Python Coding

```
import tkinter
from tkinter import *
from tkinter import ttk
from tkinter import ttk
from tkinter import messagebox
import mysql.connector as sql
from SHOW import *
import datetime as dt
import time
from subprocess import call

def SplashScreen():
    splashscreen = Tk()
    splashscreen.overrideredirect(1)
    splashscreen.geometry(
        f"825x500+{(splashscreen.winfo_screenwidth() - 825) //
2}+{(splashscreen.winfo_screenheight() - 500) // 2}")
    splashscreen.configure(bg='green',bd=10,relief=SUNKEN)


    Label(splashscreen, text='KAWASAKI', font='Algerian 35',
fg='black', bg='green',bd=10,relief=RAISED).pack()

    #Add image
    image1 = PhotoImage(file="DEV.png")
    label = Label(splashscreen, image=image1, relief = 'raise', bd =
5).pack()

    Label(splashscreen, text="Version 2.O", font='ALGERIAN 10 ',
bg='green', fg='black',bd=10,relief=RAISED).place(x=695, y=55)
    pgbar = ttk.Progressbar(splashscreen, orient='horizontal',
length=600, mode='indeterminate')
    Label(splashscreen, text="Designed By: Dev Chhatrala ",
font='Algerian 13', bg='green',
fg='black',bd=10,relief=RAISED).place(x=517, y=350)
    Label(splashscreen, text="12th Science-B", font='Algerian 13',
bg='green', fg='black',bd=10,relief=RAISED).place(x=640, y=400)
    pgbar.place(x=70, y=450)
    pgbar['maximum'] = 100


txt=Label(splashscreen,text='0%',relief=GROOVE,bg='black',fg='green')#
, bg='#345', fg='#fff')
    txt.place(x=675, y=450)

    for i in range(101):
        time.sleep(0.01)
        pgbar['value'] = i
        pgbar.update()
        txt['text']=pgbar['value'],'%'

    splashscreen.destroy()

    splashscreen.mainloop()
```

```
mydb=sql.connect(host="localhost",user="root",password="12345")#connec
tion to mysql
mycur=mydb.cursor()
mycur.execute("create database if not exists car")
mycur.execute("use car")
mycur.execute('Create table if not exists bike1(cid varchar(30), fname
varchar(30),lname varchar(30), g varchar(30), \
mno varchar(30), id varchar(30), idno varchar(50), \
cname varchar(30), model varchar(30), ftype varchar(30), \
ctype varchar(30), colour varchar(30), cprice varchar(30))')
'''
mycur.execute("create table if not exists appointment"
            "("
            "idno varchar(12) primary key,"
            "name char(50),"
            "age char(3),"
            "gender char(1),"
            "phone varchar(10),"
            "bg varchar(3))")

'''



class KAWASAKI:
    def_init_(self,root):
        self.root=root
        self.root.iconbitmap("L.png")
        self.root.title("BIKE WORLD")
        self.root.geometry("1350x690+0+0")

#==================== Variable bike1============================#
        self.model=StringVar()
        self.cname=StringVar()
        self.colour=StringVar()
        self.ctype=StringVar()
        self.ftype=StringVar()
        self.mno=StringVar()
        self.fname=StringVar()
        self.lname=StringVar()
        self.g=StringVar()
        self.cid=StringVar()
        self.cprice=StringVar()
        self.id=StringVar()
        self.idno=StringVar()


lblTitle=Label(self.root,text="KAWASAKI",bg="green",fg="black",bd=10,r
elief=RIDGE,\
            font=("Bernard MT Condensed",50,"bold"),padx=10,pady=10)
        lblTitle.pack(side=TOP,fill=X)

        self.lbb=Label(self.root,bg='white')
        self.lbb.place(x=25,y=15, width=75, height=88)
        self.ig0=PhotoImage(file='wq.png')
        self.lbb.config(image=self.ig0)
```

```
        self.lbb=Label(self.root,bg='white')
        self.lbb.place(x=1275,y=15, width=75, height=88)
        self.ig8=PhotoImage(file='wq.png')
        self.lbb.config(image=self.ig8)


        frame=Frame(self.root,bd=12,relief=RIDGE,padx=20,bg="black")
        frame.place(x=0,y=120,width=1365,height=275)

        DataFrameLeft=LabelFrame(frame,text="Customer And Bike
Details",bg="green",fg="black",bd=12,relief=RIDGE,font=("times new
roman",12,"bold"),padx=2,pady=3)
        DataFrameLeft.place(x=-13,y=0,width=1327,height=250)

        lbl2=Label(DataFrameLeft,bg="green",fg="black",text="Bike
Name",font=("times new roman",15,"bold"),padx=20,pady=3)
        lbl2.grid(row=0,column=3,sticky=W)
        txtPRN_No=Entry(DataFrameLeft,
textvariable=self.cname,font=("times new roman",15),width=18)
        txtPRN_No.grid(row=0,column=4,sticky=W)

        lbl1=Label(DataFrameLeft,bg="green",fg="black",text="Customer
bike1:",font=("times new roman",13,"bold"),padx=2,pady=3)
        lbl1.grid(row=1,column=0,sticky=W)

        lblid=Label(DataFrameLeft,bg="green",fg="black",text="Customer
Id",font=("times new roman",12,"bold"),padx=2,pady=3)
        lblid.grid(row=2,column=0,sticky=W)
        txtid=Entry(DataFrameLeft, textvariable=self.cid,font=("times
new roman",12),width=20)
        txtid.grid(row=2,column=1,sticky=W)

        lblname=Label(DataFrameLeft,bg="green",fg="black",text="Frist
Name",font=("times new roman",12,"bold"),padx=30,pady=3)
        lblname.grid(row=2,column=2,sticky=W)
        txtname=Entry(DataFrameLeft,
textvariable=self.fname,font=("times new roman",12),width=20)
        txtname.grid(row=2,column=3,sticky=W)

        lbllname=Label(DataFrameLeft,bg="green",fg="black",text="Last
Name",font=("times new roman",12,"bold"),padx=55,pady=3)
        lbllname.grid(row=2,column=4,sticky=W)
        txtlname=Entry(DataFrameLeft,
textvariable=self.lname,font=("times new roman",12),width=20)
        txtlname.grid(row=2,column=5,sticky=W)


lblg=Label(DataFrameLeft,bg="green",fg="black",text="Gender",font=("ti
mes new roman",12,"bold"),padx=30,pady=3)
        lblg.grid(row=2,column=6,sticky=W)

cmbg=ttk.Combobox(DataFrameLeft,textvariable=self.g,font=("times new
roman",12,"bold"),width=18,state="readonly")
        cmbg["value"]=(" ","Male","Female","Other")
        cmbg.current(0)
        cmbg.grid(row=2,column=7,sticky=W)
```

```
        lblp=Label(DataFrameLeft,bg="green",fg="black",text="Id Proof
Type",font=("times new roman",12,"bold"),padx=30,pady=3)
        lblp.grid(row=3,column=2,sticky=W)

cmbp=ttk.Combobox(DataFrameLeft,textvariable=self.id,font=("times new
roman",12,"bold"),width=18,state="readonly")
        cmbp["value"]=(" ","Aadhar Card","Voter Id","Other")
        cmbp.current(0)
        cmbp.grid(row=3,column=3,sticky=W)


lblMobilenumber=Label(DataFrameLeft,bg="green",fg="black",text="Mobile
Number",font=("times new roman",12,"bold"),padx=2,pady=3)
        lblMobilenumber.grid(row=3,column=0,sticky=W)
        txtMobilenumber=Entry(DataFrameLeft,
textvariable=self.mno,font=("times new roman",12),width=20)
        txtMobilenumber.grid(row=3,column=1,sticky=W)


lblMobilepnumber=Label(DataFrameLeft,bg="green",fg="black",text="Aadha
r/Voter/other Id No.",font=("times new
roman",12,"bold"),padx=2,pady=3)
        lblMobilepnumber.grid(row=3,column=4,sticky=W)
        txtMobilepnumber=Entry(DataFrameLeft,
textvariable=self.idno,font=("times new roman",12),width=20)
        txtMobilepnumber.grid(row=3,column=5,sticky=W)


lblCarbike1=Label(DataFrameLeft,bg="green",fg="black",text="Bike
bike1:",font=("times new roman",13,"bold"),padx=2,pady=3)
        lblCarbike1.grid(row=4,column=0,sticky=W)

        lblmodel=Label(DataFrameLeft,bg="green",fg="black",text="Bike
Model",font=("times new roman",12,"bold"),padx=2,pady=3)
        lblmodel.grid(row=5,column=0,sticky=W)
        txtmodel=Entry(DataFrameLeft,
textvariable=self.model,font=("times new roman",12),width=20)
        txtmodel.grid(row=5,column=1,sticky=W)

        lbltype=Label(DataFrameLeft,bg="green",fg="black",text="Fuel
Type",font=("times new roman",12,"bold"),padx=30,pady=3)
        lbltype.grid(row=5,column=2,sticky=W)

cmbtype=ttk.Combobox(DataFrameLeft,textvariable=self.ftype,font=("time
s new roman",12,"bold"),width=18,state="readonly")
        cmbtype["value"]=(" ","Petrol","E-Bike")
        cmbtype.current(0)
        cmbtype.grid(row=5,column=3,sticky=W)

        lblctype=Label(DataFrameLeft,bg="green",fg="black",text="Bike
Type",font=("times new roman",12,"bold"),padx=55,pady=3)
        lblctype.grid(row=5,column=4,sticky=W)

cmbctype=ttk.Combobox(DataFrameLeft,textvariable=self.ctype,font=("tim
es new roman",12,"bold"),width=18,state="readonly")
        cmbctype["value"]=(" ","1-seater","2-seater")
        cmbctype.current(0)
        cmbctype.grid(row=5,column=5,sticky=W)
```

```
        lblcolour=Label(DataFrameLeft,bg="green",fg="black",text="Bike
Colour",font=("times new roman",12,"bold"),padx=30,pady=3)
        lblcolour.grid(row=5,column=6,sticky=W)

cmbcolour=ttk.Combobox(DataFrameLeft,textvariable=self.colour,font=("t
imes new roman",12,"bold"),width=18,state="readonly")
        cmbcolour["value"]=(" ","Blue","Green","Dark Blue",'Black')
        cmbcolour.current(0)
        cmbcolour.grid(row=5,column=7,sticky=W)



        lblprice=Label(DataFrameLeft,bg="green",fg="black",text="Bike
Price",font=("times new roman",12,"bold"),padx=2,pady=3)
        lblprice.grid(row=6,column=0,sticky=W)
        txtprice=Entry(DataFrameLeft,
textvariable=self.cprice,font=("times new roman",12),width=20)
        txtprice.grid(row=6,column=1,sticky=W)




#==================== DataFrame Right====================#


FrameButton=Frame(self.root,bd=12,relief=RIDGE,padx=20,bg="green")
        FrameButton.place(x=0,y=395,width=1366,height=140)


btnAddData1=Button(FrameButton,command=self.add_data,text="",font=("ar
ial",40,"bold"),width=6,bg="white",fg="black",relief=RAISED,bd=5)
        btnAddData1.grid(row=0,column=0,padx=0)
        self.lbb=Label(self.root,bg='white')
        self.lbb.place(x=30,y=415, width=210, height=100)
        self.ig=PhotoImage(file='p.png')
        self.lbb.config(image=self.ig)



        self.lbb=Label(self.root,bg='GREY')
        self.lbb.place(x=350,y=415, width=232, height=97)
        self.ig2=PhotoImage(file='00.png')
        self.lbb.config(image=self.ig2)


        self.lbb=Label(self.root,bg='WHITE')
        self.lbb.place(x=720,y=415, width=252, height=100)
        self.ig3=PhotoImage(file='10.png')
        self.lbb.config(image=self.ig3)


        self.lbb=Label(self.root,bg='black')
        self.lbb.place(x=1070,y=415, width=224, height=97)
        self.ig4=PhotoImage(file='90.png')
        self.lbb.config(image=self.ig4)
```

```
 #====================== Information Frames======================#


Framebike1=Frame(self.root,bd=12,relief=SUNKEN,padx=20,bg="green")
        Framebike1.place(x=0,y=590,width=1366,height=120)

        xScroll=ttk.Scrollbar(Framebike1,orient=HORIZONTAL)
        yScroll=ttk.Scrollbar(Framebike1,orient=VERTICAL)


self.KAWASAKI_Table=ttk.Treeview(Framebike1,column=("cid","fname","lna
me","mno","g","id","idno","cname",\
            "model","ftype","ctype","colour","price"),\
                x=xScroll.set,y=yScroll.set)

xScroll.pack(side=BOTTOM,fill=X)
yScroll.pack(side=RIGHT,fill=Y)

        xScroll.config(command=self.KAWASAKI_Table.xview)
        yScroll.config(command=self.KAWASAKI_Table.yview)

        self.KAWASAKI_Table.heading("cid",text="Customer id")
        self.KAWASAKI_Table.heading("fname",text="First name")
        self.KAWASAKI_Table.heading("lname",text="Last name")
        self.KAWASAKI_Table.heading("g",text="Gender")
        self.KAWASAKI_Table.heading("mno",text="Mobile no.")
        self.KAWASAKI_Table.heading("id",text="ID type")
        self.KAWASAKI_Table.heading("idno",text="ID no.")
        self.KAWASAKI_Table.heading("cname",text="Bike name")
        self.KAWASAKI_Table.heading("model",text="Bike model")
        self.KAWASAKI_Table.heading("ftype",text="Fuel type")
        self.KAWASAKI_Table.heading("ctype",text="Bike type")
        self.KAWASAKI_Table.heading("colour",text="Bike colour")
        self.KAWASAKI_Table.heading("price",text="Bike price")

        self.KAWASAKI_Table["show"]="headings"
        self.KAWASAKI_Table.pack(fill=BOTH,expand=1)

        self.KAWASAKI_Table.column("cid",width=100)
        self.KAWASAKI_Table.column("fname",width=100)
        self.KAWASAKI_Table.column("lname",width=100)
        self.KAWASAKI_Table.column("g",width=100)
        self.KAWASAKI_Table.column("mno",width=100)
        self.KAWASAKI_Table.column("id",width=100)
        self.KAWASAKI_Table.column("idno",width=100)
        self.KAWASAKI_Table.column("cname",width=100)
        self.KAWASAKI_Table.column("model",width=100)
        self.KAWASAKI_Table.column("ftype",width=100)
        self.KAWASAKI_Table.column("ctype",width=100)
        self.KAWASAKI_Table.column("colour",width=100)
        self.KAWASAKI_Table.column("price",width=100)
```

```
        self.fetch_data()
        self.KAWASAKI_Table.bind("<ButtonRelease-1>",self.get_cursor)



#====================================================================#

FrameButton=Frame(self.root,bd=12,relief=RIDGE,padx=20,bg="green")
        FrameButton.place(x=0,y=535,width=1365,height=55)


btnAddData=Button(FrameButton,command=self.add_data,text="ADD",font=("
arial",8,"bold"),width=20,bg="white",fg="black",relief=RAISED,bd=5)
        btnAddData.grid(row=0,column=0,padx=22)


btnShowData=Button(FrameButton,command=Display,text="SHOW",font=("aria
l",8,"bold"),width=15,bg="WHITE",fg="black",relief=RAISED,bd=5)
        btnShowData.grid(row=0,column=1,padx=22)


btnUpdateData=Button(FrameButton,command=self.update_data,text="UPDATE
",font=("arial",8,"bold"),width=20,bg="white",fg="black",relief=RAISED
,bd=5)
        btnUpdateData.grid(row=0,column=2,padx=22)


btnDeleteData=Button(FrameButton,command=self.delete_data,text="DELETE
",font=("arial",8,"bold"),width=20,bg="white",fg="black",relief=RAISED
,bd=5)
        btnDeleteData.grid(row=0,column=3,padx=22)


btnResetData=Button(FrameButton,command=self.reset_data,text="RESET",f
ont=("arial",8,"bold"),width=20,bg="white",fg="black",relief=RAISED,bd
=5)
        btnResetData.grid(row=0,column=4,padx=22)


btnExitData=Button(FrameButton,command=self.iExit,text="EXIT",font=("a
rial",8,"bold"),width=20,bg="white",fg="black",relief=RAISED,bd=5)
        btnExitData.grid(row=0,column=5,padx=22)

#====================================================================#
    def add_data(self):

mydb=sql.connect(host="localhost",user="root",passwd="12345",database=
"car")
        mycur=mydb.cursor()

        mycur.execute("insert into bike1
values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)",(
```

```
self.cid.get(),

self.fname.get(),

self.lname.get(),

self.mno.get(),

self.g.get(),

self.id.get(),

self.idno.get(),

self.cname.get(),

self.model.get(),

self.colour.get(),

self.ctype.get(),

self.ftype.get(),

self.cprice.get()


))
        mydb.commit()
        self.fetch_data()
        self.reset_data()
        messagebox.showinfo("Success","Member has been created
successfully.")
        mycur.close()

    def update_data(self):

mydb=sql.connect(host="localhost",user="root",passwd="12345",database=
"car")
        mycur=mydb.cursor()
        mycur.execute("update bike1 set
fname=%s,lname=%s,mno=%s,g=%s,id=%s,idno=%s,cname=%s,\
            model=%s,colour=%s,ctype=%s,ftype=%s,cprice=%s where
cid=%s",(


self.fname.get(),

self.lname.get(),

self.mno.get(),
                                                        self.g.get(),

self.id.get(),

self.idno.get(),
```

```
self.cname.get(),

self.model.get(),

self.colour.get(),

self.ctype.get(),

self.ftype.get(),

self.cprice.get(),

self.cid.get()

                                                                    ))
        mydb.commit()
        self.fetch_data()
        self.reset_data()
        mydb.close()
        messagebox.showinfo("Success","Member has been updated
successfully.")

    def fetch_data(self):

mydb=sql.connect(host="localhost",user="root",passwd="12345",database=
"car")
        mycur=mydb.cursor()
        mycur.execute("select * from bike1")
        rows=mycur.fetchall()

        if len(rows)!=0:

self.KAWASAKI_Table.delete(*self.KAWASAKI_Table.get_children())
            for i in rows:
                self.KAWASAKI_Table.insert("",END,values=i)
            mydb.commit()
        mydb.close()

    def get_cursor(self,event=""):
        cursor_row=self.KAWASAKI_Table.focus()
        content=self.KAWASAKI_Table.item(cursor_row)
        row=content["values"]
        self.cid.set(row[0]),
        self.fname.set(row[1]),
        self.lname.set(row[2]),
        self.mno.set(row[3]),
        self.g.set(row[4]),
        self.id.set(row[5]),
        self.idno.set(row[6]),
        self.cname.set(row[7]),
        self.model.set(row[8]),
        self.colour.set(row[9]),
        self.ctype.set(row[10]),
        self.ftype.set(row[11]),
```

```
        self.cprice.set(row[12]),


    def show_data(self):
        self.txtBox.insert(END,"Customer Id
Type:\t\t"+self.cid.get()+"\n")
        self.txtBox.insert(END,"Frist Name:\t\t"+self.name.get()+"\n")
        self.txtBox.insert(END,"Last Name:\t\t"+self.lname.get()+"\n")
        self.txtBox.insert(END,"Mobile No.:\t\t"+self.mno.get()+"\n")
        self.txtBox.insert(END,"gender:\t\t"+self.g.get()+"\n")
        self.txtBox.insert(END,"Id Type:\t\t"+self.id.get()+"\n")
        self.txtBox.insert(END,"Id No.:\t\t"+self.idno.get()+"\n")
        self.txtBox.insert(END,"C Name:\t\t"+self.cname.get()+"\n")
        self.txtBox.insert(END,"C Model.:\t\t"+self.model.get()+"\n")
        self.txtBox.insert(END,"Fuel Type:\t\t"+self.ftype.get()+"\n")
        self.txtBox.insert(END,"C Type:\t\t"+self.ctype.get()+"\n")
        self.txtBox.insert(END,"C Colour:\t\t"+self.colour.get()+"\n")
        self.txtBox.insert(END,"C Price:\t\t"+self.cprice.get()+"\n")




    def reset_data(self):
        self.cid.set(""),
        self.fname.set(""),
        self.lname.set(""),
        self.mno.set(""),
        self.g.set(""),
        self.id.set(""),
        self.idno.set(""),
        self.cname.set(""),
        self.model.set(""),
        self.colour.set(""),
        self.ctype.set(""),
        self.ftype.set(""),
        self.cprice.set("")

    def iExit(self):
        iExit=tkinter.messagebox.askyesno("Ford","Do you want to
exit?")
        if iExit>0:
            self.root.destroy()
            return

    def delete_data(self):
        if self.cid.get()=="" or self.fname.get()=="":
            messagebox.showerror("Error!!!","First select the
Member.")
        else:

mydb=sql.connect(host="localhost",user="root",passwd="12345",database=
"car")
            mycur=mydb.cursor()
            query="delete from bike1 where cid=%s"
            value=(self.cid.get(),)
```

```
            mycur.execute(query,value)

            mydb.commit()
            self.fetch_data()
            self.reset_data()
            mydb.close()

            messagebox.showinfo("Success","Member has been deleted
successfully.")


  #============================================================#




if __name__=="__main__":
    SplashScreen()
    root=Tk()
    obj=KAWASAKI(root)
    root.mainloop()
```