

Task : 6 Author:- Sagar Bhoi

## Payroll Payslip Genration

In [1]: *# Weekly Salary*

```
import time
import datetime
from tkinter import *
import tkinter.messagebox

root=Tk()
root.title("Welcome to Payroll system")
root.geometry('1350x650+0+0')
root.configure(background="black")

Tops=Frame(root,width=1350,height=50,bd=8,bg="red")
Tops.pack(side=TOP)

f1=Frame(root,width=600,height=600,bd=8,bg="black")
f1.pack(side=LEFT)

f2=Frame(root,width=300,height=700,bd=8,bg="black")
f2.pack(side=RIGHT)

fla=Frame(f1,width=600,height=200,bd=8,bg="black")
fla.pack(side=TOP)

flb=Frame(f1,width=300,height=600,bd=8,bg="black")
flb.pack(side=TOP)

lblinfo=Label(Tops,font=('times new roman',45,'bold'),text="Employee PayRoll Management system ",
              bd=10,fg="black")
lblinfo.grid(row=0,column=0)

def exit():
    exit = tkinter.messagebox.askyesno("Employee system",
                                       "Do you want to Exit the system ?")

    if exit>0:
        root.destroy()
    return
```

```

def reset():
    Name.set("")
    Address.set("")
    HoursWorked.set("")
    wageshour.set("")
    Payable.set("")
    Taxable.set("")
    NetPayable.set("")
    GrossPayable.set("")
    OverTimeBonus.set("")
    Employerid.set("")
    DOB.set("")
    txtpayslip.delete("1.0", END)

def enterinfo():
    txtpayslip.delete("1.0", END)
    txtpayslip.insert(END, "\t\t Pay Slip \n\n")
    txtpayslip.insert(END, "Name : \t\t" + Name.get() + "\n\n")
    txtpayslip.insert(END, "Address : \t\t" + Address.get() + "\n\n")
    txtpayslip.insert(END, "Employerid : \t\t" + Employerid.get() + "\n\n")
    txtpayslip.insert(END, "DOB : \t\t" + DOB.get() + "\n\n")
    txtpayslip.insert(END, "Hours Worked : \t\t" + HoursWorked.get() + "\n\n")
    txtpayslip.insert(END, "Net Payable : \t\t" + NetPayable.get() + "\n\n")
    txtpayslip.insert(END, "Wages Per Hour : \t\t" + wageshour.get() + "\n\n")
    txtpayslip.insert(END, "Tax Paid : \t\t" + Taxable.get() + "\n\n")
    txtpayslip.insert(END, "Payable : \t\t" + Payable.get() + "\n\n")

def weeklywages():
    txtpayslip.delete("1.0", END)
    hoursworkedperweek = float(HoursWorked.get())
    wagesperhours = float(wageshour.get())

    paydue = wagesperhours * hoursworkedperweek
    paymentdue = "INR", str('%.2f'%(paydue))
    Payable.set(paymentdue)

    tax = paydue * 0.2
    taxable = "INR", str('%.2f'%(tax))
    Taxable.set(taxable)

    netpay = paydue - tax
    netpays = "INR", str('%.2f'%(netpay))
    NetPayable.set(netpays)

    if hoursworkedperweek > 40:

```

```

        overtimehours = (hoursworkedperweek - 40) + wagesperhours * 1.5
        overtime = "INR", str('%.2f'%(overtimehours))
        OverTimeBonus.set(overtime)
    elif hoursworkedperweek <= 40:
        overtimepay = (hoursworkedperweek - 40) + wagesperhours * 1.5
        overtimehrs = "INR", str('%.2f'%(overtimepay))
        OverTimeBonus.set(overtimehrs)
    return

```

```

Name=StringVar()
Address=StringVar()
HoursWorked=StringVar()
wageshour=StringVar()
Payable=StringVar()
Taxable=StringVar()
NetPayable=StringVar()
GrossPayable=StringVar()
OverTimeBonus=StringVar()
Employerid=StringVar()
DOB=StringVar()
TimeOfOrder=StringVar()
DateOfOrder=StringVar()

```

```

DateOfOrder.set(time.strftime("%d/%m/%Y"))

```

```

lblName=Label(fla,text="Name",font=('times new roman',16,'bold'),
              bd=20,fg="white",bg="black").grid(row=0,column=0)
lblAddress=Label(fla,text="Address",font=('times new roman',16,'bold'),
                 bd=20,bg="black",fg="white").grid(row=0,column=2)
lblEmployerid=Label(fla,text="Employerid",font=('times new roman',16,'bold'),
                    bd=20,bg="black",fg="white").grid(row=1,column=0)
lblDob=Label(fla,text="DOB",font=('times new roman',16,'bold'),
             bd=20,bg="black",fg="white").grid(row=1,column=2)
lblHoursWorked=Label(fla,text="Hours Worked",font=('times new roman',16,'bold'),
                     bd=20,bg="black",fg="white").grid(row=2,column=0)
lblHourlyRate=Label(fla,text="Hourly Rate",font=('times new roman',16,'bold'),
                    bd=20,bg="black",fg="white").grid(row=2,column=2)
lblTax=Label(fla,text="Tax",font=('times new roman',16,'bold'),
             bd=20,anchor='w',bg="black",fg="white").grid(row=3,column=0)
lblOverTime=Label(fla,text="OverTime",font=('times new roma',16,'bold'),
                  bd=20,bg="black",fg="white").grid(row=3,column=2)
lblGrossPay=Label(fla,text="GrossPay",font=('times new roman',16,'bold'),
                  bd=20,bg="black",fg="white").grid(row=4,column=0)
lblNetPay=Label(fla,text="Net Pay",font=('times new roman',16,'bold'),

```

```

        bd=20,bg="black",fg="white").grid(row=4,column=2)

etxname=Entry(fla,textvariable=Name,font=('times new roman',16,'bold'),
              bd=16,width=22,justify='left')
etxname.grid(row=0,column=1)

etxaddress=Entry(fla,textvariable=Address,font=('times new roman',16,'bold'),
                 bd=16,width=22,justify='left')
etxaddress.grid(row=0,column=3)

etxemployer=Entry(fla,textvariable=Employerid,font=('times new roman',16,'bold'),
                  bd=16,width=22,justify='left')
etxemployer.grid(row=1,column=1)

etxhoursworked=Entry(fla,textvariable=HoursWorked,font=('times new roman',16,'bold'),
                     bd=16,width=22,justify='left')
etxhoursworked.grid(row=2,column=1)

etxwagesperhours=Entry(fla,textvariable=wageshour,font=('times new roman',16,'bold'),
                       bd=16,width=22,justify='left')
etxwagesperhours.grid(row=2,column=3)

etxnin=Entry(fla,textvariable=DOB,font=('times new roman',16,'bold'),
             bd=16,width=22,justify='left')
etxnin.grid(row=1,column=3)

etxgrosspay=Entry(fla,textvariable=Payable,font=('times new roman',16,'bold'),
                  bd=16,width=22,justify='left')
etxgrosspay.grid(row=4,column=1)

etxnetpay=Entry(fla,textvariable=NetPayable,font=('times new roman',16,'bold'),
                bd=16,width=22,justify='left')
etxnetpay.grid(row=4,column=3)

etxtax=Entry(fla,textvariable=Taxable,font=('times new roman',16,'bold'),
             bd=16,width=22,justify='left')
etxtax.grid(row=3,column=1)

etxovertime=Entry(fla,textvariable=OverTimeBonus,font=('times new roman',16,'bold'),
                  bd=16,width=22,justify='left')
etxovertime.grid(row=3,column=3)

#===== Text Widget =====

payslip=Label(f2,textvariable=DateOfOrder,font=('times new roman',21,'bold'),

```

```

        bg="black",fg="white").grid(row=0,column=0)
txtpayslip=Text(f2,height=22,width=34,bd=16,font=('times new roman',13,'bold'),
                bg="white",fg="black")
txtpayslip.grid(row=1,column=0)

#===== buttons =====

btnweeksalary=Button(flb,text='Weekly Salary',padx=16,pady=16,bd=8,font=('times new roman',16,'bold'),
                    width=14,fg="black",bg="black",command=weeklywages).grid(row=0,column=0)

btnreset=Button(flb,text='Reset',padx=16,pady=16,bd=8,font=('times new roman',16,'bold'),
                width=14,command=reset,bg="black",fg="black").grid(row=0,column=1)

btnpayslip=Button(flb,text='View Payslip',padx=16,pady=16,bd=8,font=('times new roman',16,'bold'),
                  width=14,command=enterinfo,bg="black",fg="black").grid(row=0,column=2)

btnexit=Button(flb,text='Exit System',padx=16,pady=16,bd=8,font=('times new roman',16,'bold'),
               width=14,command=exit,bg="black",fg="black").grid(row=0,column=3)

root.mainloop()

```

In [2]: *# For Monthly Salary*

```

import time
import datetime
from tkinter import *
import tkinter.messagebox

root=Tk()
root.title("Welcome to Payroll system")
root.geometry('1350x650+0+0')
root.configure(background="black")

Tops=Frame(root,width=1350,height=50,bd=8,bg="red")
Tops.pack(side=TOP)

f1=Frame(root,width=600,height=600,bd=8,bg="black")
f1.pack(side=LEFT)

f2=Frame(root,width=300,height=700,bd=8,bg="black")
f2.pack(side=RIGHT)

fla=Frame(f1,width=600,height=200,bd=8,bg="black")
fla.pack(side=TOP)

flb=Frame(f1,width=300,height=600,bd=8,bg="black")

```

```

flb.pack(side=TOP)

lblinfo=Label(Tops,font=('times new roman',45,'bold'),text="Employee PayRoll Management system ",
              bd=10,fg="black")
lblinfo.grid(row=0,column=0)

def exit():
    exit = tkinter.messagebox.askyesno("Employee system",
                                       "Do you want to Exit the system ?")

    if exit>0:
        root.destroy()
        return

def reset():
    Name.set("")
    Address.set("")
    HoursWorked.set("")
    wageshour.set("")
    Payable.set("")
    Taxable.set("")
    NetPayable.set("")
    GrossPayable.set("")
    OverTimeBonus.set("")
    Employerid.set("")
    DOB.set("")
    txtpayslip.delete("1.0", END)

def enterinfo():
    txtpayslip.delete("1.0", END)
    txtpayslip.insert(END, "\t\t Pay Slip \n\n")
    txtpayslip.insert(END, "Name : \t\t" + Name.get() + "\n\n")
    txtpayslip.insert(END, "Address : \t\t" + Address.get() + "\n\n")
    txtpayslip.insert(END, "Employerid : \t\t" + Employerid.get() + "\n\n")
    txtpayslip.insert(END, "DOB : \t\t" + DOB.get() + "\n\n")
    txtpayslip.insert(END, "Hours Worked : \t\t" + HoursWorked.get() + "\n\n")
    txtpayslip.insert(END, "Net Payable : \t\t" + NetPayable.get() + "\n\n")
    txtpayslip.insert(END, "Wages Per Hour : \t\t" + wageshour.get() + "\n\n")
    txtpayslip.insert(END, "Tax Paid : \t\t" + Taxable.get() + "\n\n")
    txtpayslip.insert(END, "Payable : \t\t" + Payable.get() + "\n\n")

def monthlywages():
    txtpayslip.delete("1.0", END)
    hoursworkedperweek = float(HoursWorked.get())
    wagesperhours = float(wageshour.get())

    paydue = wagesperhours * hoursworkedperweek

```

```

paymentdue = "INR", str('%.2f'%(paydue))
Payable.set(paymentdue)

tax = paydue * 0.2
taxable = "INR", str('%.2f'%(tax))
Taxable.set(taxable)

netpay = paydue - tax
netpays = "INR", str('%.2f'%(netpay))
NetPayable.set(netpays)

if hoursworkedperweek > 160:
    overtimehours = (hoursworkedperweek - 160) + wagesperhours * 1.5
    overtime = "INR", str('%.2f'%(overtimehours))
    OverTimeBonus.set(overtime)
elif hoursworkedperweek <= 160:
    overtimepay = (hoursworkedperweek - 160) + wagesperhours * 1.5
    overtimehrs = "INR", str('%.2f'%(overtimepay))
    OverTimeBonus.set(overtimehrs)
return

```

```

Name=StringVar()
Address=StringVar()
HoursWorked=StringVar()
wageshour=StringVar()
Payable=StringVar()
Taxable=StringVar()
NetPayable=StringVar()
GrossPayable=StringVar()
OverTimeBonus=StringVar()
Employerid=StringVar()
DOB=StringVar()
TimeOfOrder=StringVar()
DateOfOrder=StringVar()

```

```

DateOfOrder.set(time.strftime("%d/%m/%Y"))

```

```

lblName=Label(fla,text="Name",font=('times new roman',16,'bold'),
              bd=20,fg="white",bg="black").grid(row=0,column=0)
lblAddress=Label(fla,text="Address",font=('times new roman',16,'bold'),
                 bd=20,bg="black",fg="white").grid(row=0,column=2)
lblEmployerid=Label(fla,text="Employerid",font=('times new roman',16,'bold'),
                    bd=20,bg="black",fg="white").grid(row=1,column=0)
lblDob=Label(fla,text="DOB",font=('times new roman',16,'bold'),

```

```

        bd=20,bg="black",fg="white").grid(row=1,column=2)
lblHoursWorked=Label(fla,text="Hours Worked",font=('times new roman',16,'bold'),
        bd=20,bg="black",fg="white").grid(row=2,column=0)
lblHourlyRate=Label(fla,text="Hourly Rate",font=('times new roman',16,'bold'),
        bd=20,bg="black",fg="white").grid(row=2,column=2)
lblTax=Label(fla,text="Tax",font=('times new roman',16,'bold'),
        bd=20,anchor='w',bg="black",fg="white").grid(row=3,column=0)
lblOverTime=Label(fla,text="OverTime",font=('times new roma',16,'bold'),
        bd=20,bg="black",fg="white").grid(row=3,column=2)
lblGrossPay=Label(fla,text="GrossPay",font=('times new roman',16,'bold'),
        bd=20,bg="black",fg="white").grid(row=4,column=0)
lblNetPay=Label(fla,text="Net Pay",font=('times new roman',16,'bold'),
        bd=20,bg="black",fg="white").grid(row=4,column=2)

etxname=Entry(fla,textvariable=Name,font=('times new roman',16,'bold'),
        bd=16,width=22,justify='left')
etxname.grid(row=0,column=1)

etxaddress=Entry(fla,textvariable=Address,font=('times new roman',16,'bold'),
        bd=16,width=22,justify='left')
etxaddress.grid(row=0,column=3)

etxemployer=Entry(fla,textvariable=Employerid,font=('times new roman',16,'bold'),
        bd=16,width=22,justify='left')
etxemployer.grid(row=1,column=1)

etxhoursworked=Entry(fla,textvariable=HoursWorked,font=('times new roman',16,'bold'),
        bd=16,width=22,justify='left')
etxhoursworked.grid(row=2,column=1)

etxwagesperhours=Entry(fla,textvariable=wageshour,font=('times new roman',16,'bold'),
        bd=16,width=22,justify='left')
etxwagesperhours.grid(row=2,column=3)

etxnin=Entry(fla,textvariable=DOB,font=('times new roman',16,'bold'),
        bd=16,width=22,justify='left')
etxnin.grid(row=1,column=3)

etxgrosspay=Entry(fla,textvariable=Payable,font=('times new roman',16,'bold'),
        bd=16,width=22,justify='left')
etxgrosspay.grid(row=4,column=1)

etxnetpay=Entry(fla,textvariable=NetPayable,font=('times new roman',16,'bold'),
        bd=16,width=22,justify='left')
etxnetpay.grid(row=4,column=3)

```



```

etxtax=Entry(fla,textvariable=Taxable,font=('times new roman',16,'bold'),
            bd=16,width=22,justify='left')
etxtax.grid(row=3,column=1)

etxovertime=Entry(fla,textvariable=OverTimeBonus,font=('times new roman',16,'bold'),
                bd=16,width=22,justify='left')
etxovertime.grid(row=3,column=3)

#===== Text Widget =====

payslip=Label(f2,textvariable=DateOfOrder,font=('times new roman',21,'bold'),
             bg="black",fg="white").grid(row=0,column=0)
txtpayslip=Text(f2,height=22,width=34,bd=16,font=('times new roman',13,'bold'),
              bg="white",fg="black")
txtpayslip.grid(row=1,column=0)

#===== buttons =====

btnsalary=Button(flb,text='Monthly Salary',padx=16,pady=16,bd=8,font=('times new roman',16,'bold'),
                width=14,fg="black",bg="black",command=monthlywages).grid(row=0,column=0)

btnreset=Button(flb,text='Reset',padx=16,pady=16,bd=8,font=('times new roman',16,'bold'),
               width=14,command=reset,bg="black",fg="black").grid(row=0,column=1)

btnpayslip=Button(flb,text='View Payslip',padx=16,pady=16,bd=8,font=('times new roman',16,'bold'),
                 width=14,command=enterinfo,bg="black",fg="black").grid(row=0,column=2)

btnexit=Button(flb,text='Exit System',padx=16,pady=16,bd=8,font=('times new roman',16,'bold'),
               width=14,command=exit,bg="black",fg="black").grid(row=0,column=3)

root.mainloop()

```

## Cash Flow Report

```
In [3]: import pandas as pd
```

```
In [4]: cf_nov = pd.read_excel("Kalyani_Balance_Sheet_November_2022.xlsx")
cf_nov
```

Out[4]:

	Name	Branch Team member	Department	Fees	Chennai	Aurangabad	Unnamed: 6	Student Count	Chennai.1	Aurangabad.1	Total
0	Thirumurugan	Kalyani	Data Science	499	100	399	NaN	103	15092	36305.0	51397.0
1	Harsh Dodiya	Kalyani	Data Science	499	100	399	NaN	Month	2022-11-01 00:00:00	NaN	NaN
2	Kunal pahuja	Kalyani	Data Science	499	200	299	NaN	NaN	NaN	NaN	NaN
3	Arya Singh	Kalyani	Data Science	499	100	399	NaN	NaN	NaN	NaN	NaN
4	Prashansa Shree	Kalyani	Data Science	499	100	399	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...
98	Komal Sharma-K	Kalyani	Data Science	499	100	399	NaN	NaN	NaN	NaN	NaN
99	Roopal Vashishth-K	Kalyani	Data Science	499	100	399	NaN	NaN	NaN	NaN	NaN
100	Ashish Sahu-k	Kalyani	Data Science	499	200	299	NaN	NaN	NaN	NaN	NaN
101	Divya Bhendawadekar-k	Kalyani	Data Science	499	200	299	NaN	NaN	NaN	NaN	NaN
102	Ben Jose Joseph-k	Kalyani	Data Science	499	100	399	NaN	NaN	NaN	NaN	NaN

103 rows × 11 columns

```
In [5]: cf_nov = cf_nov.iloc[:,0:6]
```

```
In [6]: cf_nov
```

Out[6]:

	Name	Branch	Team member	Department	Fees	Chennai	Aurangabad
0	Thirumurugan		Kalyani	Data Science	499	100	399
1	Harsh Dodiya		Kalyani	Data Science	499	100	399
2	Kunal pahuja		Kalyani	Data Science	499	200	299
3	Arya Singh		Kalyani	Data Science	499	100	399
4	Prashansa Shree		Kalyani	Data Science	499	100	399
...	...		...	...	...	...	...
98	Komal Sharma-K		Kalyani	Data Science	499	100	399
99	Roopal Vashishth-K		Kalyani	Data Science	499	100	399
100	Ashish Sahu-k		Kalyani	Data Science	499	200	299
101	Divya Bhendawadekar-k		Kalyani	Data Science	499	200	299
102	Ben Jose Joseph-k		Kalyani	Data Science	499	100	399

103 rows × 6 columns

```
In [7]: Aurangabad = cf_nov["Aurangabad"].sum()
Chennai = cf_nov["Chennai"].sum()
print("Total Cash Flow in different Cities:")

a = input("Enter number 1 for Chennai and 2 for Aurangabad")

if a=="1":
    print(f"Total Cash Flow in Chennai in November: {Chennai}")

elif a=="2":
    print(f"Total Cash Flow in Aurangabad in November: {Aurangabad}")
else:
    print("Wrong Input")
```

```
Total Cash Flow in different Cities:
Enter number 1 for Chennai and 2 for Aurangabad2
Total Cash Flow in Aurangabad in November: 36305
```

```
In [8]: cf_dec = pd.read_excel("Kalyani_Balance_Sheet_December_2022.xlsx")
cf_dec
```

Out[8]:

	Name	Branch	Team member	Department	Chennai	Aurangabad	Fees	Month	Student Count	Chennai.1	Aurangabad.1	Total
0	Sana Tashfeen-k		Kalyani	Data Science	100	399	499	2022-12-01	123.0	16295.0	45082.0	61377.0
1	Yogesh Ujjainkar-k		Kalyani	Data Science	100	399	499	NaT	NaN	NaN	NaN	NaN
2	Aishwarya choudhari-k		Kalyani	Data Science	100	399	499	NaT	NaN	NaN	NaN	NaN
3	Ashish Mishra-k		Kalyani	Data Science	100	399	499	NaT	NaN	NaN	NaN	NaN
4	Vedant satav=k		Kalyani	Data Analyst	200	299	499	NaT	NaN	NaN	NaN	NaN
...	...		...	...	...	...	...	...	...	...	...	...
118	Sujeet Singh Rajpoot-k		Kalyani	Data Science	100	399	499	NaT	NaN	NaN	NaN	NaN
119	Kaligatla Sree Samhitha-K		Kalyani	Data Science	100	399	499	NaT	NaN	NaN	NaN	NaN
120	Manuru Sai Suhas-K		Kalyani	Data Science	100	399	499	NaT	NaN	NaN	NaN	NaN
121	Harsh Prasad-K		Kalyani	Data Science	100	399	499	NaT	NaN	NaN	NaN	NaN
122	Princy Gurnani-K		Kalyani	Data Science	100	399	499	NaT	NaN	NaN	NaN	NaN

123 rows × 11 columns

```
In [9]: cf_dec = cf_dec.iloc[:, 0:6]
cf_dec
```

Out[9]:

	Name	Branch	Team member	Department	Chennai	Aurangabad	Fees
0	Sana Tashfeen-k		Kalyani	Data Science	100	399	499
1	Yogesh Ujjainkar-k		Kalyani	Data Science	100	399	499
2	Aishwarya choudhari-k		Kalyani	Data Science	100	399	499
3	Ashish Mishra-k		Kalyani	Data Science	100	399	499
4	Vedant satav=k		Kalyani	Data Analyst	200	299	499
...	...		...	...	...	...	...
118	Sujeet Singh Rajpoot-k		Kalyani	Data Science	100	399	499
119	Kaligatla Sree Samhitha-K		Kalyani	Data Science	100	399	499
120	Manuru Sai Suhas-K		Kalyani	Data Science	100	399	499
121	Harsh Prasad-K		Kalyani	Data Science	100	399	499
122	Princy Gurnani-K		Kalyani	Data Science	100	399	499

123 rows × 6 columns

In [10]:

```
Aurangabad = cf_dec["Aurangabad"].sum()
Chennai = cf_dec["Chennai"].sum()
print("Total Cash Flow in different Cities:")

a = input("Enter number 1 for Chennai and 2 for Aurangabad")

if a=="1":
    print(f"Total Cash Flow in Chennai in November: {Chennai}")

elif a=="2":
    print(f"Total Cash Flow in Aurangabad in November: {Aurangabad}")
else:
    print("Wrong Input")
```

Total Cash Flow in different Cities:  
Enter number 1 for Chennai and 2 for Aurangabad2  
Total Cash Flow in Aurangabad in November: 45082

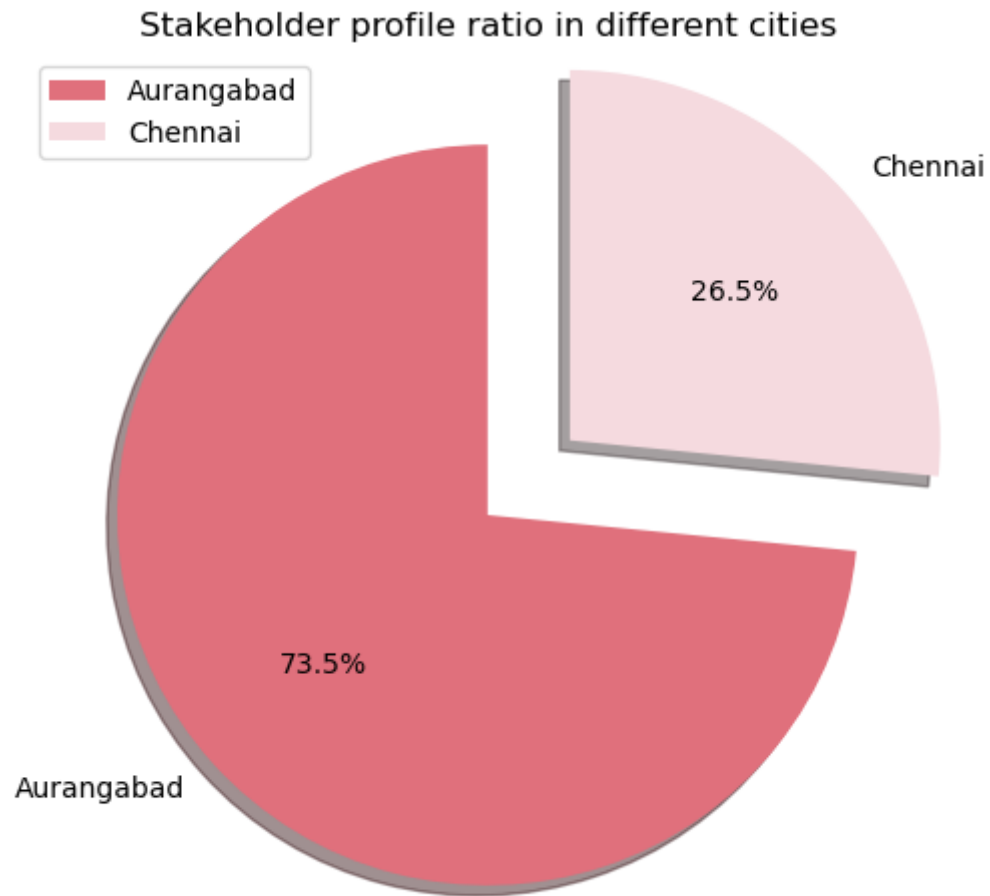
# Stakeholder Distribute Report

In [11]:

```
import matplotlib.pyplot as plt
```

```
In [12]: label="Aurangabad","Chennai"  
         sizes=[Aurangabad,Chennai]  
         colors=["#e0707c", "#F5DADF"]  
         explode=(0,0.3)  
  
plt.pie(sizes,explode=explode,labels=label,autopct="%1.1f%%",shadow=True,startangle=90,colors=colors)  
plt.title("Stakeholder profile ratio in different cities")  
fig = plt.gcf()  
fig.set_size_inches(6, 6)  
plt.legend()
```

Out[12]: <matplotlib.legend.Legend at 0x1a9a455b0a0>



In [ ]: