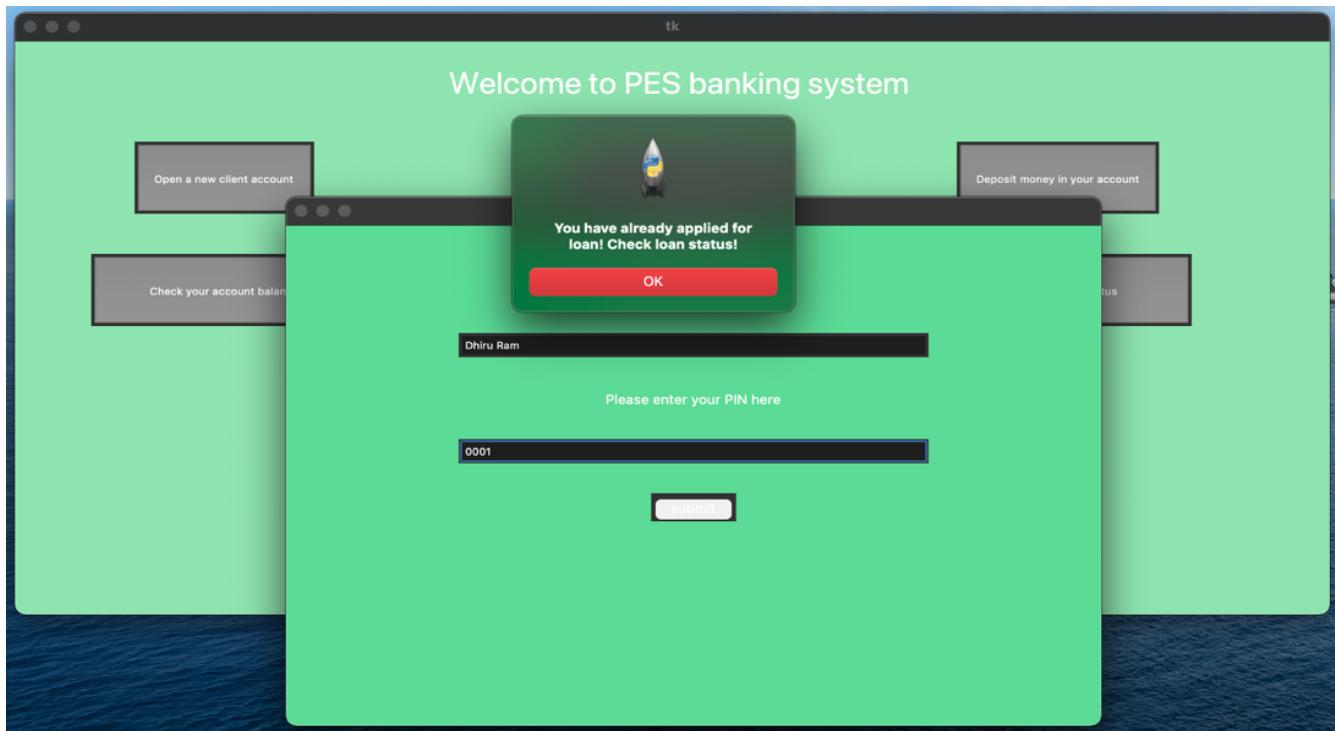


Loan Status Window

TESTING:



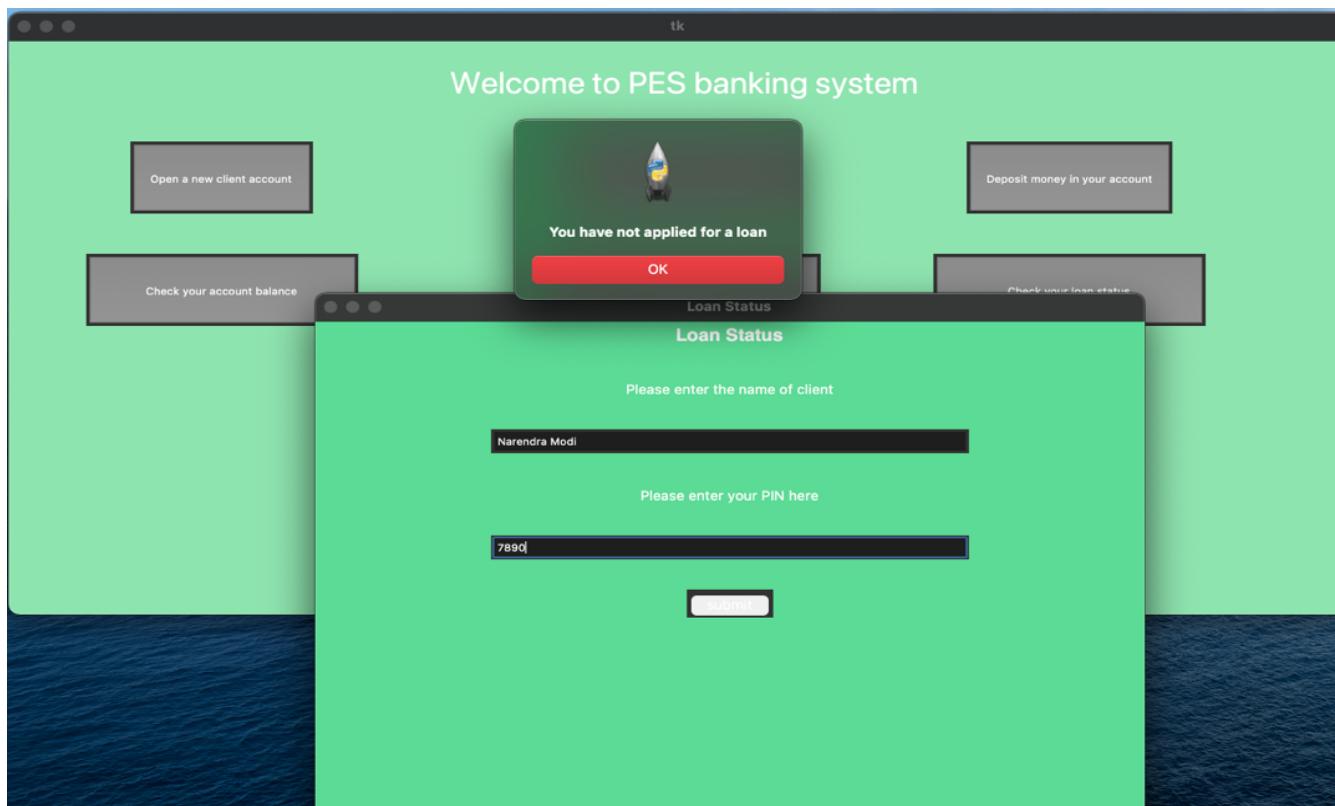
Error Message 1 – Incorrect Name and/or Pin



Error Message 2 – Loan account already exists (Prevents client from taking multiple loans)



Error Message 3 – Loan amount exceeds limit



Error Message 4 – Loan Account required for viewing Loan Status

## **CONCLUSIONS:**

The Bank Management Project not only provided us an insight on the inner workings of a bank, but also given us an indulging hands-on experience with the Python language. This project has helped us polish our fundamentals and basics of Python, and at the same time, has inspired us to get better and more creative at coding and problem solving.

## **FUTURE ENHANCEMENTS:**

Due to lack of extensive knowledge, we were unable to implement several ideas that we would love to add in the future, once we are well equipped with Python. Some of such ideas include:

- A Bank Admin account
- An Activity Log file
- Client Files System
- An OTP generator for extended security

REFERENCES: