

Project

- Desktop Project
- Banking Automation
- Front End
- tkinter
- Bank end
- SQLite (Portable)
- Logic (Python Concepts)

Front End Design(Presentation part)

```
In [5]: from tkinter import *
import time # i use time module because we can format date and time easily
from tkinter.ttk import Combobox, Treeview, Style, Scrollbar # for list , table, style
from tkinter import messagebox # for all dialog box
import sqlite3 # database
from PIL import Image, ImageTk

try:
    conobj=sqlite3.connect(database='banking.sqlite3')
    curobj=conobj.cursor()
    curobj.execute("create table accounts(acn_no integer primary key autoincrement,
    curobj.execute("create table txns(acn_no int,txn_amt float,txn_type text,txn_up
    conobj.commit()
    print("table created")
except:
    print("something went wrong.might be table exists")
conobj.close()

win=Tk()
win.title("Banking Automation") # window title
win.state('zoomed') # window sized
win.configure(bg='lightblue') # window color
win.resizable(width=False,height=False)
# bank Logo image
img=Image.open("bank.ico")
photo=ImageTk.PhotoImage(img)
img.resize((50,50),Image.Resampling.NEAREST)

# window icon image left top one
win.iconbitmap("bank.ico") # window Icon

title_lbl=Label(win,text="Banking Automation",font=('arial',55,'bold','underline'),
title_lbl.pack()

logo_lbl=Label(image=photo)
logo_lbl.place(relx=.77,rely=0)
```

```

def update_clock():
    digi_clock=Label(win,text=time.strftime('%H:%M:%S %p'),font=('arial',18,'bold'))
    digi_clock.place(relx=0.05,rely=.1)
    digi_clock.after(1000,update_clock)
update_clock()

date_lb1=Label(win,text=time.strftime('%d %B-%y'),font=('arial',18,'bold'),bg='lightpink')
date_lb1.place(relx=.88,rely=.1)

def main_screen():
    frm=Frame(win)
    frm.configure(bg='pink')
    frm.place(relx=0,rely=.15,relwidth=1,relheight=.78)

    def newclick():
        frm.destroy()
        openaccount_screen()

    def forgotclick():
        frm.destroy()
        forgot_pass_screen()

    def login_click():
        acn=acn_entry.get()
        ps=pass_entry.get()
        if len(acn)==0 or len(ps)==0:
            messagebox.showerror("Login","Empty fields are not allowed")
            return
        elif not acn.isdigit():
            messagebox.showerror("Login","Incorrect ACN")
            return
        else:
            conobj=sqlite3.connect(database="banking.sqlite3")
            curobj=conobj.cursor()
            curobj.execute("select * from accounts where acn_no=? and acn_pass=?", (acn,ps))
            tup=curobj.fetchone()
            conobj.close()
            if tup==None:
                messagebox.showerror("login","Invalid ACN/Pass !!")
            else:
                global uname,uacn
                uacn=tup[0]
                uname=tup[1]
                frm.destroy()
                welcome_screen()

    def clear():
        acn_entry.delete(0,"end")
        pass_entry.delete(0,"end")
        acn_entry.focus()

    acn_lb1=Label(frm,text="ACN",font=('arial',30,'bold'),fg='blue',bg='pink')
    acn_lb1.place(relx=.35,rely=.1)

```

```

acn_entry=Entry(frm,font=('arial',20,'bold'),bd=5)
acn_entry.place(relx=.45,relx=.1)
acn_entry.focus()

pass_lbl=Label(frm,text="PASS",font=('arial',28,'bold'),fg='blue',bg='pink')
pass_lbl.place(relx=.35,relx=.2)

pass_entry=Entry(frm,font=('arial',20,'bold'),bd=5,show='*')
pass_entry.place(relx=.45,relx=.2)

btn_login=Button(frm,command=login_click,text='Login',font=('arial',20,'bold'),
btn_login.place(relx=.45,relx=.3)

btn_clear=Button(frm,command=clear,text='clear',font=('arial',20,'bold'),width=
btn_clear.place(relx=.54,relx=.3)

btn_forgotpass=Button(frm,command=forgotclick,text='Recover Password',font=('ar
btn_forgotpass.place(relx=.45,relx=.45)

btn_openaccount=Button(frm,command=newclick,text='Open account',font=('arial',2
btn_openaccount.place(relx=.45,relx=.58)

def footer():
    footer_lb=Label(win,text="Created By VARUN SABHARWAL",font=('arial',18,'bold'),
    footer_lb.place(relx=.38,relx=.94)

def openaccount_screen():
    frm=Frame(win)
    frm.configure(bg='pink')
    frm.place(relx=0,relx=.15,relwidth=1,relheight=.78)

def back():
    frm.destroy()
    main_screen()

def newacn_db():
    import re
    name=name_entry.get()
    ps=pass_entry.get()
    email=email_entry.get()
    mob=mob_entry.get()
    acntype=cb_entry.get()
    bal=0
    opendate=str(time.ctime())
    pattern=r"[a-zA-z0-9_\.]+\@[a-zA-Z]{2,5}[.][a-zA-Z]{2,5}"
    #pn=r"[a-zA-Z]+\{0,40}"

    if name.isalpha():
        pass
    else:
        messagebox.showerror("Validtaion","Alphabet Only")
        return
    if len(name)==0 or len(ps)==0 or len(email)==0 or len(mob)==0:
        messagebox.showerror("validation","Empty fileds are not allowed")
        return
    if len(mob)!=10 or mob[0] not in '6789':

```

```

        messagebox.showerror("validation","invalid mobile number")
        return
    if re.match(pattern,email):

        conobj=sqlite3.connect(database='banking.sqlite3')
        curobj=conobj.cursor()
        curobj.execute("insert into accounts(acn_name,acn_pass,acn_email,acn_mo
        conobj.commit()
        curobj.close()
        curobj=conobj.cursor()
        curobj.execute("select max(acn_no) from accounts")
        tup=curobj.fetchone()
        conobj.close()

        messagebox.showinfo("New Account",f"Your Account is opened with ACN={t
    else:
        messagebox.showerror("Validation","Email id is not correct")
        return
btn_back=Button(frm,command=back,text='Back',font=('arial',20,'bold'),bg='light
btn_back.place(relx=0,relx=0)

name_lb=Label(frm,text="Name",font=('arial',28,'bold'),fg='blue',bg='pink')
name_lb.place(relx=.35,relx=.15)

name_entry=Entry(frm,font=('arial',20,'bold'),bd=5)
name_entry.place(relx=.45,relx=.15)

pass_lb=Label(frm,text="Pass",font=('arial',28,'bold'),fg='blue',bg='pink')
pass_lb.place(relx=.36,relx=.25)

pass_entry=Entry(frm,font=('arial',20,'bold'),bd=5,show="*")
pass_entry.place(relx=.45,relx=.25)

email_lb=Label(frm,text="Email",font=('arial',28,'bold'),fg='blue',bg='pink')
email_lb.place(relx=.35,relx=.35)

email_entry=Entry(frm,font=('arial',20,'bold'),bd=5)
email_entry.place(relx=.45,relx=.35)

mob_lb=Label(frm,text="Mob",font=('arial',28,'bold'),fg='blue',bg='pink')
mob_lb.place(relx=.37,relx=.45)

mob_entry=Entry(frm,font=('arial',20,'bold'),bd=5)
mob_entry.place(relx=.45,relx=.45)

cb_lb=Label(frm,text="Type",font=('arial',28,'bold'),fg='blue',bg='pink')
cb_lb.place(relx=.36,relx=.55)

cb_entry=Combobox(frm,font=('arial',28,'bold'),values=['Saving','Current','FD'])
cb_entry.current(0)
cb_entry.place(relx=.45,relx=.55)

btn_submit=Button(frm,command=newacn_db,text='Submit',font=('arial',20,'bold'),
btn_submit.place(relx=.45,relx=.68)

btn_clear=Button(frm,text='Clear',font=('arial',20,'bold'),bg='lightblue')

```

```

btn_clear.place(relx=.61,relly=.68)

def forgot_pass_screen():
    frm=Frame(win)
    frm.configure(bg='pink')
    frm.place(relx=0,relly=.15,relwidth=1,relheight=.78)

def back():
    frm.destroy()
    main_screen()

def recoverpass_db():
    acn=acn_entry.get()
    email=email_entry.get()
    mob=mob_entry.get()

    conobj=sqlite3.connect(database="banking.sqlite3")
    curobj=conobj.cursor()
    curobj.execute("select acn_pass from accounts where acn_no=? and acn_email=
tup=curobj.fetchone()
    conobj.close()
    if tup==None:
        messagebox.showerror("Recover Pass","Acount does not exist !!")
    else:
        messagebox.showinfo("Recover Pass",f"Your Password = {tup[0]}")

def clear():
    acn_entry.delete(0,"end")
    email_entry.delete(0,"end")
    mob_entry.delete(0,'end')
    acn_entry.focus()

btn_back=Button(frm,command=back,text='Back',font=('arial',20,'bold'),bg='light
btn_back.place(relx=0,relly=0)

acn_lb=Label(frm,text="ACN. No",font=('arial',28,'bold'),fg='blue',bg='pink')
acn_lb.place(relx=.33,relly=.15)

acn_entry=Entry(frm,font=('arial',20,'bold'),bd=5)
acn_entry.place(relx=.45,relly=.15)

email_lb=Label(frm,text="Email",font=('arial',28,'bold'),fg='blue',bg='pink')
email_lb.place(relx=.35,relly=.33)

email_entry=Entry(frm,font=('arial',20,'bold'),bd=5)
email_entry.place(relx=.45,relly=.33)

mob_lb=Label(frm,text="Mob",font=('arial',28,'bold'),fg='blue',bg='pink')
mob_lb.place(relx=.37,relly=.5)

mob_entry=Entry(frm,font=('arial',20,'bold'),bd=5)
mob_entry.place(relx=.45,relly=.5)

btn_submit=Button(frm,command=recoverpass_db,text='Recover',font=('arial',20,'b
btn_submit.place(relx=.45,relly=.7)

```

```

btn_clear=Button(frm,command=clear,text='Clear',font=('arial',20,'bold'),bg='lightblue')
btn_clear.place(relx=.61,relly=.7)

def welcome_screen():
    frm=Frame(win)
    frm.configure(bg='pink')
    frm.place(relx=0,relly=.15,relwidth=1,relheight=.78)

def logout():
    res=messagebox.askyesno('Logout',"Do you want to logout?")
    if res==True:
        frm.destroy()
        main_screen()

def update_screen():
    ifrm=Frame(frm,highlightthickness=2,highlightbackground = "black")
    ifrm.configure(bg='White')
    ifrm.place(relx=.2,relly=.14,relwidth=.7,relheight=.64)
    tittle_lb=Label(ifrm,text="This is Update Profile Screen",font=('arial',20,'bold'))
    tittle_lb.pack()

def update_db():
    name=name_entry.get()
    ps=pass_entry.get()
    email=email_entry.get()
    mob=mob_entry.get()

    if name.isalpha():
        pass
    else:
        messagebox.showerror("Validtaion","Alphabet Only")
        return
    if len(name)==0 or len(ps)==0 or len(email)==0 or len(mob)==0:
        messagebox.showerror("validation","Empty fileds are not allowed")
        return
    if len(mob)!=10 or mob[0] not in '6789':
        messagebox.showerror("validation","invalid mobile number")
        return

    conobj=sqlite3.connect(database="banking.sqlite3")
    curobj=conobj.cursor()
    curobj.execute("update accounts set acn_name=?,acn_pass=?,acn_email=?,acn_mob=?")
    conobj.commit()
    conobj.close()
    messagebox.showinfo("Update Profile","Profile Updated")
    frm.destroy()
    global uname
    uname=name
    welcome_screen()

conobj=sqlite3.connect(database="banking.sqlite3")
curobj=curobj.cursor()
curobj.execute("select * from accounts where acn_no=?",(uacn,))
tup=curobj.fetchone()
conobj.close()

```

```

name_lb=Label(ifrm,text="Name",font=('arial',20,'bold'),fg='blue',bg='white')
name_lb.place(relx=.1,rely=.2)

name_entry=Entry(ifrm,font=('arial',20,'bold'),bd=4)
name_entry.place(relx=.2,rely=.2)
name_entry.insert(0,tup[1])

pass_lb=Label(ifrm,text="Pass",font=('arial',20,'bold'),fg='blue',bg='white')
pass_lb.place(relx=.53,rely=.2)

pass_entry=Entry(ifrm,font=('arial',20,'bold'),bd=4)
pass_entry.place(relx=.62,rely=.2)
pass_entry.insert(0,tup[2])

email_lb=Label(ifrm,text="Email",font=('arial',20,'bold'),fg='blue',bg='white')
email_lb.place(relx=.1,rely=.5)

email_entry=Entry(ifrm,font=('arial',20,'bold'),bd=4)
email_entry.place(relx=.2,rely=.5)
email_entry.insert(0,tup[3])

mob_lb=Label(ifrm,text="Mob.",font=('arial',20,'bold'),fg='blue',bg='white')
mob_lb.place(relx=.54,rely=.5)

mob_entry=Entry(ifrm,font=('arial',20,'bold'),bd=4)
mob_entry.place(relx=.62,rely=.5)
mob_entry.insert(0,tup[4])

btn_update=Button(ifrm,command=update_db,text='Update',font=('arial',20,'bold'),fg='blue',bg='white')
btn_update.place(relx=.64,rely=.7)

def checkbal_screen():
    ifrm=Frame(frm,highlightthickness=2,highlightbackground = "black")
    ifrm.configure(bg='White')
    ifrm.place(relx=.2,rely=.14,relwidth=.7,relheight=.64)
    bal_lb=Label(ifrm,text="This is Check Bal Screen",font=('arial',20,'bold'),fg='blue',bg='white')
    bal_lb.pack()

    conobj=sqlite3.connect(database="banking.sqlite3")
    curobj=conobj.cursor()
    curobj.execute("Select acn_bal,acn_opendate from accounts where acn_no=?", (tup[0],))
    tup=curobj.fetchone()
    conobj.close()

    availbal_lb=Label(ifrm,text=f"Available balance \t\t {tup[0]}",font=('arial',20,'bold'),fg='blue',bg='white')
    availbal_lb.place(relx=.2,rely=.2)

    opendate_lb=Label(ifrm,text=f"Acn opendate \t\t {tup[1]}",font=('arial',20,'bold'),fg='blue',bg='white')
    opendate_lb.place(relx=.2,rely=.4)

def despoit_screen():
    ifrm=Frame(frm,highlightthickness=2,highlightbackground = "black")
    ifrm.configure(bg='White')
    ifrm.place(relx=.2,rely=.14,relwidth=.7,relheight=.64)

```

```
tittle_lb=Label(ifrm,text="This is Despoit Screen",font=('arial',20,'bold'))
tittle_lb.pack()
```

```
def deposit_db():
    try:
        amt=float(amt_entry.get())
        if amt>0:
            conobj=sqlite3.connect(database="banking.sqlite3")
            curobj=conobj.cursor()
            curobj.execute("select acn_bal from accounts where acn_no=?", (u
            tup=curobj.fetchone() # fecthing old balance
            bal=tup[0]
            curobj.close()

            curobj=conobj.cursor()
            curobj.execute("update accounts set acn_bal=acn_bal+? where acn
            curobj.execute("insert into txns values(?,?,?,?,"),(uacn,amt,'
            conobj.commit()
            conobj.close()
            messagebox.showinfo("Deposit",f"{amt} deposit")
        else:
            messagebox.showerror("Error", "Number will be Postive Only")
    except:
        messagebox.showerror("Validation", "Number Only")
```

```
amt_lb=Label(ifrm,text="Amount",font=('arial',20,'bold'),fg='blue',bg='whit
amt_lb.place(relx=.3,relx=.3)
```

```
amt_entry=Entry(ifrm,font=('arial',20,'bold'),bd=5)
amt_entry.place(relx=.45,relx=.3)
```

```
btn_submit=Button(ifrm,command=deposit_db,text='Despoit',font=('arial',20,'
btn_submit.place(relx=.50,relx=.6)
```

```
def withdraw_screen():
    ifrm=Frame(frm,highlightthickness=2,highlightbackground = "black")
    ifrm.configure(bg='White')
    ifrm.place(relx=.2,relx=.14,relwidth=.7,relheight=.64)
    tittle_lb=Label(ifrm,text="This is withdraw Screen",font=('arial',20,'bold')
    tittle_lb.pack()
```

```
def withdraw_db():
    try:
        amt=float(amt_entry.get())
        if amt>0:
            conobj=sqlite3.connect(database="banking.sqlite3")
            curobj=conobj.cursor()
            curobj.execute("select acn_bal from accounts where acn_no=?", (u
            tup=curobj.fetchone() # fecthing old balance
            bal=tup[0]
            curobj.close()
            if bal>=amt:
                curobj=conobj.cursor()
                curobj.execute("update accounts set acn_bal=acn_bal-? where
                curobj.execute("insert into txns values(?,?,?,?,"),(uacn,a
```



```

        conobj.commit()
        conobj.close()
        messagebox.showinfo("Withdraw",f"{amt} withdrawn")
    else:
        messagebox.showerror("Withdraw",f"{amt} withdrawn")
    else:
        messagebox.showerror("Error", "Number will be Postive Only")
except:
    messagebox.showerror("Validation", "Number Only")

amt_lb=Label(ifrm,text="Amount",font=('arial',20,'bold'),fg='blue',bg='white')
amt_lb.place(relx=.3,relx=.3)

amt_entry=Entry(ifrm,font=('arial',20,'bold'),bd=5)
amt_entry.place(relx=.45,relx=.3)

btn_submit=Button(ifrm,command=withdraw_db,text='Withdraw',font=('arial',20,'bold'))
btn_submit.place(relx=.50,relx=.6)

def transfer_screen():
    ifrm=Frame(frm,highlightthickness=2,highlightbackground = "black")
    ifrm.configure(bg='white')
    ifrm.place(relx=.2,relx=.14,relwidth=.7,relheight=.64)
    tittle_lb=Label(ifrm,text="This is Transfer Screen",font=('arial',20,'bold'))
    tittle_lb.pack()

def transfer_db():
    try:
        amt=float(amt_entry.get())
        toacn=to_entry.get()

        conobj=sqlite3.connect(database="banking.sqlite3")
        curobj=conobj.cursor()
        curobj.execute("select acn_bal from accounts where acn_no=?", (uacn,))
        tup=curobj.fetchone() # fetching old balance from current account
        bal_frm=tup[0]
        curobj.close()

        conobj=sqlite3.connect(database="banking.sqlite3")
        curobj=conobj.cursor()
        curobj.execute("select acn_bal from accounts where acn_no=?", (toacn,))
        tup=curobj.fetchone() # fetching old balance for to account
        bal_to=tup[0]
        curobj.close()

        curobj=conobj.cursor()
        curobj.execute("select acn_no from accounts where acn_no=?", (toacn,))
        tup=curobj.fetchone() # fetching old balance
        if tup==None:
            messagebox.showerror("Transfer", "To ACN {toacn} does not exist")
        else:
            if bal_frm>=amt:
                curobj=conobj.cursor()
                curobj.execute("update accounts set acn_bal=acn_bal-? where acn_no=?", (amt, toacn))
                curobj.execute("update accounts set acn_bal=acn_bal+? where acn_no=?", (amt, uacn))
            else:
                messagebox.showerror("Transfer", "Amount is greater than current balance")
    except:
        messagebox.showerror("Validation", "Invalid Input")

```

```

        curobj.execute("insert into txns values(?,?,?,?,?)",(uacn,a
        curobj.execute("insert into txns values(?,?,?,?,?)",(toacn,

        conobj.commit()
        conobj.close()
        messagebox.showinfo("Transfer Amt",f"{amt} transfer to ACN
    else:
        messagebox.showerror("Withdraw Amt","Insufficient Bal")
except:
    messagebox.showerror("Validtion","Number Only & To ACN {toacn} does

to_lb=Label(ifrm,text="To",font=('arial',20,'bold'),fg='blue',bg='white')
to_lb.place(relx=.3,relly=.2)

to_entry=Entry(ifrm,font=('arial',20,'bold'),bd=5)
to_entry.place(relx=.4,relly=.2)

amt_lb=Label(ifrm,text="Amt",font=('arial',20,'bold'),fg='blue',bg='white')
amt_lb.place(relx=.3,relly=.4)

amt_entry=Entry(ifrm,font=('arial',20,'bold'),bd=5)
amt_entry.place(relx=.4,relly=.4)

btn_submit=Button(ifrm,command=transfer_db,text='Submit',font=('arial',20,'
btn_submit.place(relx=.5,relly=.6)

def txn_screen():
    ifrm=Frame(frm,highlightthickness=2,highlightbackground = "black")
    ifrm.configure(bg='White')
    ifrm.place(relx=.2,relly=.14,relwidth=.7,relheight=.64)
    #tittle_lb=Label(ifrm,text="This is Transaction Screen",font=('arial',20,'b
    #tittle_lb.pack()

    tv=Treeview(ifrm)
    tv.place(x=0,y=0,relheight=1,relwidth=1)

    style = Style()
    style.configure("Treeview.Heading", font=('Arial',15,'bold'),foreground='bl
    sb=Scrollbar(ifrm,orient='vertical',command=tv.yview)
    sb.place(relx=.98,y=0,relheight=1)

    tv['columns']=('Txn date','Txn amount','Txn type','Updated bal')

    tv.column('Txn date',width=150,anchor='c')
    tv.column('Txn amount',width=100,anchor='c')
    tv.column('Txn type',width=100,anchor='c')
    tv.column('Updated bal',width=100,anchor='c')

    tv.heading('Txn date',text='Txn date')
    tv.heading('Txn amount',text='Txn amount')
    tv.heading('Txn type',text='Txn type')
    tv.heading('Updated bal',text='Updated bal')

    tv['show']='headings'
    con=sqlite3.connect(database="banking.sqlite3")

```

```

cur=con.cursor()
cur.execute("select txn_date,txn_amt,txn_type,txn_updateval from txns where

for row in cur:
    tv.insert("", "end", values=(row[0],row[1],row[2],row[3]),tags='ft')
    tv.tag_configure('ft', font=(' ',12))
con.close()

def csv():
    import csv
    con=sqlite3.connect(database="banking.sqlite3")
    cur=con.cursor()
    cur.execute("Select * from txns where acn_no=?", (uacn,))
    tup=cur.fetchall()
    con.close()

    file=open("Statement.csv", 'w')

    writer = csv.writer(file)
    for t in tup:
        writer.writerow(t)
    file.close()
    messagebox.showinfo("CSV", "File is Downloaded")

btn_logout=Button(frm,command=logout,text='Logout',font=('arial',20,'bold'),bg=
btn_logout.place(relx=.92,relly=0)

welcome_lb=Label(frm,text=f"Welcome {uname}",font=('arial',28,'bold'),fg='blue'
welcome_lb.pack()

acn1_lb=Label(frm,text=f"ACN No. {uacn}",font=('arial',23,'bold'),fg='purple',b
acn1_lb.place(x=0,y=0)

btn_update=Button(frm,command=update_screen,width=12,text='Update Profile',font
btn_update.place(relx=.0,relly=.1)

btn_bal=Button(frm,command=checkbal_screen,width=12,text='check Bal',font=('ari
btn_bal.place(relx=.0,relly=.25)

btn_despoit=Button(frm,command=despoit_screen,width=12,text='Despoit',font=('ar
btn_despoit.place(relx=.0,relly=.40)

btn_withdraw=Button(frm,command=withdraw_screen,width=12,text='Withdraw',font=(
btn_withdraw.place(relx=.0,relly=.55)

btn_transfer=Button(frm,command=transfer_screen,width=12,text='Transfer',font=(
btn_transfer.place(relx=.0,relly=.70)

btn_transfer=Button(frm,command=txn_screen,width=12,text='Txn History',font=('a
btn_transfer.place(relx=.0,relly=.85)

btn_excel=Button(frm,command=csv,text='Stm.Download',font=('arial',20,'bold'),b
btn_excel.place(relx=.74,relly=0)

main_screen()

```

```
footer()  
win.mainloop()
```

something went wrong.might be table exists

Database Design

- SQLite(portable)

Tables

- accounts - acn_no integer primary key autoincrement
- acn_name text
- acn_pass text
- acn_email text
- acn_mob text
- acn_bal text
- acn_type text
- acn_opendate text

- txns
 - acn_no int
 - txn_amt float
 - txn_updatebal float
 - txn_date

In []: