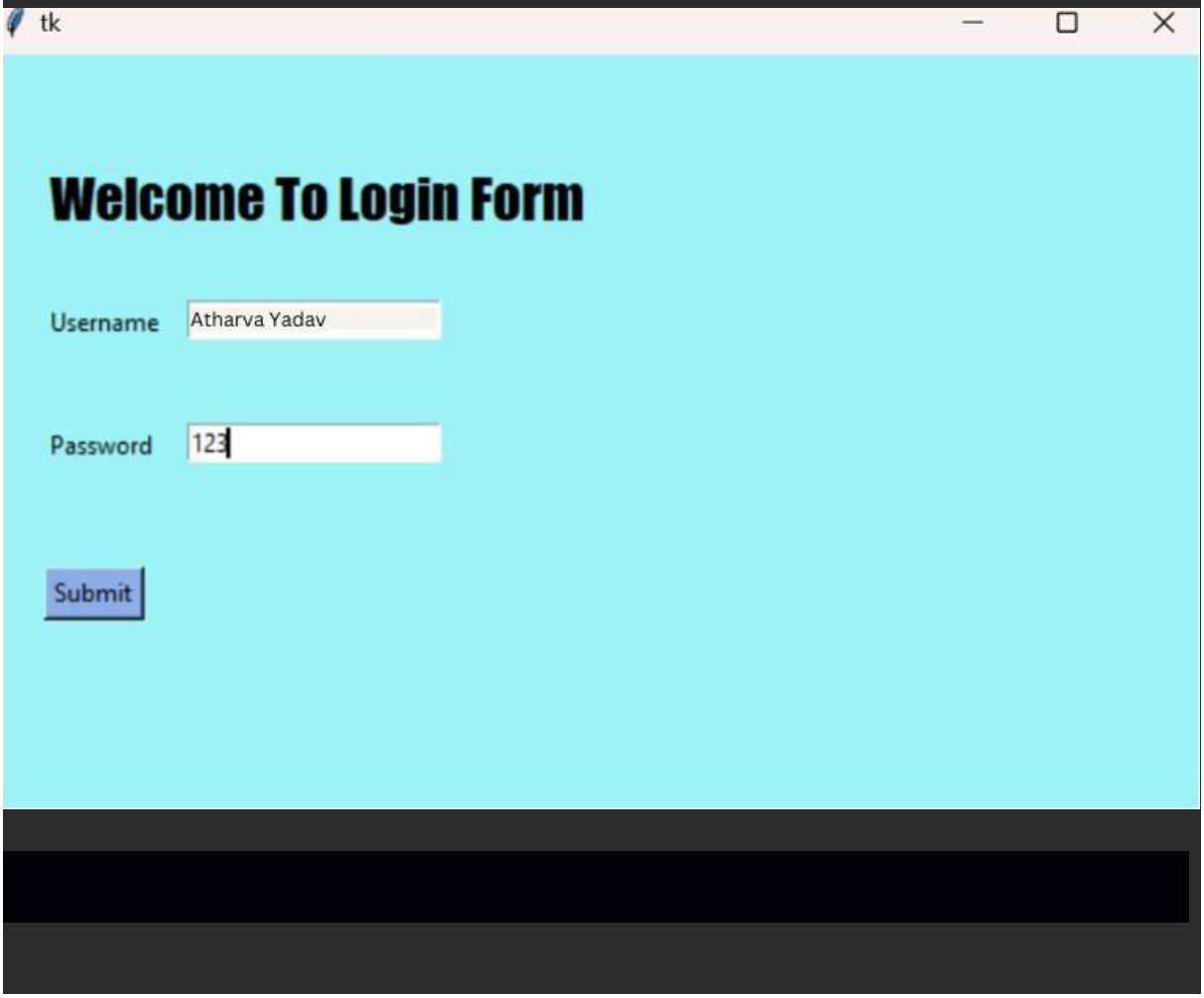
THADOMAL SHAHANI
TSEC
ENGINEERING COLLEGE

ATHARVA YADAV
ROLL NO : 127
BATCH : S23

```python
from tkinter import *

window = Tk()
window.geometry("600x400")
window.configure(background="#9ef3f7")

hello_user = Label(window,text="Welcome To Login Form",
font=("Impact",20),background="#9ef3f7")
hello_user.place(x=30, y=50)

exp_username = "Om Pawaskar"
exp_password = "123"
# creating label
username = Label(window, text="Username",background="#9ef3f7")
username.place(x=30, y=120)

password = Label(window, text="Password",background="#9ef3f7")
password.place(x=30, y=180)
entry_box = Entry(window, width=20)
entry_box.place(x=100, y=120)

entry_box2 = Entry(window, width=20)
entry_box2.place(x=100, y=180)

result = Label(window, text="",background="#9ef3f7")
result.place(x=30, y=300)

def submit():
    if entry_box.get() == exp_username and entry_box2.get() == exp_password:
        result.configure(text="Login Success", font=("Impact", 20))
    else:
        result.configure(text="Login Failed, Invalid Credentials",
        font=("Impact", 20))
```

```
submit_button = Button(window, text="Submit", command=submit,
background="#8caee6")

submit_button.place(x=30, y=250)
window.mainloop()
```




```
from tkinter import *
import math
import tkinter.messagebox

root = Tk()
root.title("Scientific Calculator")
root.configure(background='white')
root.resizable(width=False, height=False)
root.geometry("480x568+455+90")
calc = Frame(root)
calc.grid()
```

```
class
Calc:
  def __init__(self):
      self.total = 0
      self.current = ''
      self.input_value = True
      self.check_sum = False
      self.op = ''
      self.result = False

  def numberEnter(self, num):
      self.result = False
      firstnum = txtDisplay.get()
      secondnum = str(num)
      if self.input_value:
          self.current = secondnum
```

```
          self.input_value = False
      else:
          if secondnum == '.':
              if secondnum in firstnum:
                  return
          self.current = firstnum + secondnum
      self.display(self.current)

  def sum_of_total(self):
      self.result = True
      self.current = float(self.current)
      if self.check_sum:
          self.valid_function()
      else:
          self.total = float(txtDisplay.get())

  def display(self, value):
      txtDisplay.delete(0, END)
      txtDisplay.insert(0, value)

                      def
     valid_function(self):
     if self.op == "add":
         self.total += self.current
     if self.op == "sub":
         self.total -= self.current
     if self.op == "multi":
         self.total *= self.current
     if self.op == "divide":
         self.total /= self.current
```

```
        if self.op == "mod":
            self.total %= self.current
        self.input_value = True
        self.check_sum = False
        self.display(self.total)

    def operation(self, op):
        self.current = float(self.current)
        if self.check_sum:
            self.valid_function()
        elif not self.result:
            self.total = self.current
            self.input_value = True
        self.check_sum = True
        self.op = op
        self.result = False

    def Clear_Entry(self):
        self.result = False
        self.current = "0"
        self.display(0)
        self.input_value = True

    def All_Clear_Entry(self):
```

```
        self.Clear_Entry()
        self.total = 0

    def pi(self):
        self.result = False
        self.current = math.pi
        self.display(self.current)

    def tau(self):
        self.result = False
        self.current = math.tau
        self.display(self.current)

    def e(self):
        self.result = False
        self.current = math.e
        self.display(self.current)

    def mathPM(self):
        self.result = False
       self.current = -(float(txtDisplay.get()))
        self.display(self.current)
```

```
            def
    squared(self):
    self.result = False
   self.current = math.sqrt(float(txtDisplay.get()))
    self.display(self.current)

           def
    cos(self):
    self.result = False
   self.current = math.cos(math.radians(float(txtDisplay.get())))
    self.display(self.current)

def cosh(self):
    self.result = False
   self.current = math.cosh(math.radians(float(txtDisplay.get())))
    self.display(self.current)

def tan(self):
    self.result = False
   self.current = math.tan(math.radians(float(txtDisplay.get())))
    self.display(self.current)

def tanh(self):
    self.result = False
   self.current = math.tanh(math.radians(float(txtDisplay.get())))
    self.display(self.current)

def sin(self):
    self.result = False
   self.current = math.sin(math.radians(float(txtDisplay.get())))
    self.display(self.current)
```

```
def sinh(self):
    self.result = False
   self.current = math.sinh(math.radians(float(txtDisplay.get())))
    self.display(self.current)

def log(self):
    self.result = False
    self.current = math.log(float(txtDisplay.get()))
    self.display(self.current)

def exp(self):
    self.result = False
    self.current = math.exp(float(txtDisplay.get()))
```

```
        self.display(self.current)

    def acosh(self):
        self.result = False
        self.current = math.acosh(float(txtDisplay.get()))
        self.display(self.current)

                def
        asinh(self):
        self.result = False
       self.current = math.asinh(float(txtDisplay.get()))
        self.display(self.current)


                def
        expm1(self):
        self.result = False
       self.current = math.expm1(float(txtDisplay.get()))
        self.display(self.current)

                 def
        lgamma(self):
        self.result = False
       self.current = math.lgamma(float(txtDisplay.get()))
        self.display(self.current)

    def degrees(self):
        self.result = False
        self.current = math.degrees(float(txtDisplay.get()))
        self.display(self.current)

    def log2(self):
        self.result = False
        self.current = math.log2(float(txtDisplay.get()))
        self.display(self.current)

    def log10(self):
        self.result = False
        self.current = math.log10(float(txtDisplay.get()))
        self.display(self.current)

    def log1p(self):
        self.result = False
        self.current = math.log1p(float(txtDisplay.get()))
```

```
        self.display(self.current)
```

```
added_value = Calc()

txtDisplay = Entry(calc, font=('Helvetica', 20, 'bold'),
                   bg='black', fg='white',
                   bd=30, width=28, justify=RIGHT)
txtDisplay.grid(row=0, column=0, columnspan=4, pady=1)
txtDisplay.insert(0, "0")

numberpad = "789456123"
i = 0
btn = []
for j in range(2, 5):
   for k in range(3):
       btn.append(Button(calc, width=6, height=2,
                         bg='black', fg='white',
                         font=('Helvetica', 20, 'bold'),
                         bd=4, text=numberpad[i]))
      btn[i].grid(row=j, column=k, pady=1)
     btn[i]["command"] = lambda x=numberpad[i]: added_value.numberEnter(x)
       i += 1


Button(calc,          text=chr(67),
width=6,
     height=2, bg='powder blue',
     font=('Helvetica', 20, 'bold')
     , bd=4, command=added_value.Clear_Entry
     ).grid(row=1, column=0, pady=1)

Button(calc,      text=chr(67)      +
chr(69),
     width=6, height=2,
     bg='powder blue',
      font=('Helvetica', 20, 'bold'),
      bd=4,
      command=added_value.All_Clear_Entry
      ).grid(row=1, column=1, pady=1)

Button(calc, text="\u221A", width=6, height=2,
      bg='powder blue', font=('Helvetica',
                              20, 'bold'),
      bd=4, command=added_value.squared
      ).grid(row=1, column=2, pady=1)

Button(calc, text="+", width=6, height=2,
      bg='powder blue',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=lambda: added_value.operation("add")
```

```
      ).grid(row=1, column=3, pady=1)

Button(calc, text="-", width=6,
      height=2, bg='powder blue',
      font=('Helvetica', 20, 'bold'),
```

```
      bd=4, command=lambda: added_value.operation("sub")
      ).grid(row=2, column=3, pady=1)

Button(calc, text="x", width=6,
      height=2, bg='powder blue',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=lambda: added_value.operation("multi")
      ).grid(row=3, column=3, pady=1)

Button(calc, text="/", width=6,
      height=2, bg='powder blue',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=lambda: added_value.operation("divide")
      ).grid(row=4, column=3, pady=1)

Button(calc, text="0", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=lambda: added_value.numberEnter(0)
      ).grid(row=5, column=0, pady=1)

Button(calc,          text=".",
width=6,
     height=2, bg='powder blue',
     font=('Helvetica', 20, 'bold'),
     bd=4, command=lambda: added_value.numberEnter(".")
     ).grid(row=5, column=1, pady=1)

Button(calc,          text=chr(177),
width=6,
    height=2, bg='powder blue', font=('Helvetica', 20, 'bold'),
     bd=4, command=added_value.mathPM
     ).grid(row=5, column=2, pady=1)

Button(calc,          text="=",
width=6,
      height=2, bg='powder blue',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=added_value.sum_of_total
      ).grid(row=5, column=3, pady=1)
```

```
# ROW 1 :
Button(calc, text="pi", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=added_value.pi
      ).grid(row=1, column=4, pady=1)

Button(calc, text="Cos", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=added_value.cos
      ).grid(row=1, column=5, pady=1)

Button(calc, text="tan", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=added_value.tan
      ).grid(row=1, column=6, pady=1)

Button(calc, text="sin", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=added_value.sin
      ).grid(row=1, column=7, pady=1)

# ROW 2 :
Button(calc, text="2pi", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=added_value.tau
      ).grid(row=2, column=4, pady=1)

Button(calc, text="Cosh", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
     bd=4, command=added_value.cosh
     ).grid(row=2, column=5, pady=1)


Button(calc,          text="tanh",
width=6,
     height=2, bg='black', fg='white',
     font=('Helvetica', 20, 'bold'),
     bd=4, command=added_value.tanh
     ).grid(row=2, column=6, pady=1)
```

```
Button(calc,          text="sinh",
width=6,
     height=2, bg='black', fg='white',
     font=('Helvetica', 20, 'bold'),
     bd=4, command=added_value.sinh
      ).grid(row=2, column=7, pady=1)

# ROW 3 :
Button(calc, text="log", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=added_value.log
      ).grid(row=3, column=4, pady=1)

Button(calc, text="exp", width=6, height=2,
      bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=added_value.exp
      ).grid(row=3, column=5, pady=1)

Button(calc, text="Mod", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=lambda: added_value.operation("mod")
      ).grid(row=3, column=6, pady=1)
```

```
Button(calc, text="e", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=added_value.e
      ).grid(row=3, column=7, pady=1)

# ROW 4 :
Button(calc, text="log10", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=added_value.log10
      ).grid(row=4, column=4, pady=1)

Button(calc, text="log1p", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=added_value.log1p
      ).grid(row=4, column=5, pady=1)

Button(calc,          text="expm1",
width=6,
```

```
     height=2, bg='black', fg='white',
     font=('Helvetica', 20, 'bold'),
     bd=4, command=added_value.expm1
     ).grid(row=4, column=6, pady=1)


Button(calc,          text="gamma",
width=6,
     height=2, bg='black', fg='white',
     font=('Helvetica', 20, 'bold'),
     bd=4, command=added_value.lgamma
     ).grid(row=4, column=7, pady=1)
# ROW 5 :
Button(calc,          text="log2",
width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=added_value.log2
      ).grid(row=5, column=4, pady=1)

Button(calc, text="deg", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=added_value.degrees
      ).grid(row=5, column=5, pady=1)

Button(calc, text="acosh", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
      bd=4, command=added_value.acosh
      ).grid(row=5, column=6, pady=1)

Button(calc, text="asinh", width=6,
      height=2, bg='black', fg='white',
      font=('Helvetica', 20, 'bold'),
```

```
      bd=4, command=added_value.asinh
      ).grid(row=5, column=7, pady=1)

lblDisplay = Label(calc, text="Scientific Calculator",
                   font=('Helvetica', 30, 'bold'),
                   bg='black', fg='white', justify=CENTER)

lblDisplay.grid(row=0, column=4, columnspan=4)


def iExit():
   iExit = tkinter.messagebox.askyesno("Scientific Calculator",
```

```
                                        "Do you want to exit ?")
    if iExit > 0:
        root.destroy()
        return


def Scientific():
    root.resizable(width=False, height=False)
   root.geometry("944x568+0+0")



def
Standard():
   root.resizable(width=False, height=False)
    root.geometry("480x568+0+0")


menubar = Menu(calc)

# ManuBar 1 :
filemenu       =       Menu(menubar,
tearoff=0)
menubar.add_cascade(label='File',
menu=filemenu)
filemenu.add_command(label="Standard", command=Standard)
filemenu.add_command(label="Scientific", command=Scientific)
filemenu.add_separator()
filemenu.add_command(label="Exit", command=iExit)

# ManuBar 2 :
editmenu = Menu(menubar, tearoff=0)
menubar.add_cascade(label='Edit', menu=editmenu)
editmenu.add_command(label="Cut")
editmenu.add_command(label="Copy")
editmenu.add_separator()
editmenu.add_command(label="Paste")

root.config(menu=menubar)

root.mainloop()
```


Scientific Calculator
File
Edit
675.0
C
CE
√
+
7
8
9
-
4
5
6
x
1
2
3
/
0
.
±
=