*UNIVERSITY OF BALTISTAN, SKARDU*

***(DEPARTMENT OF COMPUTER SCIENCE)***

***NAME: SHABBIR HUSSAIN***

***REG NO: S22-BSCS-087***

***SEMESTER: Fall 4th Semester, 2023.***

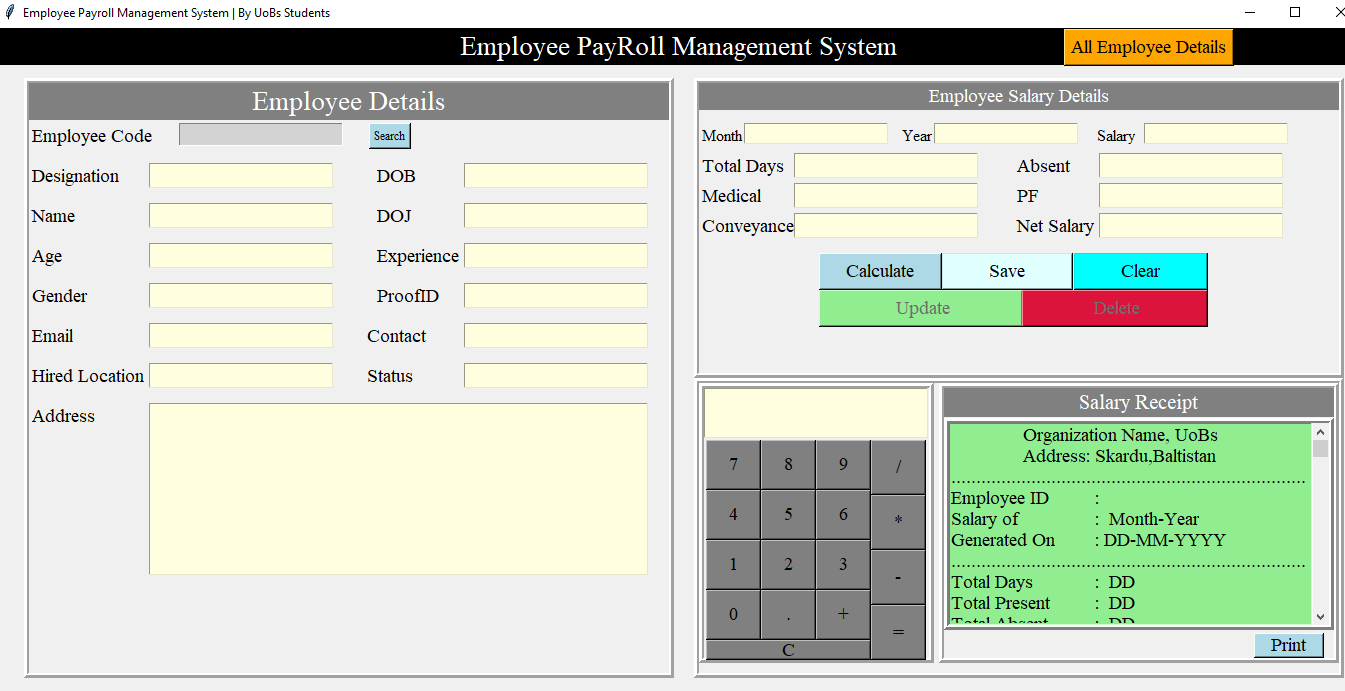
***SUBJECT: Introduction to Database system***

***SUBMITTED TO: Ma’am Noreen***

***SECTION: “B***

**Project: Employee Management System**

Q: Prepare a software to store the required records including pay receipt of the employee, this software should also provide the salary receipt printing option to the user.



**The Code of Project : Employee Management System**

"""Description:

Before Run this code you have to start your xampp Server and create

a database named as 'employeemanangementsystem' then

create a table named as 'emp\_salary'

that contain 24 records var\_emp\_code,  var\_Designation , name , age,

gender , email , hiredlo, txt\_address , dob , doj , experience,

proofid , contact , status , month , year , salary , medical,

pf , conveyance ,  totaldays , absents ,netsalary , receipt

make a folder named as receipt and give the folder path in

file\_=open("F:/Employee Management System/receipt") """

import tkinter

from tkinter import \*

from tkinter import messagebox

from tkinter import ttk

import pymysql

import time

import os

import tempfile

*class* employeesystem:

*def* \_\_init\_\_(*self*,*root*):

*self*.root=*root*

*self*.root.title('Employee Payroll Management System | By SK Company pvt Ltd.')

*self*.root.geometry('1450x700+0+0')

*self*.root.configure(*bg*='green')

        # Set the window icon

*self*.root.iconbitmap(*r*"F:/Employee Manangement System/skjan.jpg")

        title1=Label(*self*.root,*text*='Employee PayRoll Management System',*font*=('times new roman bold italic',20),

*bg*='darkblue',*fg*='white').place(*x*=0,*y*=0,*relwidth*=1)

        btn\_show\_employees=Button(*self*.root,*text*='All Employee Details',*font*=('times new roman bold italic',14),

*fg*='white',*bg*='green',*command*=*self*.employeeframe).place(*x*=1070,*y*=0)

        # Add a copyright label

        copyright\_label = Label(*root*, *text*="© 2023 SK Company Pvt Ltd. All Rights Reserved", *fg*="white",*bg*='blue',*font*=('vardana bold italic',20))

        copyright\_label.pack(*side*="bottom")

        ############### Frame1#########################

        ###########variables###############

*self*.var\_emp\_code=StringVar()

*self*.var\_Designation=StringVar()

*self*.name=StringVar()

*self*.age=StringVar()

*self*.gender=StringVar()

*self*.email=StringVar()

*self*.hiredloc=StringVar()

        #self.address=StringVar()

*self*.dob=StringVar()

*self*.doj=StringVar()

*self*.experience=StringVar()

*self*.proofid=StringVar()

*self*.contact=StringVar()

*self*.status=StringVar()

# ------------------------------------------Frame 1------------------------------------------------------

        frame1=Frame(*self*.root,*bd*=5,*relief*=RIDGE,*bg*='green')

        frame1.place(*x*=30,*y*=50,*width*=650,*height*=600)

        title2=Label(frame1,*text*='Employee Details',*font*=('times new roman bold italic',20),*bg*='darkblue',*fg*='white').place(*x*=0,*y*=0,*relwidth*=1)

        lbl\_empcode=Label(frame1,*text*='Employee Code',*font*=('times new roman bold italic',14),*fg*='black',*bg*='lightgray')

        lbl\_empcode.place(*x*=0,*y*=40)

*self*.entry\_empcode=Entry(frame1,*font*=('times new roman bold italic',13),*textvariable*=*self*.var\_emp\_code,*fg*='black',*bg*='lightgray')

*self*.entry\_empcode.place(*x*=150,*y*=40)

        btn\_search=Button(frame1,*text*='Search',*font*=('times new roman bold italic',10),*command*=*self*.search,*fg*='black',*bg*='lightblue')

        btn\_search.place(*x*=340,*y*=40)

        lbl\_designation=Label(frame1,*text*='Designation',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=0,*y*=80)

        entry\_designation=Entry(frame1,*font*=('times new roman bold italic',14),*textvariable*=*self*.var\_Designation,*fg*='black',*bg*='lightyellow').place(*x*=120,*y*=80)

        lbl\_name=Label(frame1,*text*='Name',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=0,*y*=120)

        entry\_name=Entry(frame1,*font*=('times new roman bold italic',14),*textvariable*=*self*.name,*fg*='black',*bg*='lightyellow').place(*x*=120,*y*=120)

        lbl\_age=Label(frame1,*text*='Age',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=0,*y*=160)

        entry\_age=Entry(frame1,*font*=('times new roman bold italic',14),*textvariable*=*self*.age,*fg*='black',*bg*='lightyellow').place(*x*=120,*y*=160)

        lbl\_gender=Label(frame1,*text*='Gender',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=0,*y*=200)

        entry\_dgender=Entry(frame1,*font*=('times new roman bold italic',14),*textvariable*=*self*.gender,*fg*='black',*bg*='lightyellow').place(*x*=120,*y*=200)

        lbl\_email=Label(frame1,*text*='Email',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=0,*y*=240)

        entry\_email=Entry(frame1,*font*=('times new roman bold italic',14),*textvariable*=*self*.email,*fg*='black',*bg*='lightyellow').place(*x*=120,*y*=240)

        lbl\_hireloc=Label(frame1,*text*='Hired Location',*font*=('times new roman',14),*fg*='black').place(*x*=0,*y*=280)

        entry\_hireloc=Entry(frame1,*font*=('times new roman bold italic',14),*textvariable*=*self*.hiredloc,*fg*='black',*bg*='lightyellow').place(*x*=120,*y*=280)

        lbl\_address=Label(frame1,*text*='Address',*font*=('times new roman',14),*fg*='black').place(*x*=0,*y*=320)

*self*.txt\_address=Text(frame1,*font*=('times new roman bold italic',14),*fg*='black',*bg*='lightyellow',*width*=51,*height*=3)

*self*.txt\_address.place(*x*=120,*y*=320)

        lbl\_DOB=Label(frame1,*text*='DOB',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=345,*y*=80)

        entry\_DOB=Entry(frame1,*font*=('times new roman bold italic',14),*textvariable*=*self*.dob,*fg*='black',*bg*='lightyellow').place(*x*=435,*y*=80)

        lbl\_DOJ=Label(frame1,*text*='DOJ',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=345,*y*=120)

        entry\_DOJ=Entry(frame1,*font*=('times new roman bold italic',14),*textvariable*=*self*.doj,*fg*='black',*bg*='lightyellow').place(*x*=435,*y*=120)

        lbl\_experience=Label(frame1,*text*='Experience',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=345,*y*=160)

        entry\_experience=Entry(frame1,*font*=('times new roman bold italic',14),*textvariable*=*self*.experience,*fg*='black',*bg*='lightyellow').place(*x*=441,*y*=160)

        lbl\_proofid=Label(frame1,*text*='ProofID',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=345,*y*=200)

        entry\_proofid=Entry(frame1,*font*=('times new roman bold italic',14),*textvariable*=*self*.proofid,*fg*='black',*bg*='lightyellow').place(*x*=435,*y*=200)

        lbl\_contact=Label(frame1,*text*='Contact',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=335,*y*=240)

        entry\_contact=Entry(frame1,*font*=('times new roman bold italic',14),*textvariable*=*self*.contact,*fg*='black',*bg*='lightyellow').place(*x*=435,*y*=240)

        lbl\_status=Label(frame1,*text*='Status',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=335,*y*=280)

        entry\_status=Entry(frame1,*font*=('times new roman bold italic',14),*textvariable*=*self*.status,*fg*='black',*bg*='lightyellow').place(*x*=435,*y*=280)

        ############### Frame2#########################

        ##########################variavles##############

*self*.month=StringVar()

*self*.year=StringVar()

*self*.salary=StringVar()

*self*.totaldays=StringVar()

*self*.absents=StringVar()

*self*.medical=StringVar()

*self*.conveyance =StringVar()

*self*.pf=StringVar()

*self*.netsalary=StringVar()

# ---------------------------------------- Frame 2 ----------------------------------------------------

        frame2=Frame(*self*.root,*bd*=5,*relief*=RIDGE,*bg*='green')

        frame2.place(*x*=700,*y*=50,*width*=650,*height*=300)

        title3=Label(frame2,*text*='Employee Salary Details',*font*=('times new roman bold italic',20),*bg*='darkblue',*fg*='white').place(*x*=0,*y*=0,*relwidth*=1)

        lbl\_month=Label(frame2,*text*='Month',*font*=('times new roman bold italic',10),*fg*='black').place(*x*=0,*y*=40)

        entry\_month=Entry(frame2,*font*=('times new roman bold italic',11),*textvariable*=*self*.month,*fg*='black',*bg*='lightyellow').place(*x*=45,*y*=40)

        lbl\_year=Label(frame2,*text*='Year',*font*=('times new roman bold italic',10),*fg*='black').place(*x*=200,*y*=40)

        entry\_year=Entry(frame2,*font*=('times new roman bold italic',11),*fg*='black',*textvariable*=*self*.year,*bg*='lightyellow').place(*x*=235,*y*=40)

        lbl\_salary=Label(frame2,*text*='Salary',*font*=('times new roman bold italic',10),*fg*='black').place(*x*=395,*y*=40)

        entry\_salary=Entry(frame2,*font*=('times new roman bold italic',11),*fg*='black',*textvariable*=*self*.salary,*bg*='lightyellow').place(*x*=445,*y*=40)

        lbl\_totaldays=Label(frame2,*text*='Total Days',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=0,*y*=70)

        entry\_totaldays=Entry(frame2,*font*=('times new roman bold italic',14),*textvariable*=*self*.totaldays,*fg*='black',*bg*='lightyellow').place(*x*=95,*y*=70)

        lbl\_medical=Label(frame2,*text*='Medical',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=0,*y*=100)

        entry\_medical=Entry(frame2,*font*=('times new roman bold italic',14),*textvariable*=*self*.medical,*fg*='black',*bg*='lightyellow').place(*x*=95,*y*=100)

        lbl\_conveyance =Label(frame2,*text*='Conveyance',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=0,*y*=130)

        entry\_conveyance =Entry(frame2,*font*=('times new roman bold italic',14),*textvariable*=*self*.conveyance ,*fg*='black',*bg*='lightyellow').place(*x*=102,*y*=130)

        lbl\_absents=Label(frame2,*text*='Absent',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=315,*y*=70)

        entry\_absent=Entry(frame2,*font*=('times new roman bold italic',14),*textvariable*=*self*.absents,*fg*='black',*bg*='lightyellow').place(*x*=400,*y*=70)

        lbl\_PF=Label(frame2,*text*='PF',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=315,*y*=100)

        entry\_PF=Entry(frame2,*font*=('times new roman bold italic',14),*textvariable*=*self*.pf,*fg*='black',*bg*='lightyellow').place(*x*=400,*y*=100)

        lbl\_netsalary=Label(frame2,*text*='Net Salary',*font*=('times new roman bold italic',14),*fg*='black').place(*x*=315,*y*=130)

        entry\_netsalary=Entry(frame2,*font*=('times new roman bold italic',14),*textvariable*=*self*.netsalary,*fg*='black',*bg*='lightyellow').place(*x*=405,*y*=130)

        btn\_calculate=Button(frame2,*text*='    Calculate    ',*command*=*self*.calculate,*font*=('times new roman',14),

*fg*='black',*bg*='lightblue').place(*x*=120,*y*=170)

*self*.btn\_save=Button(frame2,*text*='        Save        ',*command*=*self*.add,*font*=('times new roman',14),*fg*='black',*bg*='lightcyan')

*self*.btn\_save.place(*x*=243,*y*=170)

        btn\_clear=Button(frame2,*text*='        Clear        ',*command*=*self*.clear,*font*=('times new roman',14),

*fg*='black',*bg*='aqua').place(*x*=375,*y*=170)

*self*.btn\_update=Button(frame2,*text*='              Update             ',*command*=*self*.update,*font*=('times new roman',14),

*fg*='black',*bg*='lightgreen')

*self*.btn\_update.place(*x*=120,*y*=207)

*self*.btn\_delete=Button(frame2,*text*='             Delete            ',*command*=*self*.delete,*font*=('times new roman',14),

*fg*='black',*bg*='crimson')

*self*.btn\_delete.place(*x*=323,*y*=207)

        #self.check\_connection()

# -------------------------------------------- Frame 3 ---------------------------------------------------

        frame3=Frame(*self*.root,*bd*=5,*relief*=RIDGE , *bg*='green')

        frame3.place(*x*=700,*y*=350,*width*=650,*height*=300)

*self*.var\_text=StringVar()

*self*.var\_operator=''

*def* btn\_click(*num*):

*self*.var\_operator=*self*.var\_operator+str(*num*)

*self*.var\_text.set(*self*.var\_operator)

*def* result():

            res=str(eval(*self*.var\_operator))

*self*.var\_text.set(res)

*self*.var\_operator=''

*def* clear\_cal():

*self*.var\_text.set('')

*self*.var\_operator=''

        cal\_frame=Frame(frame3,*bd*=5,*relief*=RIDGE, *bg*='green')

        cal\_frame.place(*x*=0,*y*=0,*width*=235,*height*=280)

        entry\_cal=Entry(cal\_frame,*font*=('times new roman',18,'bold'),

*textvariable*=*self*.var\_text,*justify*=RIGHT,*fg*='black',*bg*='lightyellow')

        entry\_cal.place(*x*=0,*y*=0,*width*=224,*height*=50)

        btn\_n7=Button(cal\_frame,*text*=' 7 ',*command*=*lambda*:btn\_click(7),*font*=('times new roman bold italic',18),

*fg*='white',*bg*='green').place(*x*=2,*y*=52,*width*=55,*height*=50)

        btn\_n8=Button(cal\_frame,*text*=' 8 ',*command*=*lambda*:btn\_click(8),*font*=('times new roman bold italic',18),

*fg*='white',*bg*='green').place(*x*=57,*y*=52,*width*=55,*height*=50)

        btn\_n9=Button(cal\_frame,*text*=' 9 ',*command*=*lambda*:btn\_click(9),*font*=('times new roman bold italic',18),

*fg*='white',*bg*='green').place(*x*=112,*y*=52,*width*=55,*height*=50)

        btn\_ndiv=Button(cal\_frame,*text*=' / ',*command*=*lambda*:btn\_click('/'),*font*=('times new roman bold italic',14),

*fg*='white',*bg*='green').place(*x*=167,*y*=52,*width*=55,*height*=55)

        btn\_n4=Button(cal\_frame,*text*=' 4 ',*command*=*lambda*:btn\_click(4),*font*=('times new roman bold italic',18),

*fg*='white',*bg*='green').place(*x*=2,*y*=102,*width*=55,*height*=50)

        btn\_n5=Button(cal\_frame,*text*=' 5 ',*command*=*lambda*:btn\_click(5),*font*=('times new roman bold italic',18),

*fg*='white',*bg*='green').place(*x*=57,*y*=102,*width*=55,*height*=50)

        btn\_n6=Button(cal\_frame,*text*=' 6 ',*command*=*lambda*:btn\_click(6),*font*=('times new roman bold italic',18),

*fg*='white',*bg*='green').place(*x*=112,*y*=102,*width*=55,*height*=50)

        btn\_nmul=Button(cal\_frame,*text*=' \* ',*command*=*lambda*:btn\_click('\*'),*font*=('times new roman bold italic',18),

*fg*='white',*bg*='green').place(*x*=167,*y*=107,*width*=55,*height*=55)

        btn\_n1=Button(cal\_frame,*text*=' 1 ',*command*=*lambda*:btn\_click(1),*font*=('times new roman',18),

*fg*='white',*bg*='green').place(*x*=2,*y*=152,*width*=55,*height*=50)

        btn\_n2=Button(cal\_frame,*text*=' 2 ',*command*=*lambda*:btn\_click(2),*font*=('times new roman',18),

*fg*='white',*bg*='green').place(*x*=57,*y*=152,*width*=55,*height*=50)

        btn\_n3=Button(cal\_frame,*text*=' 3 ',*command*=*lambda*:btn\_click(3),*font*=('times new roman',18),

*fg*='white',*bg*='green').place(*x*=112,*y*=152,*width*=55,*height*=50)

        btn\_nsub=Button(cal\_frame,*command*=*lambda*:btn\_click('-'),*text*=' - ',*font*=('times new roman',18),

*fg*='white',*bg*='green').place(*x*=167,*y*=162,*width*=55,*height*=55)

        btn\_0=Button(cal\_frame,*text*=' 0 ',*command*=*lambda*:btn\_click(0),*font*=('times new roman',18),

*fg*='white',*bg*='green').place(*x*=2,*y*=202,*width*=55,*height*=50)

        btn\_point=Button(cal\_frame,*text*=' . ',*command*=*lambda*:btn\_click('.'),*font*=('times new roman',18),

*fg*='white',*bg*='green').place(*x*=57,*y*=202,*width*=55,*height*=50)

        btn\_plus=Button(cal\_frame,*text*=' + ',*command*=*lambda*:btn\_click('+'),*font*=('times new roman',18),

*fg*='white',*bg*='green').place(*x*=112,*y*=202,*width*=55,*height*=50)

        btn\_equal=Button(cal\_frame,*text*=' = ',*command*=result,*font*=('times new roman',18),

*fg*='white',*bg*='green').place(*x*=167,*y*=217,*width*=55,*height*=55)

        btn\_c=Button(cal\_frame,*text*=' C ',*command*=clear\_cal,*font*=('times new roman',18),

*fg*='white',*bg*='green').place(*x*=2,*y*=252,*width*=165,*height*=20)

        ##########################Salary Frame############################

*self*.sample=*f*'''\tOrganization Name, UoBs\n\tAddress: Skardu,Baltistan

.......................................................................

Employee ID\t\t:

Salary of\t\t:  Month-Year

Generated On\t\t: DD-MM-YYYY

.......................................................................

Total Days\t\t:  DD

Total Present\t\t:  DD

Total Absent\t\t:  DD

Conveyance\t\t:  Rs.----

Medical\t\t:  Rs.----

PF\t\t:  Rs.----

Gross Payment\t\t:  Rs.-----

Net Salary\t\t:  Rs.-----

.......................................................................

This is computer generated slip, not required    any signature

'''

        sub\_frame3=Frame(frame3,*bd*=5,*relief*=RIDGE, *bg*='green')

        sub\_frame3.place(*x*=240,*y*=0,*width*=400,*height*=280)

        title3=Label(sub\_frame3,*text*='Salary Receipt',*font*=('times new roman bold italic',16),*bg*='darkblue',*fg*='white')

        title3.place(*x*=0,*y*=0,*relwidth*=1)

        sub\_frame4=Frame(sub\_frame3,*bd*=5,*relief*=RIDGE,*bg*='green')

        sub\_frame4.place(*x*=0,*y*=30,*height*=212,*width*=390)

        btn\_print=Button(sub\_frame3,*text*=' Print ',*command*=*self*.print,*font*=('times new roman bold italic',18),

*fg*='black',*bg*='blue')

        btn\_print.place(*x*=270,*y*=240,*width*=100,*height*=25)

        yscrollbar=Scrollbar(sub\_frame4,*orient*=VERTICAL)

        yscrollbar.pack(*fill*=Y,*side*=RIGHT)

*self*.txt\_salary\_receipt=Text(sub\_frame4,*font*=('times new roman bold italic',14),*bg*='green',*fg*='white')

*self*.txt\_salary\_receipt.pack(*fill*=BOTH,*expand*=1)

        yscrollbar.config(*command*=*self*.txt\_salary\_receipt.yview)

*self*.txt\_salary\_receipt.insert('1.0',*self*.sample)

    ###################functions#####################

*def* clear(*self*):

*self*.btn\_save.config(*state*=NORMAL)

*self*.btn\_update.config(*state*=NORMAL)

*self*.btn\_delete.config(*state*=NORMAL)

*self*.entry\_empcode.config(*state*=NORMAL)

*self*.var\_emp\_code.set('')

*self*.var\_Designation.set(' ')

*self*.name.set(' ')

*self*.age.set(' ')

*self*.gender.set(' ')

*self*.email.set(' ')

*self*.hiredloc.set('')

*self*.txt\_address.delete('1.0',END),

*self*.dob.set(' ')

*self*.doj.set(' ')

*self*.experience.set(' ')

*self*.proofid.set(' ')

*self*.contact.set(' ')

*self*.status.set(' ')

*self*.month.set(' ')

*self*.year.set(' ')

*self*.salary.set(' ')

*self*.medical.set(' ')

*self*.pf.set(' ')

*self*.conveyance .set(' ')

*self*.totaldays.set(' ')

*self*.absents.set(' ')

*self*.netsalary.set(' ')

        #self.var\_emp\_code.set()+'.txt'

        #file\_=open('salaryreceipt/'+str(' '),'r')

*self*.txt\_salary\_receipt.delete('1.0',END)

*self*.txt\_salary\_receipt.insert('1.0',*self*.sample)

*def* delete(*self*):

        if *self*.var\_emp\_code.get()=='':

            messagebox.showerror('Error','Employee ID is required')

        else:

            try:

                con=pymysql.connect(*host*='localhost',*user*='root',*db*='employeemanagementsystem')

                cur=con.cursor()

                cur.execute("select \* from emp\_salary where var\_emp\_code=%s",(*self*.var\_emp\_code.get()))

                row=cur.fetchone()

                if row==None:

                    messagebox.showerror('Error','Invalid Employee ID,Please a valid ID')

                else:

                    op=messagebox.askyesno('Confirm','Do you want to delete this record?')

                    if op==True:

                        cur.execute('delete  from emp\_salary where var\_emp\_code=%s',(*self*.var\_emp\_code.get()))

                        command=*self*.clear()

                        con.commit()

                        con.close()

            except Exception as ex:

                messagebox.showerror('Error',*f*'Error due to:{str(ex)}')

*def* search(*self*):

        try:

            con=pymysql.connect(*host*='localhost',*user*='root',*db*='employeemanagementsystem')

            cur=con.cursor()

            cur.execute("select \* from emp\_salary where var\_emp\_code=%s",(*self*.var\_emp\_code.get()))

            row=cur.fetchone()

            if row==None:

                messagebox.showerror('Error','Invalid Employee ID,Please try with another ID')

            else:

                print(row)

*self*.var\_emp\_code.set(row[0])

*self*.var\_Designation.set(row[1])

*self*.name.set(row[2])

*self*.age.set(row[3])

*self*.gender.set(row[4])

*self*.email.set(row[5])

*self*.hiredloc.set(row[6])

                #self.address.set()

*self*.txt\_address.delete('1.0',END)

*self*.txt\_address.insert(END,row[7])

*self*.dob.set(row[8])

*self*.doj.set(row[9])

*self*.experience.set(row[10])

*self*.proofid.set(row[11])

*self*.contact.set(row[12])

*self*.status.set(row[13])

*self*.month.set(row[14])

*self*.year.set(row[15])

*self*.salary.set(row[16])

*self*.medical.set(row[17])

*self*.pf.set(row[18])

*self*.conveyance .set(row[19])

*self*.totaldays.set(row[20])

*self*.absents.set(row[21])

*self*.netsalary.set(row[22])

                #self.var\_emp\_code.set()+'.txt'

                file\_=open("F:/Employee Manangement System/receipt"+str(row[23]),'r') # the file should be opened in read mood

*self*.txt\_salary\_receipt.delete('1.0',END)

                for i in file\_:

*self*.txt\_salary\_receipt.insert('1.0',i)

                file\_.close()

*self*.btn\_save.config(*state*=NORMAL)

*self*.btn\_update.config(*state*=NORMAL)

*self*.btn\_delete.config(*state*=NORMAL)

*self*.entry\_empcode.config(*state*=NORMAL)

        except Exception as ex:

            messagebox.showerror('Error',*f*'Error due to:{str(ex)}')

*def* update(*self*):

        if *self*.var\_emp\_code.get()=='' or *self*.netsalary.get()=='':

                messagebox.showerror('Error','Employee code is required')

        else:

            try:

                con=pymysql.connect(*host*='localhost',*user*='root',*db*='employeemanagementsystem')

                cur=con.cursor()

                cur.execute("select \* from emp\_salary where var\_emp\_code=%s",(*self*.var\_emp\_code.get()))

                row=cur.fetchone()

                if row==None:

                    messagebox.showerror('Error','THis employee is invalid. Try again with another ID')

                else:

                    cur.execute(" UPDATE `emp\_salary` SET `var\_designation`=%s,`name`=%s,`age`=%s,`gender`=%s,`email`=%s,`hiredloc`=%s,`address`=%s,`dob`=%s,`doj`=%s,`experience`=%s,`proofid`=%s,`contact`=%s,`status`=%s,`month`=%s,`year`=%s,`salary`=%s,`medical`=%s,`pf`=%s,`conveyance`=%s,`totaldays`=%s,`absents`=%s,`netsalary`=%s,`receipt`=%s where `var\_emp\_code`=%s",

                                (

*self*.var\_Designation.get(),

*self*.name.get(),

*self*.age.get(),

*self*.gender.get(),

*self*.email.get(),

*self*.hiredloc.get(),

                                 #self.address.get(),

*self*.txt\_address.get('1.0',END),

*self*.dob.get(),

*self*.doj.get(),

*self*.experience.get(),

*self*.proofid.get(),

*self*.contact.get(),

*self*.status.get(),

*self*.month.get(),

*self*.year.get(),

*self*.salary.get(),

*self*.medical.get(),

*self*.pf.get(),

*self*.conveyance .get(),

*self*.totaldays.get(),

*self*.absents.get(),

*self*.netsalary.get(),

*self*.var\_emp\_code.get()+'.txt',

*self*.var\_emp\_code.get()

                                )

                                )

                    con.commit()

                    con.close()

                    file\_=open("F:/Employee Manangement System/receipt"+str(*self*.var\_emp\_code.get())+'.txt','w')

                    file\_.write(*self*.txt\_salary\_receipt.get('1.0',END))

                    file\_.close()

                    messagebox.showinfo('Success','Record Updated Successfully')

            except Exception as ex:

                messagebox.showerror('Error',*f*'Error due to:{str(ex)}')

*def* add(*self*):

        if *self*.var\_emp\_code.get()=='':

                messagebox.showerror('Error','Employee code is required')

        else:

            try:

                con=pymysql.connect(*host*='localhost',*user*='root',*db*='employeemanagementsystem')

                cur=con.cursor()

                cur.execute("select \* from emp\_salary where var\_emp\_code=%s",(*self*.var\_emp\_code.get()))

                row=cur.fetchone()

                if row!=None:

                    messagebox.showerror('Error','THis employee id is already exists')

                else:

                    cur.execute("insert into emp\_salary values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)",

                                (*self*.var\_emp\_code.get(),

*self*.var\_Designation.get(),

*self*.name.get(),

*self*.age.get(),

*self*.gender.get(),

*self*.email.get(),

*self*.hiredloc.get(),

                                 #self.address.get(),

*self*.txt\_address.get('1.0',END),

*self*.dob.get(),

*self*.doj.get(),

*self*.experience.get(),

*self*.proofid.get(),

*self*.contact.get(),

*self*.status.get(),

*self*.month.get(),

*self*.year.get(),

*self*.salary.get(),

*self*.medical.get(),

*self*.pf.get(),

*self*.conveyance .get(),

*self*.totaldays.get(),

*self*.absents.get(),

*self*.netsalary.get(),

*self*.var\_emp\_code.get()+'.txt'

                                )

                                )

                    con.commit()

                    con.close()

                    file\_=open("F:/Employee Manangement System/receipt" +str(*self*.var\_emp\_code.get())+'.txt','w')

                    file\_.write(*self*.txt\_salary\_receipt.get('1.0',END))

                    file\_.close()

                    messagebox.showinfo('Success','Record added Successfully')

            except Exception as ex:

                messagebox.showerror('Error',*f*'Error due to:{str(ex)}')

*def* calculate(*self*):

        if *self*.month.get()==''or *self*.salary.get()=='' or *self*.year.get() =='' or *self*.totaldays.get()=='':

            messagebox.showerror('Error','All fields are required')

        else:

            per\_day=int(*self*.salary.get())/int(*self*.totaldays.get())

            work\_day=int(*self*.totaldays.get())-int(*self*.absents.get())

            sal\_per=per\_day\*work\_day

            add1=int(*self*.medical.get())+int(*self*.pf.get())

            add2=int(*self*.conveyance.get())

            net\_sal=add1+add2+sal\_per

*self*.netsalary.set(str(round(net\_sal,2)))

            #########receipt########################

            newsample=*f*'''\tOrganization Name, UoBs\n\tAddress: Skardu,Baltistan

.......................................................................

Employee ID\t\t:  {*self*.var\_emp\_code.get()}

Salary of\t\t:  {*self*.month.get()}-{*self*.year.get()}

Generated On\t\t: {str(time.strftime('%d-%m-%Y'))}

.......................................................................

Total Days\t\t:  {*self*.totaldays.get()}

Total Present\t\t:  {str(int(*self*.totaldays.get())-int(*self*.absents.get()))}

Total Absent\t\t:  {*self*.absents.get()}

Conveyance\t\t:  Rs.{*self*.conveyance.get()}

Medical\t\t:  Rs.{*self*.medical.get()}

PF\t\t:  Rs.{*self*.pf.get()}

Gross Payment\t\t:  Rs.{*self*.salary.get()}

Net Salary\t\t:  Rs{*self*.netsalary.get()}

......................................................................

This is computer generated slip, not required

any signature

'''

*self*.txt\_salary\_receipt.delete('1.0',END)

*self*.txt\_salary\_receipt.insert('1.0',newsample)

*def* employeeframe(*self*):

*self*.root2=Toplevel(*self*.root)

*self*.root2.title('Employee Payroll Management System | By UoBs Students')

*self*.root2.geometry('900x500+60+80')

        title1=Label(*self*.root2,*text*='Employee Details',*font*=('times new roman',20),

*bg*='black',*fg*='white').pack(*side*=TOP,*fill*=X)

*self*.root2.focus\_force()

        scrolly=Scrollbar(*self*.root2,*orient*=VERTICAL)

        scrollx=Scrollbar(*self*.root2,*orient*=HORIZONTAL)

        scrollx.pack(*side*=BOTTOM,*fill*=X)

        scrolly.pack(*side*=RIGHT,*fill*=Y)

*self*.employee\_tree=ttk.Treeview(*self*.root2,*columns*=('empcode', 'designation', 'name', 'age', 'gender', 'email', 'hiredloc',

                                                            'address', 'dob', 'doj', 'experience',

                                                            'proofid', 'contact', 'status',

                                                            'month', 'year', 'salary', 'medical',

                                                            'pf', 'conveyance', 'totaldays', 'absent',

                                                            'netsalary', 'receipt'),*yscrollcommand*=scrolly.set,*xscrollcommand*=scrollx.set)

*self*.employee\_tree.heading('empcode' ,*text*='Emp ID')

*self*.employee\_tree.heading('designation' ,*text*='Designation')

*self*.employee\_tree.heading('name' ,*text*='Name')

*self*.employee\_tree.heading('age' ,*text*='Age')

*self*.employee\_tree.heading('gender' ,*text*='Gender')

*self*.employee\_tree.heading('email' ,*text*='Email')

*self*.employee\_tree.heading('hiredloc' ,*text*='Hired location')

*self*.employee\_tree.heading('address' ,*text*='Address')

*self*.employee\_tree.heading('dob' ,*text*='Birth Date')

*self*.employee\_tree.heading('doj' ,*text*='Joining Date')

*self*.employee\_tree.heading('experience' ,*text*='Experience')

*self*.employee\_tree.heading('proofid' ,*text*='Proof ID')

*self*.employee\_tree.heading('contact' ,*text*='Contact')

*self*.employee\_tree.heading('status' ,*text*='Status')

*self*.employee\_tree.heading('month' ,*text*='Month')

*self*.employee\_tree.heading('year' ,*text*='Year')

*self*.employee\_tree.heading('salary' ,*text*='Basic Salary')

*self*.employee\_tree.heading('medical' ,*text*='Medical Allowance')

*self*.employee\_tree.heading('pf' ,*text*='PF')

*self*.employee\_tree.heading('conveyance' ,*text*='Conveyance Allowance')

*self*.employee\_tree.heading('totaldays' ,*text*='Total Days')

*self*.employee\_tree.heading('absent' ,*text*='Absents')

*self*.employee\_tree.heading('netsalary' ,*text*='NetSalary')

*self*.employee\_tree.heading('receipt' ,*text*='Salary Receipt')

*self*.employee\_tree['show']='headings'

*self*.employee\_tree.column('empcode',*width*=70)

*self*.employee\_tree.column('designation',*width*=70)

*self*.employee\_tree.column('name',*width*=70)

*self*.employee\_tree.column('age',*width*=30)

*self*.employee\_tree.column('gender',*width*=70)

*self*.employee\_tree.column('email',*width*=70)

*self*.employee\_tree.column('hiredloc',*width*=100)

*self*.employee\_tree.column('address',*width*=70)

*self*.employee\_tree.column('dob',*width*=70)

*self*.employee\_tree.column('doj',*width*=70)

*self*.employee\_tree.column('experience' ,*width*=100)

*self*.employee\_tree.column('proofid' ,*width*=100)

*self*.employee\_tree.column('contact' ,*width*=70)

*self*.employee\_tree.column('status' ,*width*=70)

*self*.employee\_tree.column('month' ,*width*=70)

*self*.employee\_tree.column('year' ,*width*=70)

*self*.employee\_tree.column('salary' ,*width*=70)

*self*.employee\_tree.column('medical' ,*width*=120)

*self*.employee\_tree.column('pf' ,*width*=70)

*self*.employee\_tree.column('conveyance' ,*width*=155)

*self*.employee\_tree.column('totaldays' ,*width*=120)

*self*.employee\_tree.column('absent' ,*width*=70)

*self*.employee\_tree.column('netsalary' ,*width*=150)

*self*.employee\_tree.column('receipt' ,*width*=150)

        scrollx.config(*command*=*self*.employee\_tree.xview)

        scrolly.config(*command*=*self*.employee\_tree.yview)

*self*.employee\_tree.pack(*fill*=BOTH,*expand*=1)

*self*.show()

*self*.root2.mainloop()

*def* show(*self*):

            try:

                con=pymysql.connect(*host*='localhost',*user*='root',*db*='employeemanagementsystem')

                cur=con.cursor()

                cur.execute("select \* from emp\_salary")

                rows=cur.fetchall()

*self*.employee\_tree.delete(\**self*.employee\_tree.get\_children())

                for row in rows:

*self*.employee\_tree.insert('',END,*values*=row)

                con.close()

            except Exception as ex:

                    messagebox.showerror('Error',*f*'Error due to:{str(ex)}')

*def* print(*self*):

        file\_=tempfile.mktemp(".txt")

        open(file\_,'w').write(*self*.txt\_salary\_receipt.get('1.0',END))

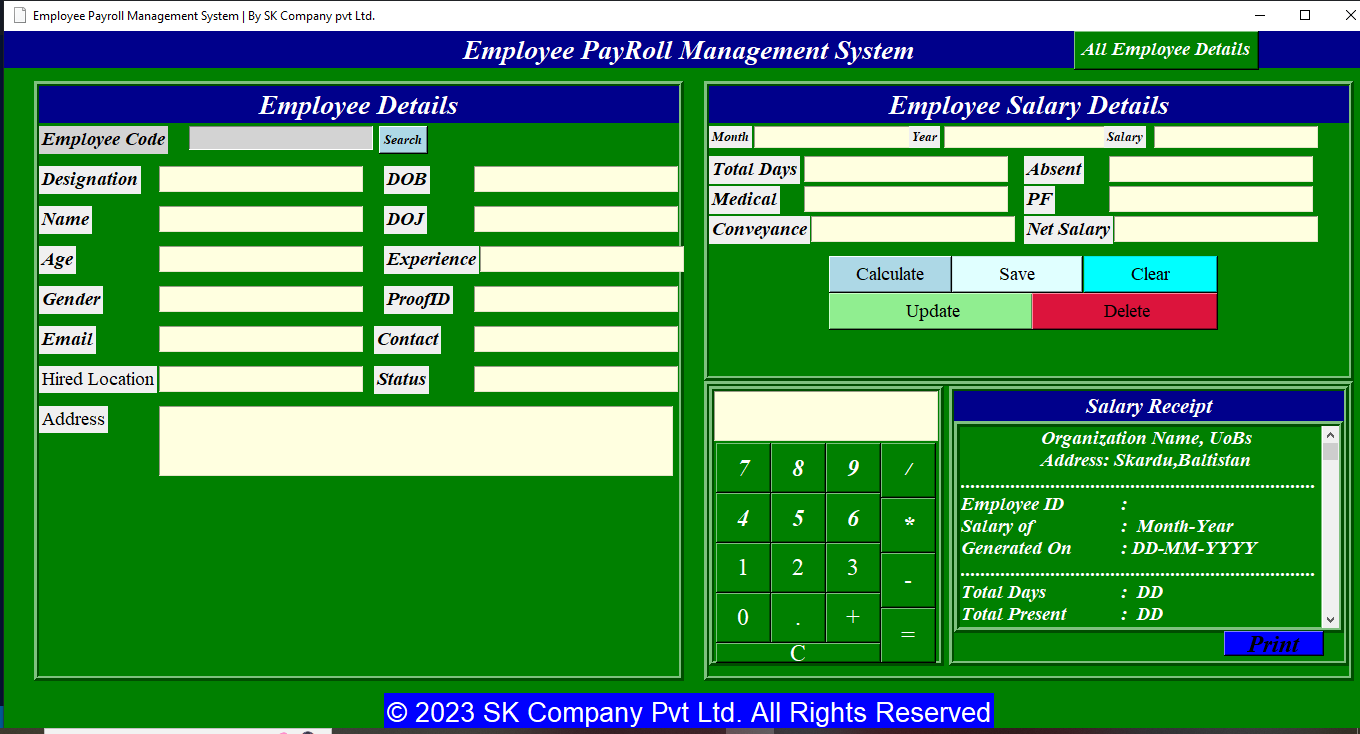
        os.startfile(file\_,'PRINT')

rt=Tk()

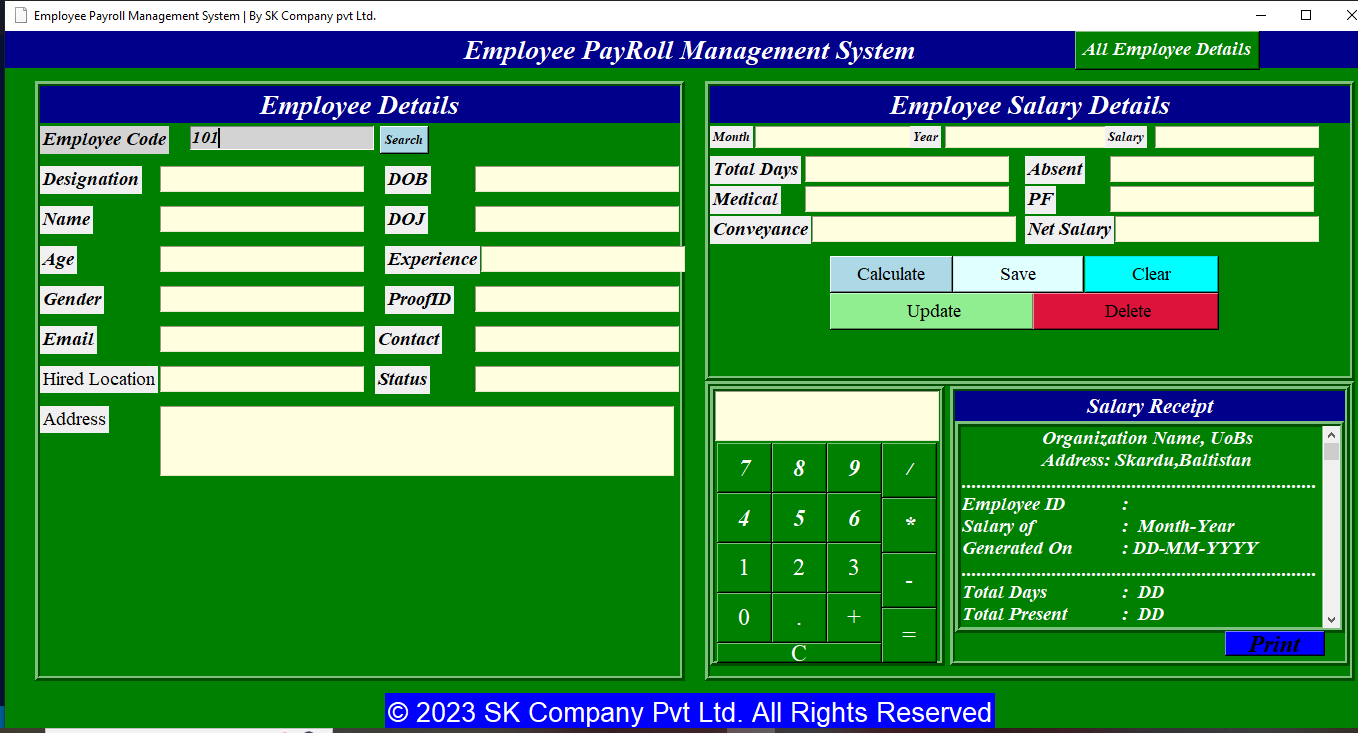
obj=employeesystem(rt)

rt.mainloop()

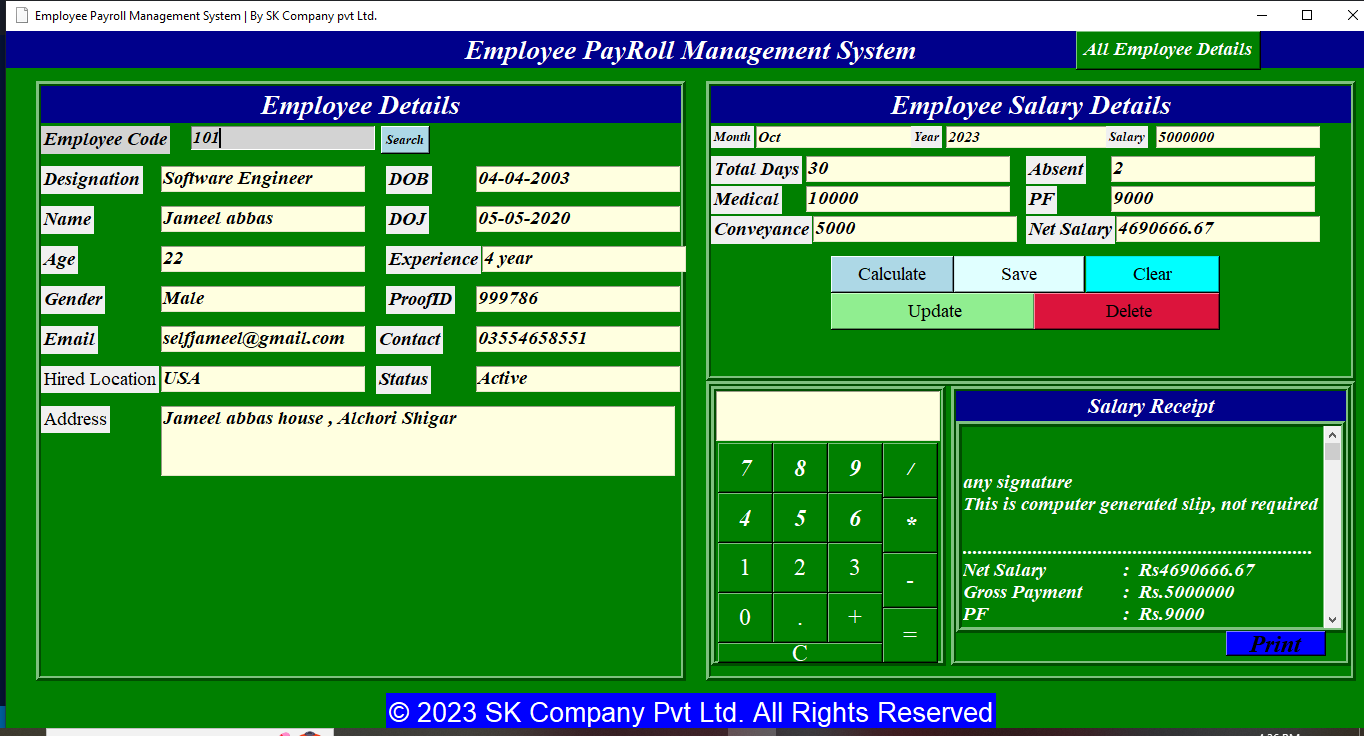
**Run This code We have:**



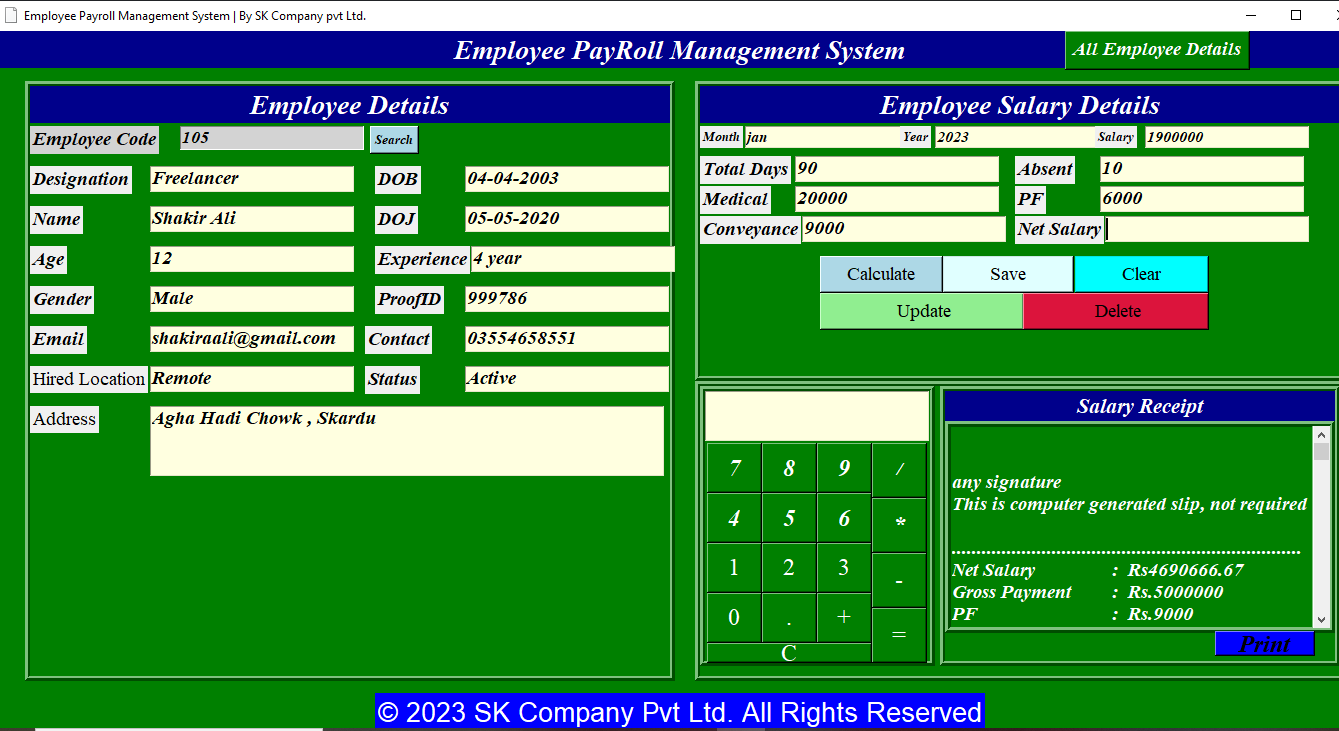
**Search emp\_code 101 to get all details:**



**Details of 101:**



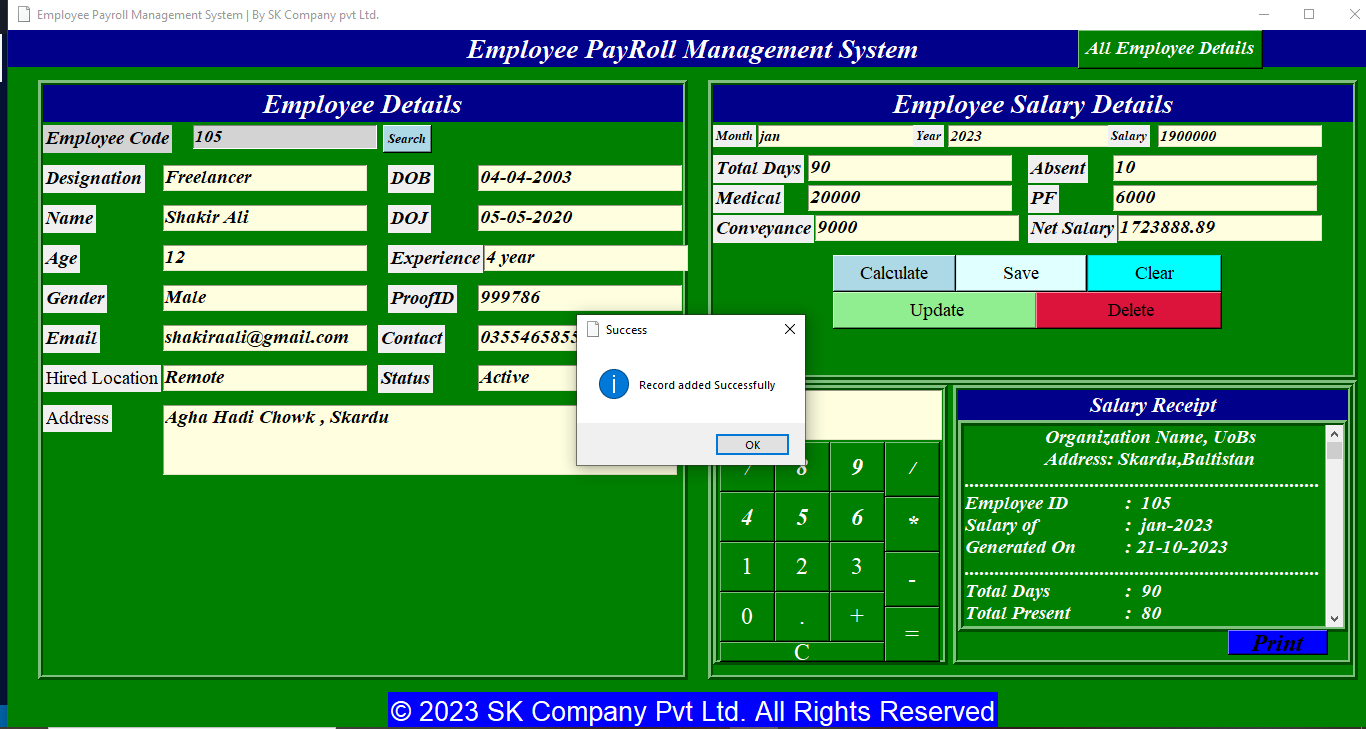
**Add new user to database and calculate net salary:**



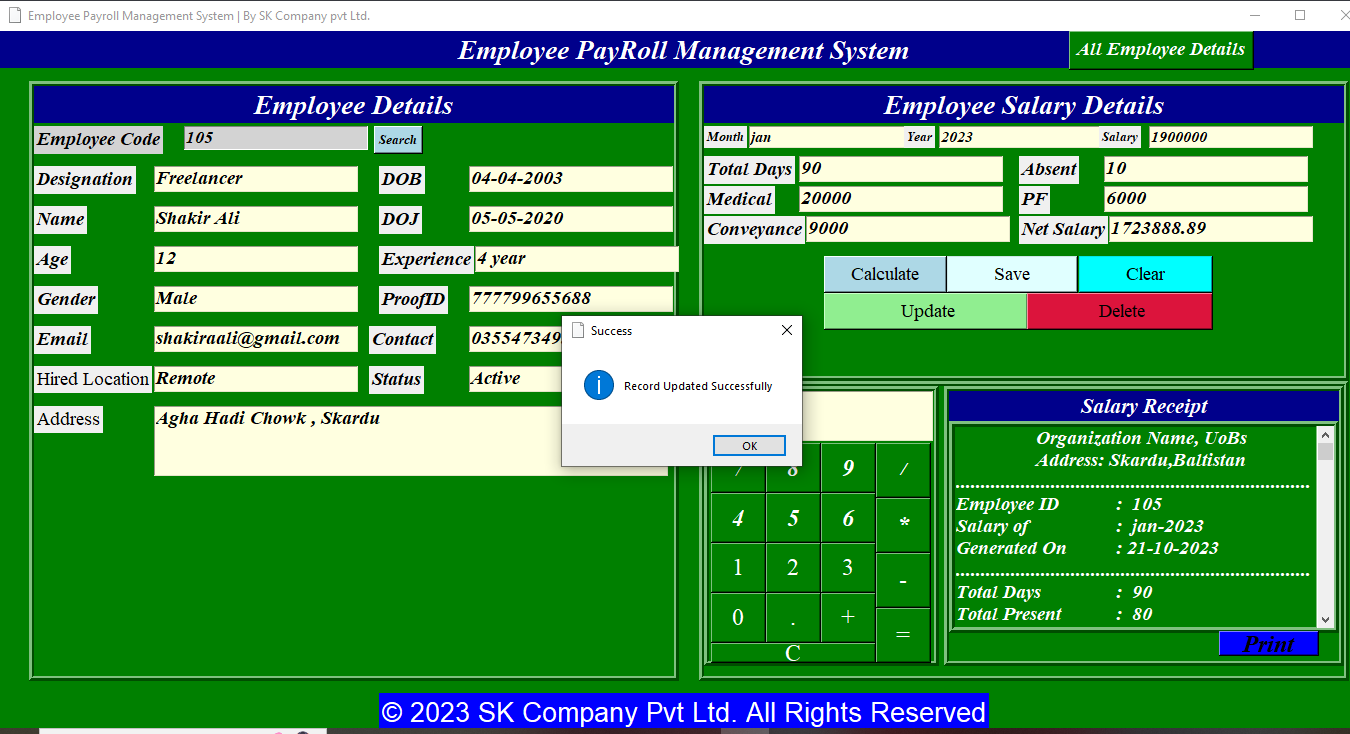
**Calculate Net Salary:**



**Save User details in database:**



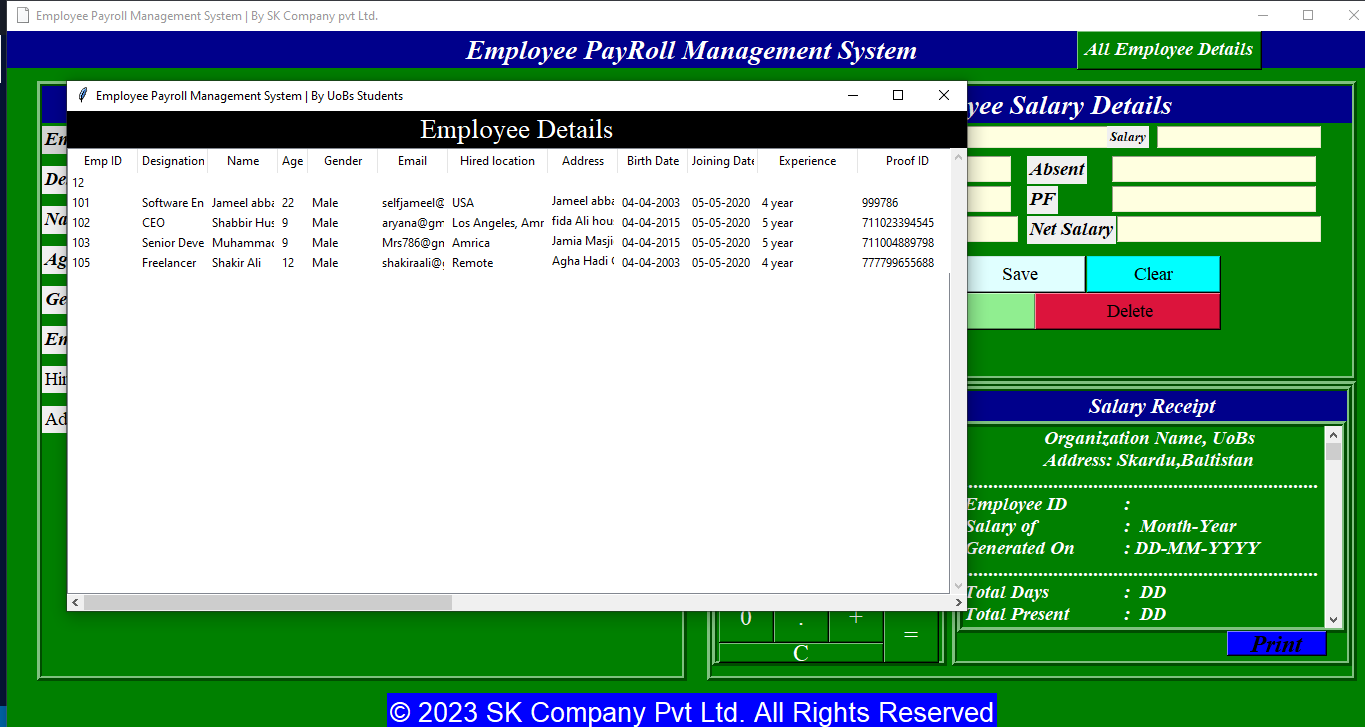
**Updating User values we update phone number:**



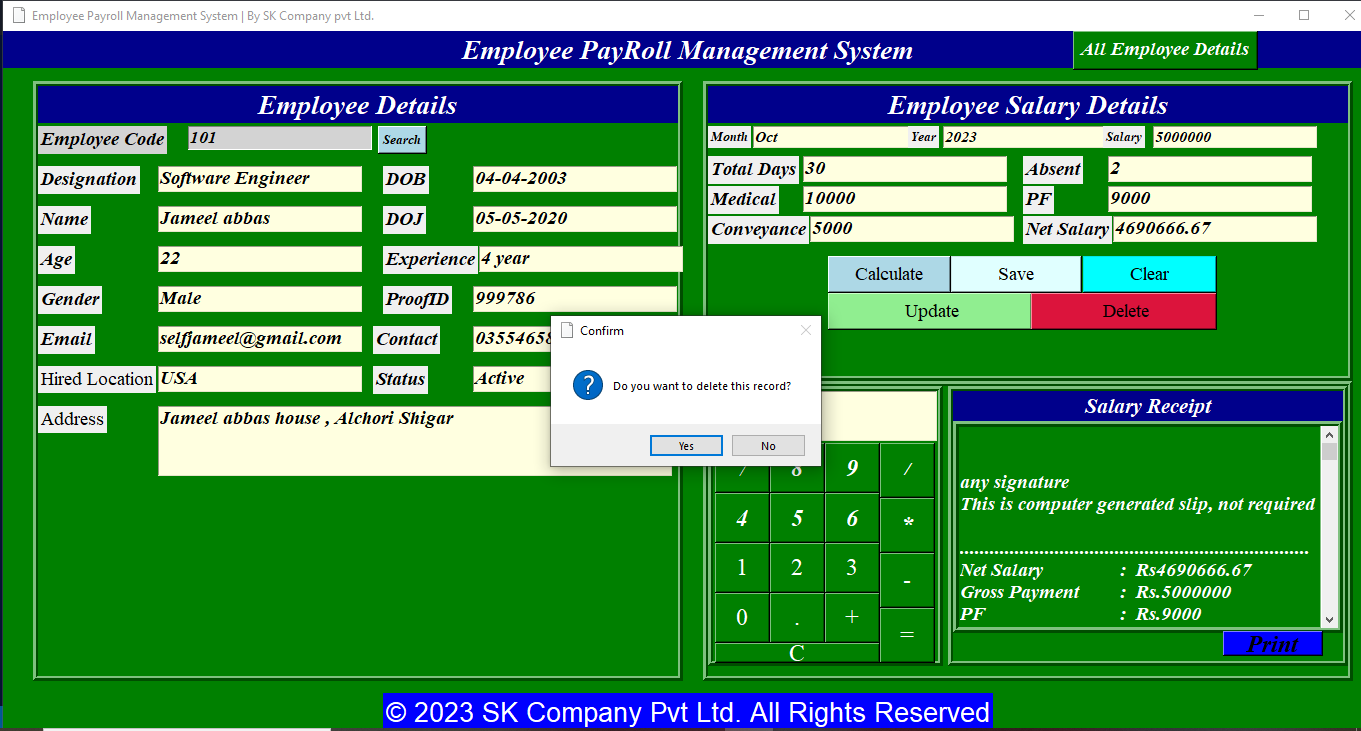
**Clear all input fields:**



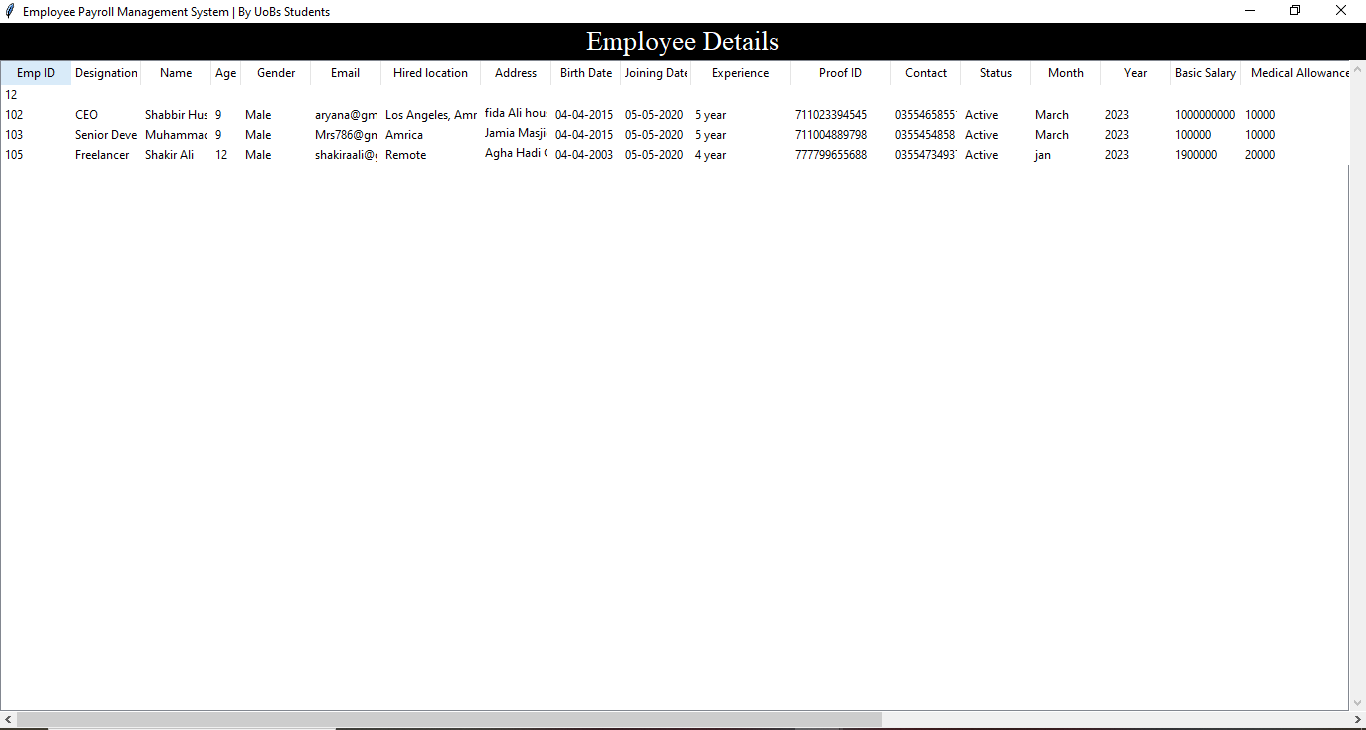
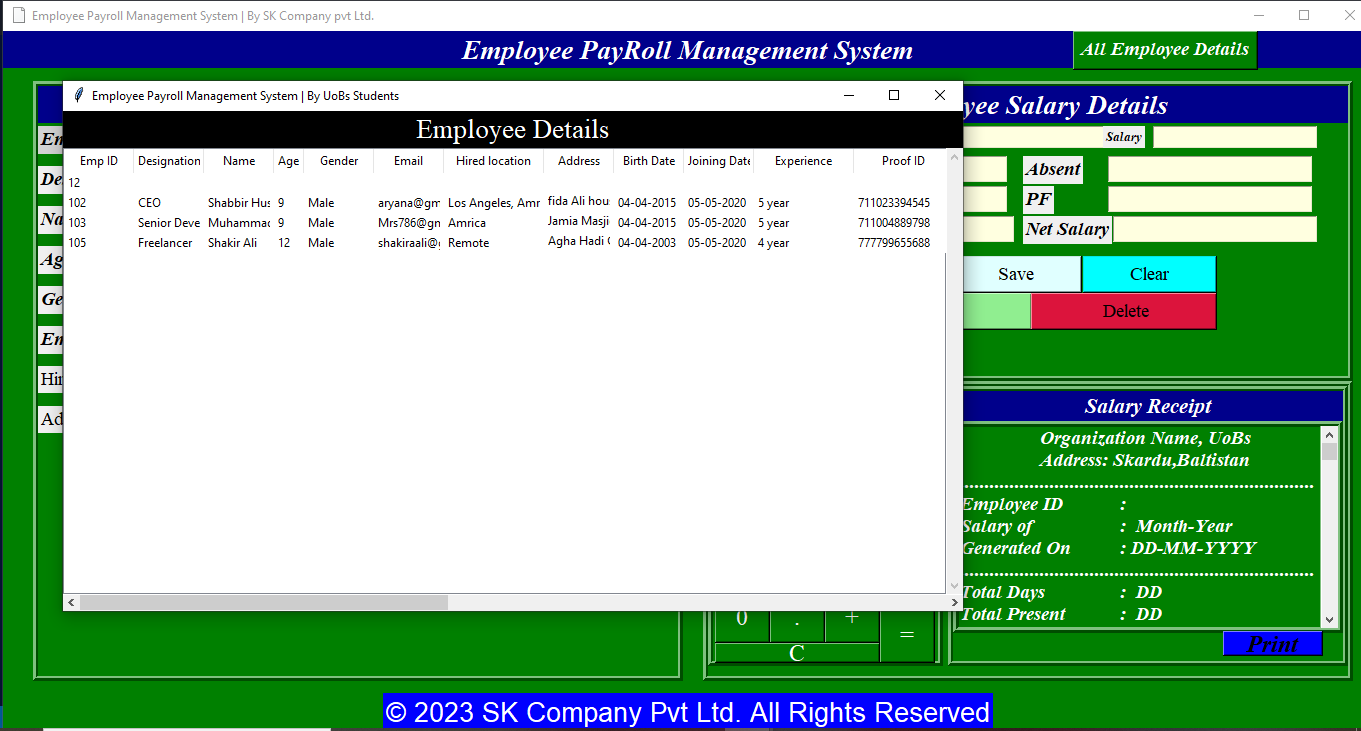
**All Employee Details :**



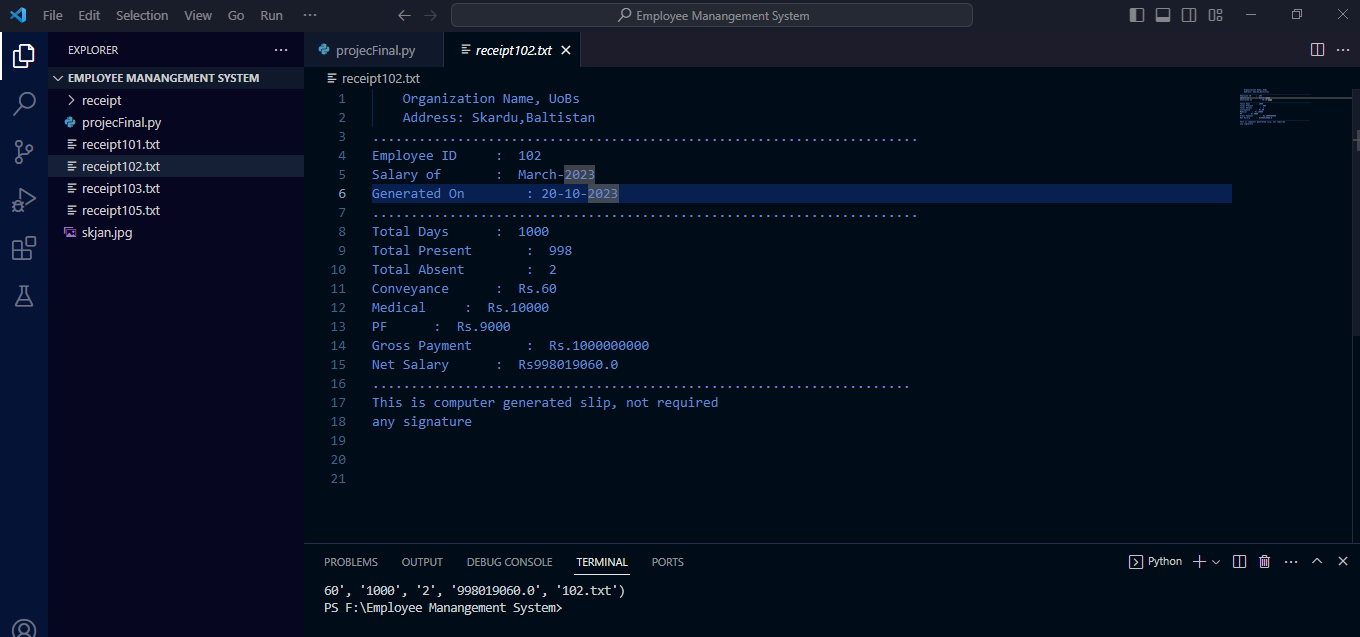
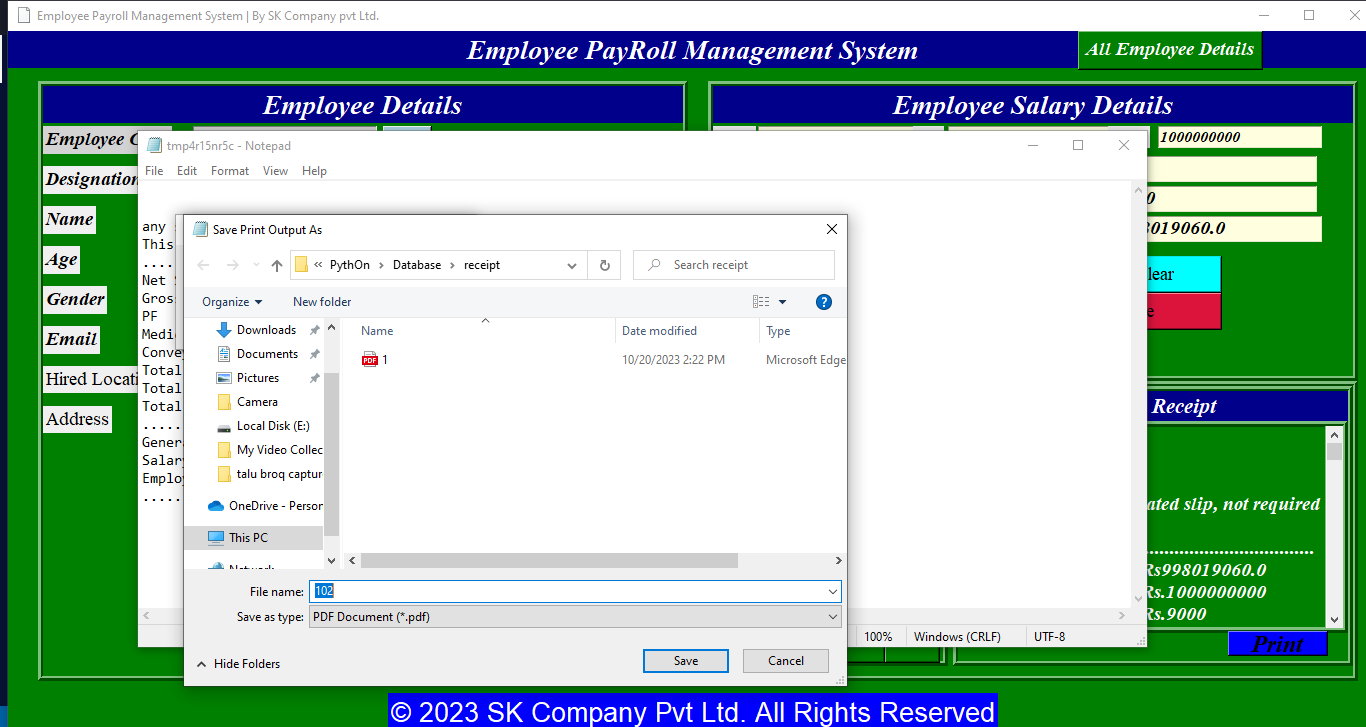
**Deleting User 101:**



**After deleting employee emp\_code 101, we have**



**Print Net salary of employee:**



**Working of calculator:**

