Term Project: Employee Payroll Database System

3444 Emerging Technologies

Mehrnoush Ashrafi

Submitted by:

Gaurav Dhawan, c0819676
Manpreet singh, c0823242
Adil sharma, c0813277
Riya Chaudhari, c0820094
Ramneet kaur, c0822287
Prabhjot kaur, c0815342

Contents

INTRODUCTION

"Employee Database And Payroll Management System" is designed to make the existing manual system automatic with the help of computerised equipment and full-edged computer software, fulfilling their requirements, so that their valuable data and information can be stored for a longer period with easy access and manipulation of the same. The required software is easily available and easy to work with. This application can maintain and view computerized records without getting redundant entries. The project describes how to manage user data for good performance and provide better services for the client.

Purpose

The purpose of this document is to describe the functionality and specifications of the design of a web application for Managing Employees and their payroll. The expected audiences of this document are the developers and the admin of the web application. Now with the help of this system the admin has the information on his finger tips and can easily prepare a good record based on their requirements.

Finally, we can say that this system will not only automate the process but save the valuable time of the manager or the admin, which can be well utilized buy his institute. This will be an additional advantage and management of power based on their free time from his normal duty

Software requirements:

Language/s Used:	Python (GUI) Based
Python version (Recommended):	2.x or 3.x
Database:	MySQL

FEATURES

There are seven features in this application:

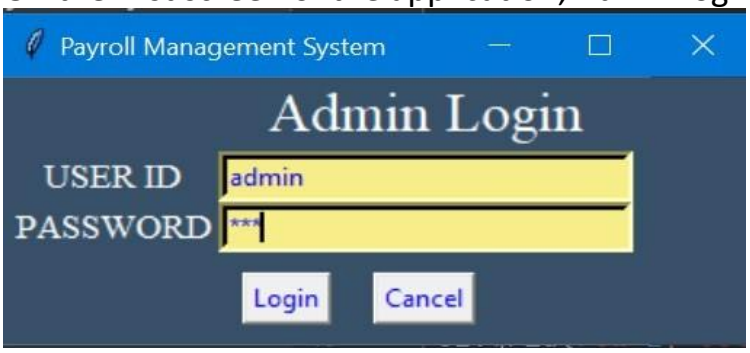
- Employee
    - Add
    - Update
    - Delete
    - View

- Salary(Payroll)
    - Add
    - Delete
    - Update
    - View

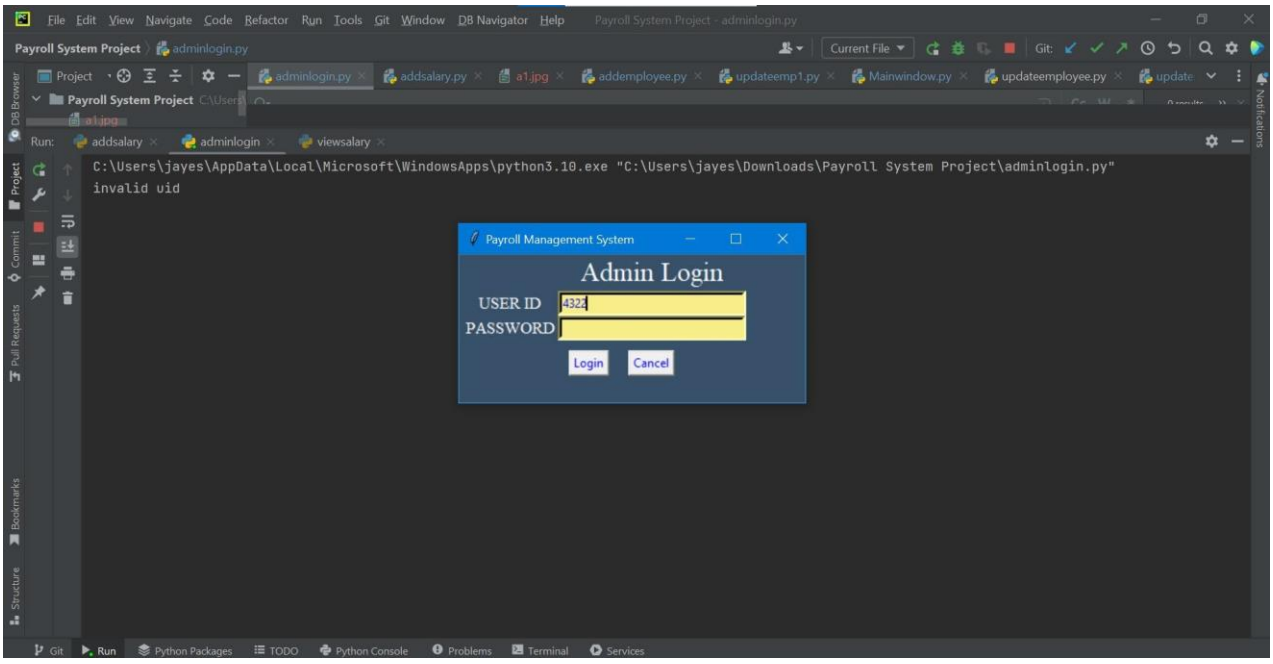
APPLICATION OPERATION

Admin Login:

On the first screen of the application, Admin login Page



The app will report that it cannot proceed if an invalid username or password is entered.




```
import os
from tkinter import *

import pymysql
import pymysql.cursors


def search_id():
    conn = pymysql.connect(host='localhost', user='root', db='payroll')
    a = conn.cursor()
    uid = e1.get()
    pwd = e2.get()
```

```
    sql = "SELECT * FROM admin where uid='%s' and  pass='%s'" % (uid, pwd)
    try:
        a.execute(sql)
        p = a.rowcount
        if p >= 1:
            os.system('python MainWindow.py')
            print("ok")
        else:
            print("invalid uid")
    except:
        print("ERROR")
    a.close()
    conn.close()


master = Tk()

master.title("Payroll Management System")
master.configure(bg="#375068")
master.geometry("350x150")

Label(master, text="Admin Login", fg="white", bg="#375068", font=("times",
20)).grid(column=1)
Label(master, text=" USER ID", fg="white", bg="#375068", font=("times",
13)).grid(row=1)
Label(master, text=" PASSWORD", fg="white", bg="#375068", font=("times",
13)).grid(row=2)

e1 = Entry(master, width=30, bd=4, fg="blue", bg="#F7EE89")
e2 = Entry(master, width=30, show="*", bd=4, fg="blue", bg="#F7EE89")
e1.grid(row=1, column=1)
e2.grid(row=2, column=1)
Button(master, text="Login", fg="blue", command=search_id).grid(row=4,
sticky=W,column=1, padx=11, pady=10)
Button(master, text="Cancel", fg="blue", command=master.quit).grid(row=4,
column=1, padx=3, pady=10)
mainloop()
```

Home page :

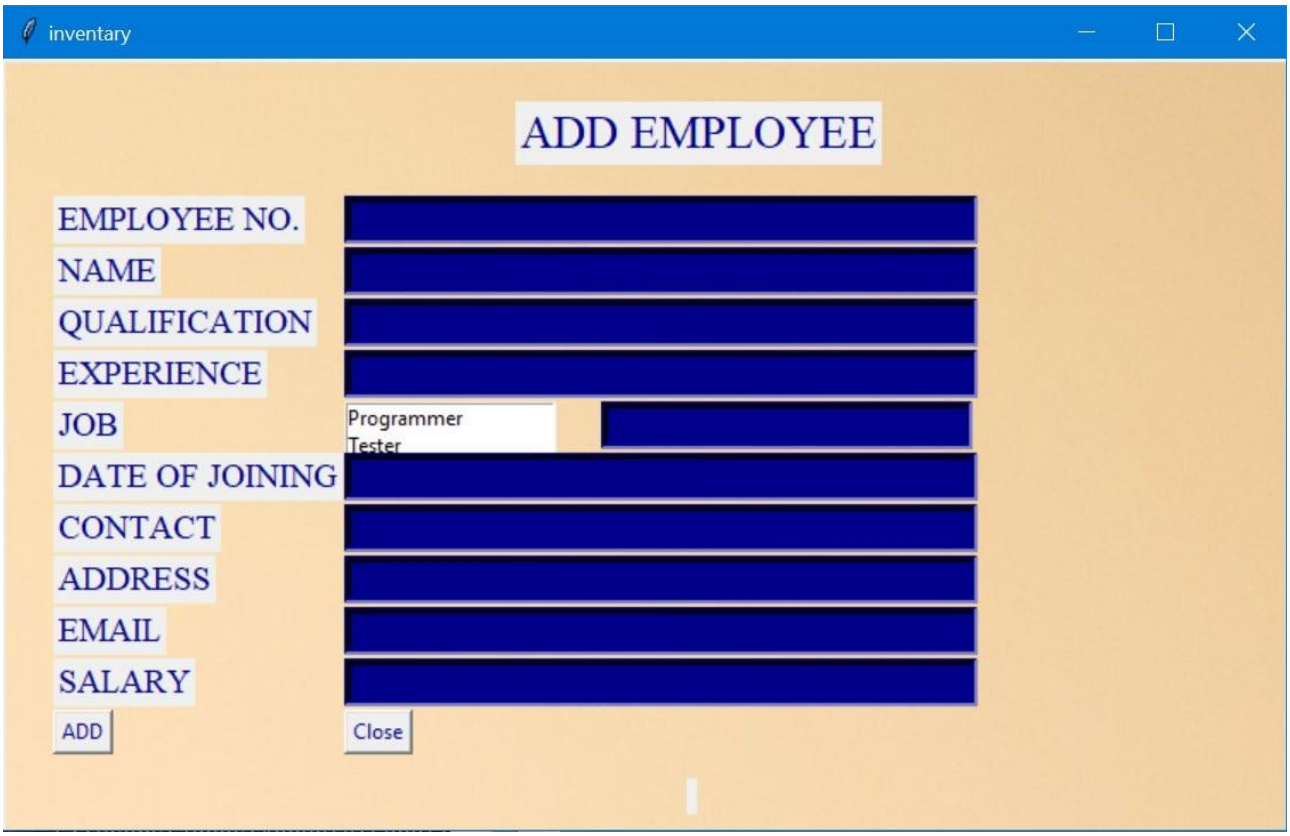
On the home page, as an administrator, you have access to all the functions of the application, as detailed.

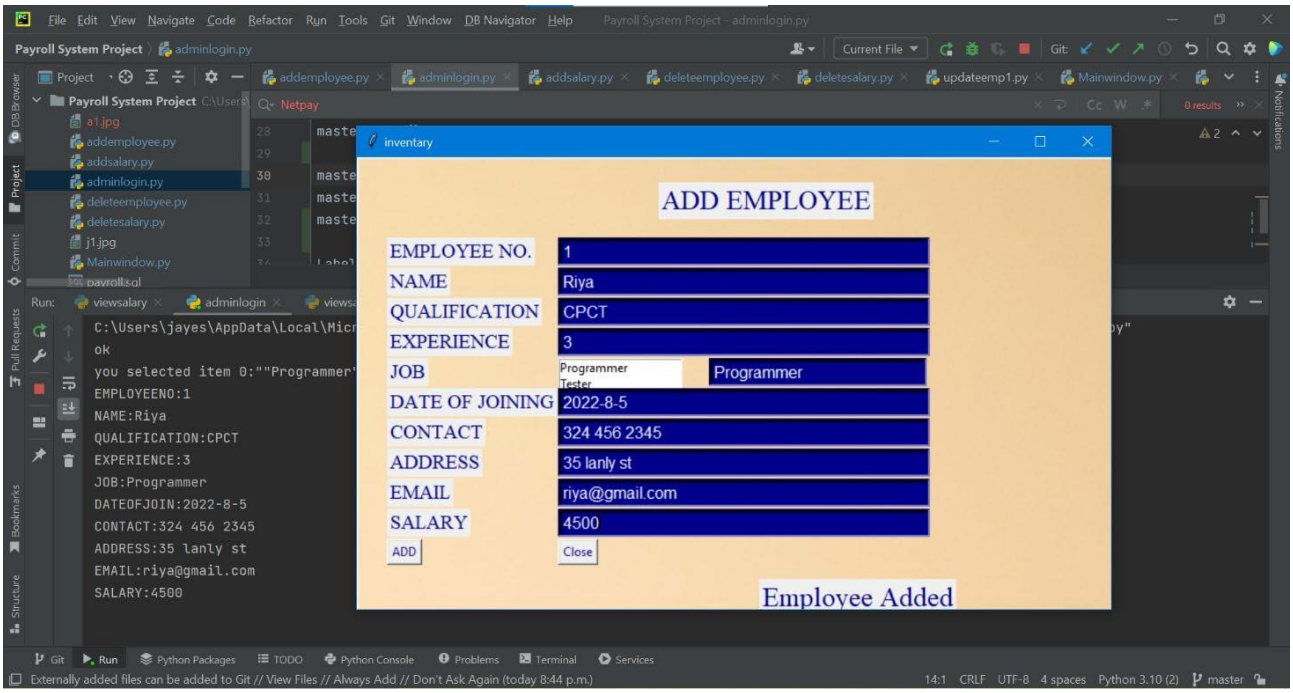

```
from tkinter import *
from tkinter.filedialog import askopenfile
from tkinter.messagebox import showerror
from PIL import ImageTk, Image
import sys, os


def AddEmployee():
    os.system('AddEmployee.py')
    print("Add Employee")


def DeleteEmployee():
    os.system('DeleteEmployee.py')


def UpdateEmp():
    os.system('UpdateEmp1.py')


def ViewEmployee():
    os.system('ViewEmployee.py')


def AddSalary():
    os.system('AddSalary.py')


def DeleteSalary():
    os.system('DeleteSalary.py')


def UpdateSalary():
    os.system('UpdateSalary.py')
```

```
def ViewSalary():
    os.system('ViewSalary.py')


root = Tk()
menu = Menu(root)
root.config(menu=menu)
root.title("Payroll Management System")

employeemenu = Menu(menu)
menu.add_cascade(label="Employee", menu=employeemenu)
employeemenu.add_command(label="Add Employee", command=AddEmployee)
employeemenu.add_command(label="Delete Employee", command=DeleteEmployee)
employeemenu.add_command(label="Update Employee", command=UpdateEmp)
employeemenu.add_command(label="View Employee", command=ViewEmployee)
employeemenu.add_command(label="Exit", command=root.quit)
salarymenu = Menu(menu)
menu.add_cascade(label="Salary", menu=salarymenu)
salarymenu.add_command(label="Add Salary", command=AddSalary)
salarymenu.add_command(label="Delete Salary", command=DeleteSalary)
salarymenu.add_command(label="Update Salary", command=UpdateSalary)
salarymenu.add_command(label="View Salary", command=ViewSalary)
salarymenu.add_command(label="Exit", command=root.quit)

img = Image.open('j1.jpg')
bg = ImageTk.PhotoImage(img)
root.geometry("1200x450")
label = Label(root, image=bg)
label.place(x=0, y=0)
label2 = Label(root, text="Payroll Management System", fg="Blue",
               font=("Arial Black", 40))
label2.pack(pady=10)
mainloop()
```

Employee menu :

Add Employee :





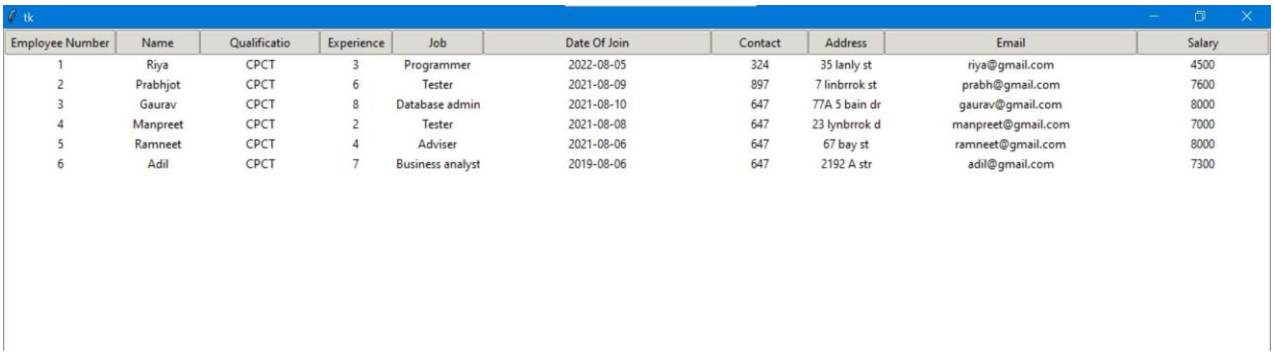

```
from tkinter import *
from PIL import ImageTk, Image
import pymysql
import pymysql.cursors
```

```
def onselect(evt):
    # q1
    global job
    w = evt.widget
    index = int(w.curselection()[0])
    value = w.get(index)
    standard = value
    e101.delete(0, END)
    e101.insert(END, standard)
    print('you selected item %d:""%s"' % (index, value))

def add_fields():
    conn = pymysql.connect(host='localhost', user='root', db='payroll')
    a = conn.cursor()
    a1 = e1.get()
    a2 = e2.get()
    a3 = e3.get()
    a4 = e4.get()
    a5 = e101.get()
    a6 = e6.get()
    a7 = e7.get()
    a8 = e8.get()
    a9 = e9.get()
    a10 = e10.get()
    insertstmt = (
            "insert into
employee(employeeno,name,qualification,experience,job,dateofjoin,contact,addres
s,email,"
            "salary) values('%s','%s','%s','%s','%s','%s','%s','%s','%s','%s')"
% (
                a1, a2, a3, a4, a5, a6, a7, a8, a9, a10))
    a.execute(insertstmt)
    L1.config(text="Employee Added", fg="#00008b", font=("times", 20))
    conn.commit()
    a.close()
    conn.close()
    print(

"EMPLOYEENO:%s\nNAME:%s\nQUALIFICATION:%s\nEXPERIENCE:%s\nJOB:%s\nDATEOFJOIN:%s
\nCONTACT:%s\nADDRESS:%s"
        "\nEMAIL:%s\nSALARY:%s" % (
            e1.get(), e2.get(), e3.get(), e4.get(), e101.get(), e6.get(),
e7.get(), e8.get(), e9.get(), e10.get()))


def PRODUCT():

    k1.insert(END, "Programmer")
    k1.insert(END, "Tester")
    k1.insert(END, "Database administrator")
    k1.insert(END, "Web Developer")



master = Tk()
master.title("inventary")
img = Image.open('a1.jpg')
bg = ImageTk.PhotoImage(img)
master.geometry("750x450")
label3 = Label(master, image=bg)
label3.place(x=0, y=0)
```

```
label = Label(master, text="ADD EMPLOYEE", fg="#00008b", font=("times", 20))
label.place(x=300, y=25)
label1 = Label(master, text="EMPLOYEE NO.", fg="#00008b", font=("times", 15))
label1.place(x=30, y=80)
label2 = Label(master, text="NAME", fg="#00008b", font=("times", 15))
label2.place(x=30, y=110)
label3 = Label(master, text="QUALIFICATION", fg="#00008b", font=("times", 15))
label3.place(x=30, y=140)
label4 = Label(master, text="EXPERIENCE", fg="#00008b", font=("times", 15))
label4.place(x=30, y=170)
label5 = Label(master, text="JOB", fg="#00008b", font=("times", 15))
label5.place(x=30, y=200)

label6 = Label(master, text="DATE OF JOINING", fg="#00008b", font=("times",
15))
label6.place(x=30, y=230)  # Format year-mon-day
label7 = Label(master, text="CONTACT", fg="#00008b", font=("times", 15))
label7.place(x=30, y=260)
label8 = Label(master, text="ADDRESS", fg="#00008b", font=("times", 15))
label8.place(x=30, y=290)
label9 = Label(master, text="EMAIL", fg="#00008b", font=("times", 15))
label9.place(x=30, y=320)

label10 = Label(master, text="SALARY", fg="#00008b", font=("times", 15))
label10.place(x=30, y=350)
e1 = Entry(master, width=40, bd=4, fg="white", bg="#00008b", font="italic",
selectbackground="cyan")
e2 = Entry(master, width=40, bd=4, fg="white", bg="#00008b", font="italic",
selectbackground="cyan")
e3 = Entry(master, width=40, bd=4, fg="white", bg="#00008b", font="italic",
selectbackground="cyan")
e4 = Entry(master, width=40, bd=4, fg="white", bg="#00008b", font="italic",
selectbackground="cyan")
k1 = Listbox(master, height=3)
k1.place(x=200, y=200)
k1.bind("<Double-1>", onselect)
e6 = Entry(master, width=40, bd=4, fg="white", bg="#00008b", font="italic",
selectbackground="cyan")
e7 = Entry(master, width=40, bd=4, fg="white", bg="#00008b", font="italic",
selectbackground="cyan")
e8 = Entry(master, width=40, bd=4, fg="white", bg="#00008b", font="italic",
selectbackground="cyan")
e9 = Entry(master, width=40, bd=4, fg="white", bg="#00008b", font="italic",
selectbackground="cyan")
e10 = Entry(master, width=40, bd=4, fg="white", bg="#00008b", font="italic",
selectbackground="cyan")

e1.place(x=200, y=80)
e2.place(x=200, y=110)
e3.place(x=200, y=140)
e4.place(x=200, y=170)
e101 = Entry(master, width=23, bd=4, fg="white", bg="#00008b", font="italic",
selectbackground="cyan")
e101.place(x=350, y=200)
e6.place(x=200, y=230)
e7.place(x=200, y=260)
e8.place(x=200, y=290)
e9.place(x=200, y=320)
e10.place(x=200, y=350)

Button(master, text="ADD", fg="#00008b", command=add_fields).place(x=30, y=380)
```

```
Button(master, text="Close", fg="#00008b", command=master.quit).place(x=200,
y=380)

L1 = Label(master, text="")
L1.place(x=400, y=420)
PRODUCT()
mainloop()
```

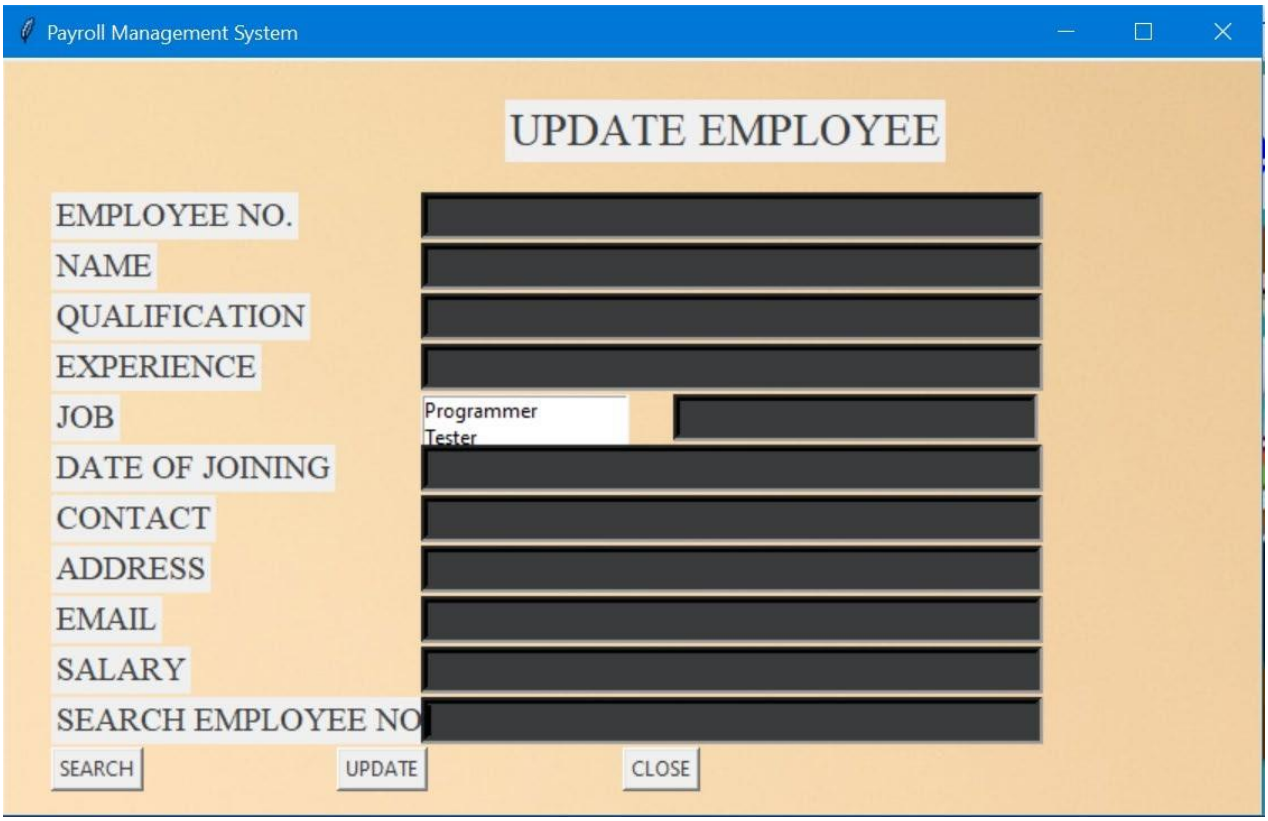
View Employee :

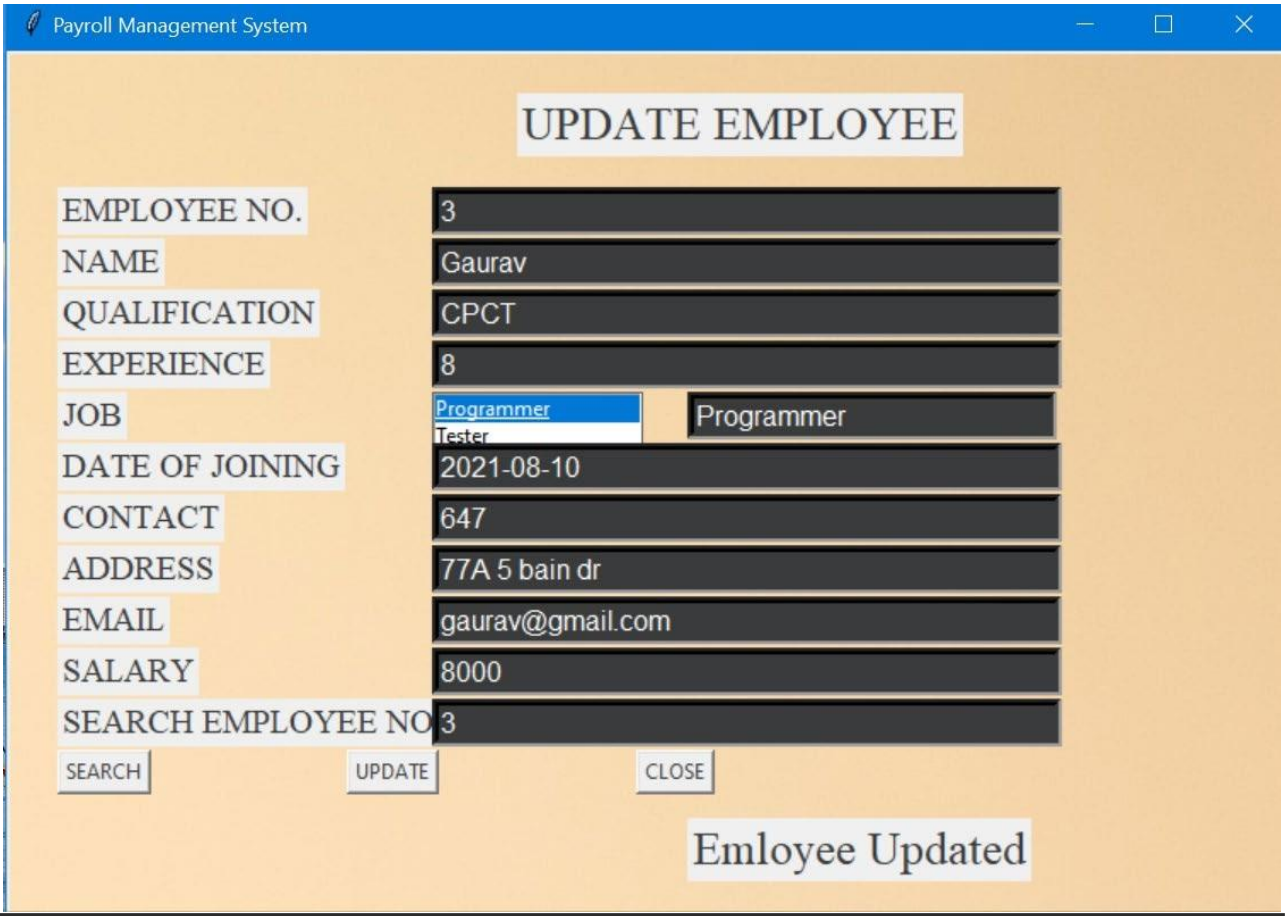



```
from tkinter import *
from tkinter import ttk
import pymysql
import pymysql.cursors
win = Tk()
win.geometry("750x450")
style = ttk.Style()
style.theme_use('clam')
tree = ttk.Treeview(win, column=("Employee Number", "Name",
"Qualification","Experience","Job","Date Of
Join","Contact","Address","Email","Salary"), show='headings', height=15)
tree.column("# 1", anchor=CENTER)
tree.heading("# 1", text="Employee Number")
tree.column("# 2", anchor=CENTER)
tree.heading("# 2", text="Name")
tree.column("# 3", anchor=CENTER)
tree.heading("# 3", text="Qualificatio")
tree.column("# 4", anchor=CENTER)
tree.heading("#4", text="Experience")
tree.column("# 5", anchor=CENTER)
tree.heading("#5", text="Job")
tree.column("# 6", anchor=CENTER)
tree.heading("#6", text="Date Of Join")
tree.column("# 7", anchor=CENTER)
tree.heading("#7", text="Contact")
tree.column("# 8", anchor=CENTER)
tree.heading("#8", text="Address")
tree.column("# 9", anchor=CENTER)
tree.heading("#9", text="Email")
tree.column("# 10", anchor=CENTER)
tree.heading("#10", text="Salary")
conn=pymysql.connect(host='localhost',user='root',db='payroll')
a=conn.cursor()
sql="SELECT * FROM employee"
try:
```

```
    numrows=a.execute(sql)
    results=a.fetchall()
    for row in results:
        empno=row[0]
        name=row[1]
        qlfn=row[2]
        exp=row[3]
        job=row[4]
        doj=row[5]
        contact=row[6]
        address=row[7]
        email=row[8]
        salary=row[9]
        tree.insert('','end', text="1",
values=(empno,name,qlfn,exp,job,doj,contact,address,email,salary))
except:
            print("ERROR")
a.close()
conn.close()
tree.pack()
win.mainloop()
```

Update Employee :



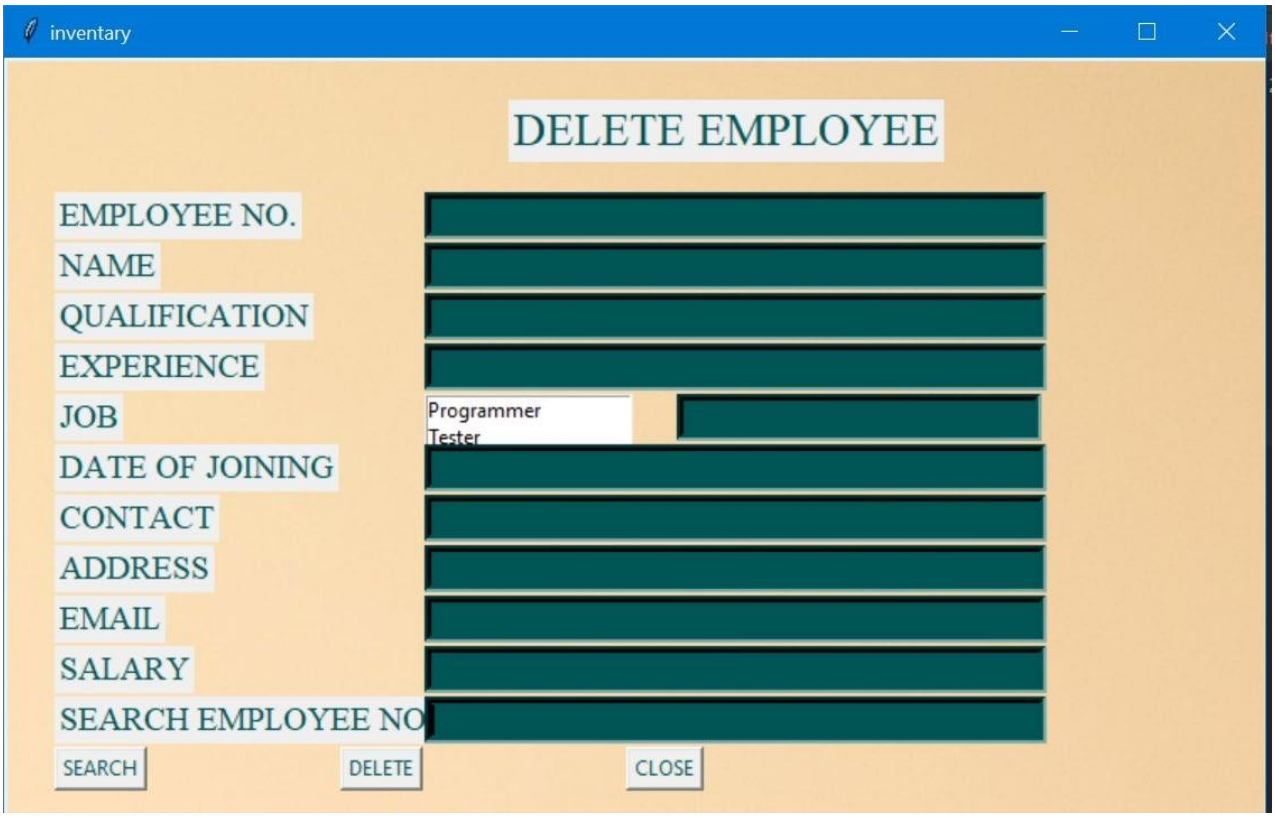


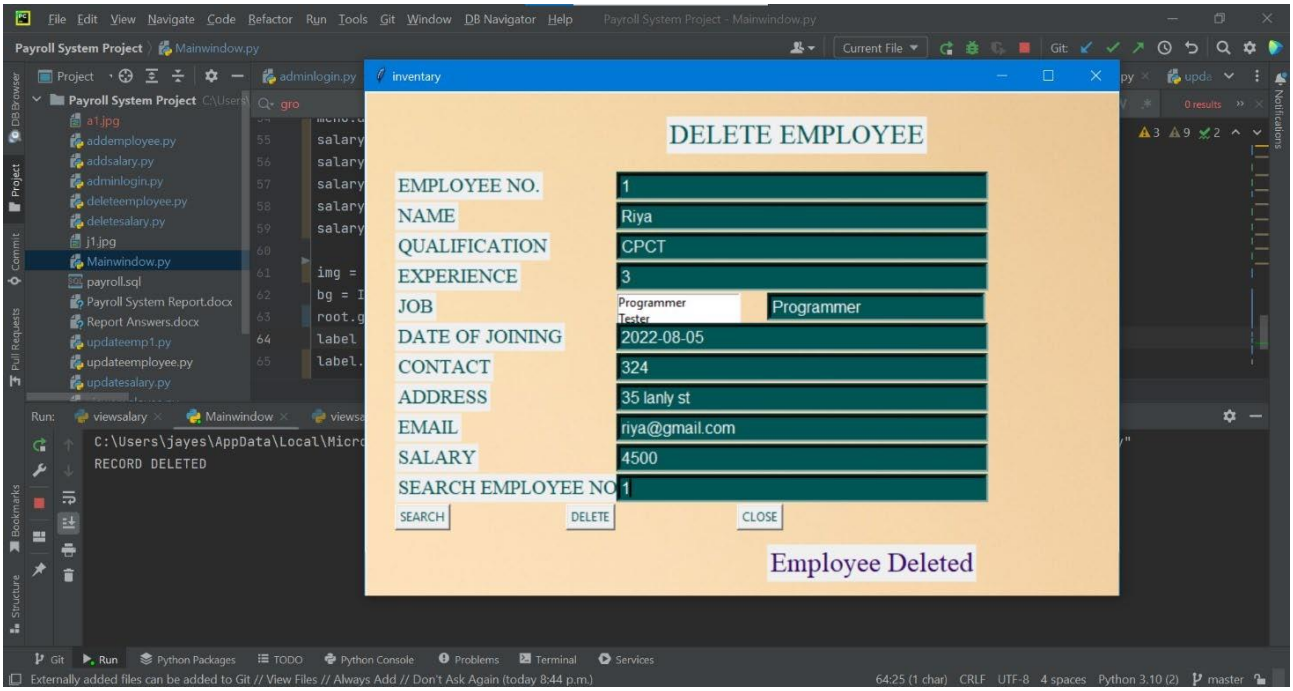

```
from tkinter import*
from PIL import ImageTk, Image
import pymysql
import pymysql.cursors


def search_record():
    conn=pymysql.connect(host='localhost',user='root',db='payroll')
    a=conn.cursor()
    args=int(e12.get())
    sql="SELECT * FROM salary where employeeno=%d"%(args)
    try:
       numrows=a.execute(sql)
       results=a.fetchall()
       for row in results:
            empno=row[0]
            name=row[1]
            month=row[2]
            ndays=row[3]
            nleaves=row[4]
            otime=row[5]
            salary=row[6]
            spday=row[7]
            deduct=row[8]
            etime=row[9]
            gpay=row[10]
            e1.insert(0,empno)
            e2.insert(1,name)
            e101.insert(2,month)
            e4.insert(3,ndays)
            e5.insert(4,nleaves)
            e6.insert(5,otime)
```

```
            e7.insert(6,salary)
            e8.insert(7,spday)
            e9.insert(8,deduct)
            e10.insert(9,etime)
            e11.insert(10,gpay)

    except:
        print("ERROR")
        a.close()
        conn.close()
def update_record():
    conn=pymysql.connect(host='localhost',user='root',db='payroll')
    a=conn.cursor()
    args1=e2.get()
    args2=e101.get()
    args3=e4.get()
    args4=e5.get()
    args5=e6.get()
    args6=e7.get()
    args7=e8.get()
    args8=e9.get()
    args9=e10.get()
    args10=e11.get()
    args11=int(e12.get())

    L1.config(text="Salary Updated",fg="#00827f",font=("times",20))
    sqlupd="UPDATE salary SET
name='%s',month='%s',noofdays='%s',noofleaves='%s',overtimeinhours='%s',salary=
'%s',salaryperday='%s',deduction='%s',extratime='%s',Netpay='%s' WHERE
employeeno=%d"%(args1,args2,args3,args4,args5,args6,args7,args8,args9,args10,ar
gs11)
    a.execute(sqlupd)
    print("Record updated")
    conn.commit()
    a.close()
    conn.close()


master=Tk()
master.title("payroll")
img =Image.open('a1.jpg')
bg = ImageTk.PhotoImage(img)
master.geometry("750x450")
label3 = Label(master, image=bg)
label3.place(x =0,y =0)

label=Label(master,text="UPDATE SALARY",fg="#00827f",font=("times",20))
label.place(x=300,y=25)
label1=Label(master,text="Employee No.",fg="#00827f",font=("times",15))
label1.place(x=30,y=80)
label2=Label(master,text="Name",fg="#00827f",font=("times",15))
label2.place(x=30,y=110)
label3=Label(master,text="Month",fg="#00827f",font=("times",15))
label3.place(x=30,y=140)
label4=Label(master,text="No. Of Days",fg="#00827f",font=("times",15))
label4.place(x=30,y=170)
label5=Label(master,text="No. Of Leaves",fg="#00827f",font=("times",15))
label5.place(x=30,y=200)

label6=Label(master,text="Overtime(in hrs.)",fg="#00827f",font=("times",15))
label6.place(x=30,y=230)
label7=Label(master,text="Salary",fg="#00827f",font=("times",15))
```

```
label7.place(x=30,y=260)
label8=Label(master,text="Salary/day",fg="#00827f",font=("times",15))
label8.place(x=30,y=290)
label9=Label(master,text="Deduction",fg="#00827f",font=("times",15))
label9.place(x=30,y=320)

label10=Label(master,text="Extra time",fg="#00827f",font=("times",15))
label10.place(x=30,y=350)
label11=Label(master,text="Net Pay",fg="#00827f",font=("times",15))
label11.place(x=30,y=380)
label12=Label(master,text="Search Employee No.",fg="#00827f",font=("times",15))
label12.place(x=30,y=410)
e1=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")
e2=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")

e4=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")
e5=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")
e6=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")
e7=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")
e8=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")
e9=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")
e10=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectback
ground="cyan")
e11=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectback
ground="cyan")
e12=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectback
ground="cyan")

e1.place(x=200,y=80)
e2.place(x=200,y=110)
e101=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbac
kground="cyan")
e101.place(x=200,y=140)
e4.place(x=200,y=170)
e5.place(x=200,y=200)
e6.place(x=200,y=230)
e7.place(x=200,y=260)
e8.place(x=200,y=290)
e9.place(x=200,y=320)
e10.place(x=200,y=350)
e11.place(x=200,y=380)
e12.place(x=200,y=410)
Button(master,text="Search",fg="#00827f",command=search_record).place(x=30,y=44
0)
Button(master,text="Update",fg="#00827f",command=update_record).place(x=250,y=4
40)
Button(master,text="CLOSE",fg="#00827f",command=master.quit).place(x=450,y=440)


L1=Label(master,text="")
L1.place  (x=400,y=470)
mainloop()
```

Delete Employee :






```
from tkinter import*
from PIL import ImageTk, Image
import pymysql
import pymysql.cursors

def onselect(evt):
    global job
```

```
    w=evt.widget
    index=int(w.curselection()[0])
    value=w.get(index)
    standard=value
    e101.delete(0,END)
    e101.insert(END,standard)
    print('you selected item %d:""%s"'%(index,value))
def search_record():
    conn=pymysql.connect(host='localhost',user='root',db='payroll')
    a=conn.cursor()
    args=int(e11.get())
    sql="SELECT * FROM employee where employeeno=%d"%(args)
    try:
       numrows=a.execute(sql)
       results=a.fetchall()
       for row in results:
            empno=row[0]
            name=row[1]
            qlfn=row[2]
            exp=row[3]
            job=row[4]
            doj=row[5]
            contact=row[6]
            address=row[7]
            email=row[8]
            salary=row[9]
            e1.insert(0,empno)
            e2.insert(1,name)
            e3.insert(2,qlfn)
            e4.insert(3,exp)
            e101.insert(4,job)
            e6.insert(5,doj)
            e7.insert(6,contact)
            e8.insert(7,address)
            e9.insert(8,email)
            e10.insert(9,salary)
    except:
        print("ERROR")
    a.close()
    conn.close()


def delete_record():
    conn=pymysql.connect(host='localhost',user='root',db='payroll')
    a=conn.cursor()
    args=int(e11.get())
    delstmt="DELETE FROM employee where employeeno=%d"%(args)
    L1.config(text="Employee Deleted",fg="#330066",font=("times",20))
    a.execute(delstmt)
    print("RECORD DELETED")
    conn.commit()
    a.close()
    conn.close()
def PRODUCT():
    k1.insert(END, "Programmer")
    k1.insert(END, "Tester")
    k1.insert(END, "Database administrator")
    k1.insert(END, "Web Developer")
master=Tk()
master.title("inventary")
img =Image.open('a1.jpg')
bg = ImageTk.PhotoImage(img)
```

```
master.geometry("750x450")
label3 = Label(master, image=bg)
label3.place(x =0,y =0)

label=Label(master,text="DELETE EMPLOYEE",fg="#005555",font=("times",20))
label.place(x=300,y=25)
label1=Label(master,text="EMPLOYEE NO.",fg="#005555",font=("times",15))
label1.place(x=30,y=80)
label2=Label(master,text="NAME",fg="#005555",font=("times",15))
label2.place(x=30,y=110)
label3=Label(master,text="QUALIFICATION",fg="#005555",font=("times",15))
label3.place(x=30,y=140)
label4=Label(master,text="EXPERIENCE",fg="#005555",font=("times",15))
label4.place(x=30,y=170)
label5=Label(master,text="JOB",fg="#005555",font=("times",15))
label5.place(x=30,y=200)

label6=Label(master,text="DATE OF JOINING",fg="#005555",font=("times",15))
label6.place(x=30,y=230)
label7=Label(master,text="CONTACT",fg="#005555",font=("times",15))
label7.place(x=30,y=260)
label8=Label(master,text="ADDRESS",fg="#005555",font=("times",15))
label8.place(x=30,y=290)
label9=Label(master,text="EMAIL",fg="#005555",font=("times",15))
label9.place(x=30,y=320)

label10=Label(master,text="SALARY",fg="#005555",font=("times",15))
label10.place(x=30,y=350)
label11=Label(master,text="SEARCH EMPLOYEE NO",fg="#005555",font=("times",15))
label11.place(x=30,y=380)

e1=Entry(master,width=40,bd=4,fg="white",bg="#005555",font="italic",selectbackg
round="cyan")
e2=Entry(master,width=40,bd=4,fg="white",bg="#005555",font="italic",selectbackg
round="cyan")
e3=Entry(master,width=40,bd=4,fg="white",bg="#005555",font="italic",selectbackg
round="cyan")
e4=Entry(master,width=40,bd=4,fg="white",bg="#005555",font="italic",selectbackg
round="cyan")
k1=Listbox(master,height=2)
k1.place(x=250,y=200)
k1.bind("<Double-1>",onselect)
e6=Entry(master,width=40,bd=4,fg="white",bg="#005555",font="italic",selectbackg
round="cyan")
e7=Entry(master,width=40,bd=4,fg="white",bg="#005555",font="italic",selectbackg
round="cyan")
e8=Entry(master,width=40,bd=4,fg="white",bg="#005555",font="italic",selectbackg
round="cyan")
e9=Entry(master,width=40,bd=4,fg="white",bg="#005555",font="italic",selectbackg
round="cyan")
e10=Entry(master,width=40,bd=4,fg="white",bg="#005555",font="italic",selectback
ground="cyan")
e11=Entry(master,width=40,bd=4,fg="white",bg="#005555",font="italic",selectback
ground="cyan")
e1.place(x=250,y=80)
e2.place(x=250,y=110)
e3.place(x=250,y=140)
e4.place(x=250,y=170)
e101=Entry(master,width=23,bd=4,fg="white",bg="#005555",font="italic",selectbac
kground="cyan")
e101.place(x=400,y=200)
e6.place(x=250,y=230)
```

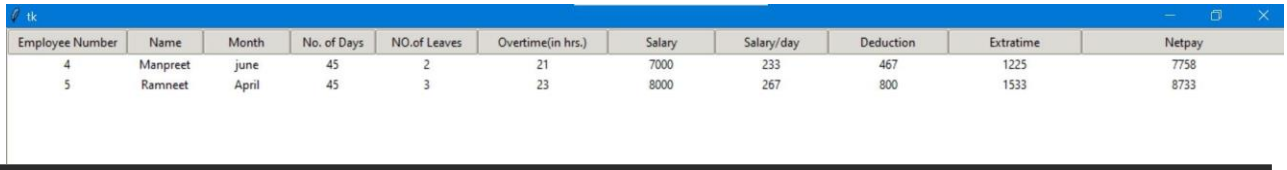
```
e7.place(x=250,y=260)
e8.place(x=250,y=290)
e9.place(x=250,y=320)
e10.place(x=250,y=350)
e11.place(x=250,y=380)
Button(master,text="SEARCH",fg="#005555",command=search_record).place(x=30,y=41
0)
Button(master,text="DELETE",fg="#005555",command=delete_record).place(x=200,y=4
10)
Button(master,text="CLOSE",fg="#005555",command=master.quit).place(x=370,y=410)


L1=Label(master,text="")
L1.place  (x=400,y=450)
PRODUCT()
mainloop()
```

Salary menu:


```
from tkinter import *
from tkinter.filedialog import askopenfile
from tkinter.messagebox import showerror
from PIL import ImageTk, Image
import sys, os


def AddEmployee():
    os.system('AddEmployee.py')
    print("Add Employee")


def DeleteEmployee():
    os.system('DeleteEmployee.py')
```

```
def UpdateEmp():
    os.system('UpdateEmp1.py')


def ViewEmployee():
    os.system('ViewEmployee.py')


def AddSalary():
    os.system('AddSalary.py')


def DeleteSalary():
    os.system('DeleteSalary.py')


def UpdateSalary():
    os.system('UpdateSalary.py')


def ViewSalary():
    os.system('ViewSalary.py')


root = Tk()
menu = Menu(root)
root.config(menu=menu)
root.title("Payroll Management System")

employeemenu = Menu(menu)
menu.add_cascade(label="Employee", menu=employeemenu)
employeemenu.add_command(label="Add Employee", command=AddEmployee)
employeemenu.add_command(label="Delete Employee", command=DeleteEmployee)
employeemenu.add_command(label="Update Employee", command=UpdateEmp)
employeemenu.add_command(label="View Employee", command=ViewEmployee)
employeemenu.add_command(label="Exit", command=root.quit)
salarymenu = Menu(menu)
menu.add_cascade(label="Salary", menu=salarymenu)
salarymenu.add_command(label="Add Salary", command=AddSalary)
salarymenu.add_command(label="Delete Salary", command=DeleteSalary)
salarymenu.add_command(label="Update Salary", command=UpdateSalary)
salarymenu.add_command(label="View Salary", command=ViewSalary)
salarymenu.add_command(label="Exit", command=root.quit)

img = Image.open('j1.jpg')
bg = ImageTk.PhotoImage(img)
root.geometry("1200x450")
label = Label(root, image=bg)
label.place(x=0, y=0)
label2 = Label(root, text="Payroll Management System", fg="Blue",
               font=("Arial Black", 40))
label2.pack(pady=10)
mainloop()
```

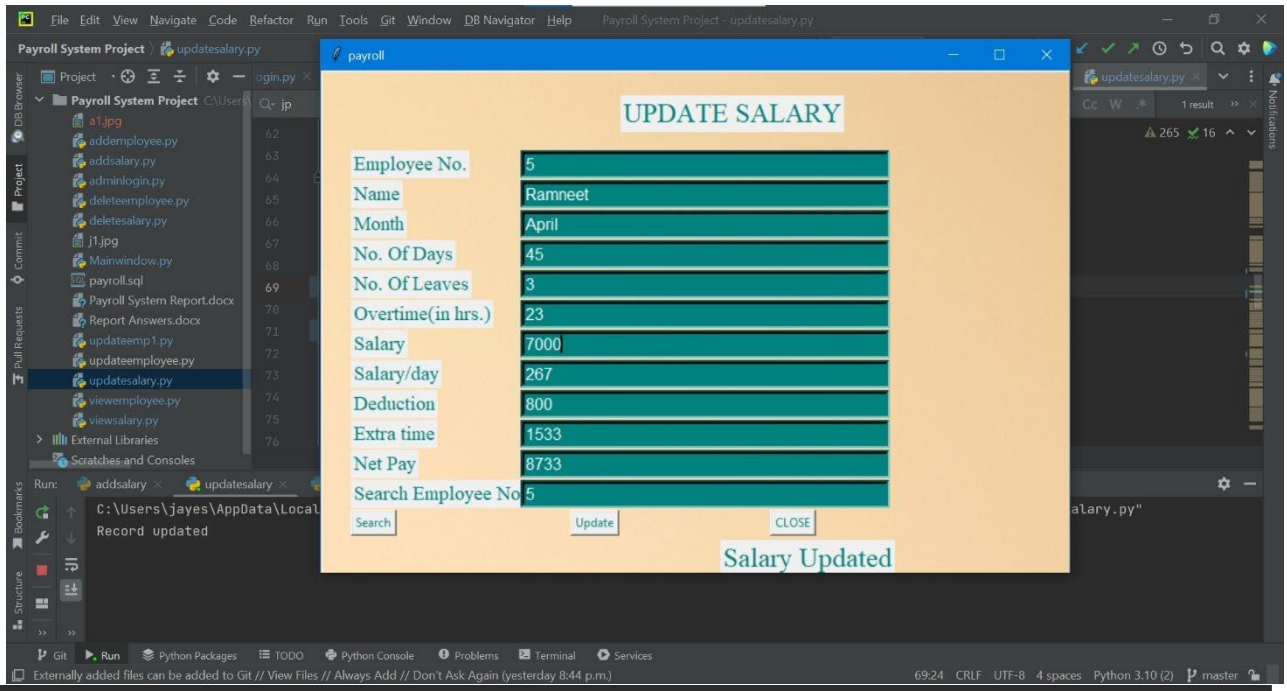
View Salary:




```
from tkinter import *
from tkinter import ttk
import pymysql
import pymysql.cursors

win = Tk()
win.geometry("750x450")
style = ttk.Style()
style.theme_use('clam')
tree = ttk.Treeview(win, column=("Employee Number", "Name", "Month", "No. of
Days", "No. of Leaves", "Overtime(in hrs.)", "Salary", "Salary/day",
"Deduction", "Extratime", "Netspay"), show='headings', height=30)
tree.column("# 1", anchor=CENTER)
tree.heading("# 1", text="Employee Number")
tree.column("# 2", anchor=CENTER)
tree.heading("# 2", text="Name")
tree.column("# 3", anchor=CENTER)
tree.heading("# 3", text="Month")
tree.column("# 4", anchor=CENTER)
```

```
tree.heading("#4", text="No. of Days")
tree.column("# 5", anchor=CENTER)
tree.heading("#5", text="NO.of Leaves")
tree.column("# 6", anchor=CENTER)
tree.heading("#6", text="Overtime(in hrs.)")
tree.column("# 7", anchor=CENTER)
tree.heading("#7", text="Salary")
tree.column("# 8", anchor=CENTER)
tree.heading("#8", text="Salary/day")
tree.column("# 9", anchor=CENTER)
tree.heading("#9", text="Deduction")
tree.column("# 10", anchor=CENTER)
tree.heading("#10", text="Extratime")
tree.column("# 11", anchor=CENTER)
tree.heading("#11", text="Netpay")
conn = pymysql.connect(host='localhost', user='root', db='payroll')
a = conn.cursor()
sql = "SELECT * FROM salary"
try:
    numrows = a.execute(sql)
    results = a.fetchall()
    for row in results:
        empno = row[0]
        name = row[1]
        month = row[2]
        ndays = row[3]
        nleaves = row[4]
        otime = row[5]
        salary = row[6]
        spday = row[7]
        deduct = row[8]
        etime = row[9]
        gpay = row[10]
        tree.insert('', 'end', text="1",
                    values=(empno, name, month, ndays, nleaves, otime, salary,
spday, deduct, etime, gpay))
except:
    print("ERROR")
a.close()
conn.close()
tree.pack()
win.mainloop()
```

Updat Salary:

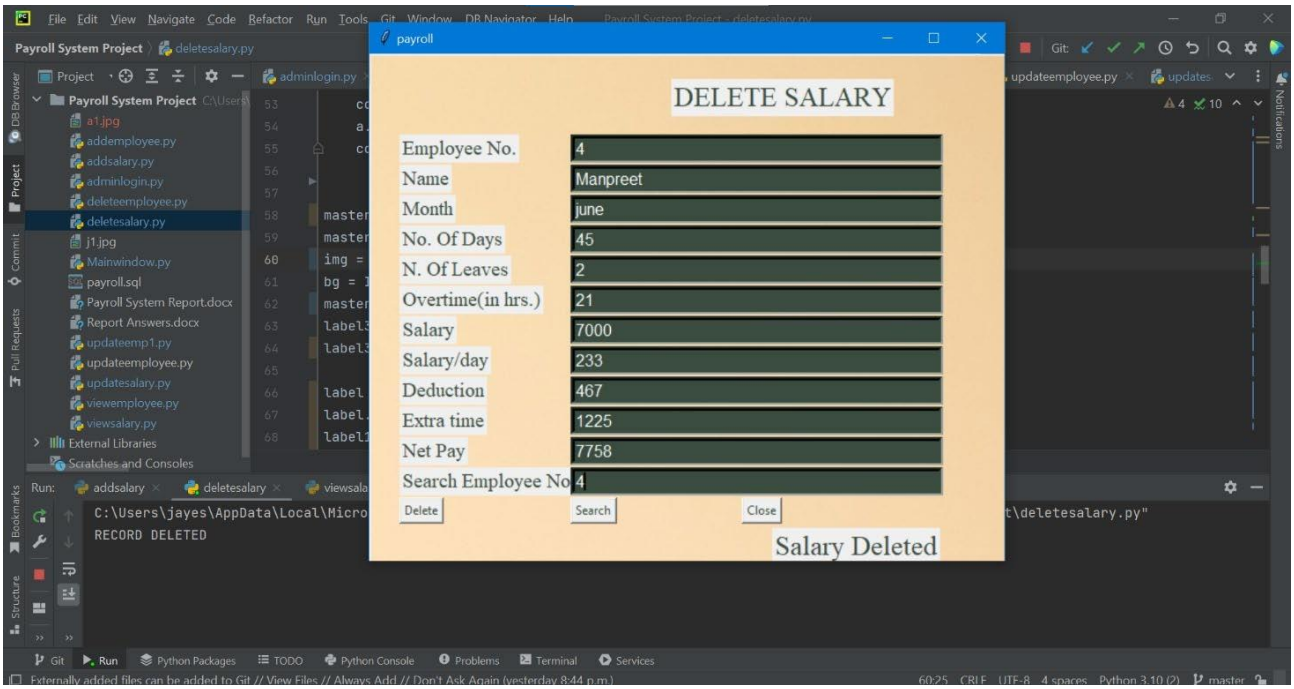



```
from tkinter import*
from PIL import ImageTk, Image
import pymysql
import pymysql.cursors


def search_record():
    conn=pymysql.connect(host='localhost',user='root',db='payroll')
    a=conn.cursor()
    args=int(e12.get())
    sql="SELECT * FROM salary where employeeno=%d"%(args)
    try:
       numrows=a.execute(sql)
       results=a.fetchall()
       for row in results:
            empno=row[0]
            name=row[1]
            month=row[2]
            ndays=row[3]
            nleaves=row[4]
            otime=row[5]
            salary=row[6]
            spday=row[7]
            deduct=row[8]
            etime=row[9]
            gpay=row[10]
            e1.insert(0,empno)
            e2.insert(1,name)
            e101.insert(2,month)
            e4.insert(3,ndays)
            e5.insert(4,nleaves)
            e6.insert(5,otime)
            e7.insert(6,salary)
            e8.insert(7,spday)
            e9.insert(8,deduct)
            e10.insert(9,etime)
            e11.insert(10,gpay)

    except:
```

```
        print("ERROR")
        a.close()
        conn.close()
def update_record():
    conn=pymysql.connect(host='localhost',user='root',db='payroll')
    a=conn.cursor()
    args1=e2.get()
    args2=e101.get()
    args3=e4.get()
    args4=e5.get()
    args5=e6.get()
    args6=e7.get()
    args7=e8.get()
    args8=e9.get()
    args9=e10.get()
    args10=e11.get()
    args11=int(e12.get())

    L1.config(text="Salary Updated",fg="#00827f",font=("times",20))
    sqlupd="UPDATE salary SET
name='%s',month='%s',noofdays='%s',noofleaves='%s',overtimeinhours='%s',salary=
'%s',salaryperday='%s',deduction='%s',extratime='%s',Netpay='%s' WHERE
employeeno=%d"%(args1,args2,args3,args4,args5,args6,args7,args8,args9,args10,ar
gs11)
    a.execute(sqlupd)
    print("Record updated")
    conn.commit()
    a.close()
    conn.close()


master=Tk()
master.title("payroll")
img =Image.open('a1.jpg')
bg = ImageTk.PhotoImage(img)
master.geometry("750x450")
label3 = Label(master, image=bg)
label3.place(x =0,y =0)

label=Label(master,text="UPDATE SALARY",fg="#00827f",font=("times",20))
label.place(x=300,y=25)
label1=Label(master,text="Employee No.",fg="#00827f",font=("times",15))
label1.place(x=30,y=80)
label2=Label(master,text="Name",fg="#00827f",font=("times",15))
label2.place(x=30,y=110)
label3=Label(master,text="Month",fg="#00827f",font=("times",15))
label3.place(x=30,y=140)
label4=Label(master,text="No. Of Days",fg="#00827f",font=("times",15))
label4.place(x=30,y=170)
label5=Label(master,text="No. Of Leaves",fg="#00827f",font=("times",15))
label5.place(x=30,y=200)

label6=Label(master,text="Overtime(in hrs.)",fg="#00827f",font=("times",15))
label6.place(x=30,y=230)
label7=Label(master,text="Salary",fg="#00827f",font=("times",15))
label7.place(x=30,y=260)
label8=Label(master,text="Salary/day",fg="#00827f",font=("times",15))
label8.place(x=30,y=290)
label9=Label(master,text="Deduction",fg="#00827f",font=("times",15))
label9.place(x=30,y=320)

label10=Label(master,text="Extra time",fg="#00827f",font=("times",15))
```

```
label10.place(x=30,y=350)
label11=Label(master,text="Net Pay",fg="#00827f",font=("times",15))
label11.place(x=30,y=380)
label12=Label(master,text="Search Employee No.",fg="#00827f",font=("times",15))
label12.place(x=30,y=410)
e1=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")
e2=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")

e4=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")
e5=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")
e6=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")
e7=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")
e8=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")
e9=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbackg
round="cyan")
e10=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectback
ground="cyan")
e11=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectback
ground="cyan")
e12=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectback
ground="cyan")

e1.place(x=200,y=80)
e2.place(x=200,y=110)
e101=Entry(master,width=40,bd=4,fg="white",bg="#00827f",font="italic",selectbac
kground="cyan")
e101.place(x=200,y=140)
e4.place(x=200,y=170)
e5.place(x=200,y=200)
e6.place(x=200,y=230)
e7.place(x=200,y=260)
e8.place(x=200,y=290)
e9.place(x=200,y=320)
e10.place(x=200,y=350)
e11.place(x=200,y=380)
e12.place(x=200,y=410)
Button(master,text="Search",fg="#00827f",command=search_record).place(x=30,y=44
0)
Button(master,text="Update",fg="#00827f",command=update_record).place(x=250,y=4
40)
Button(master,text="CLOSE",fg="#00827f",command=master.quit).place(x=450,y=440)


L1=Label(master,text="")
L1.place  (x=400,y=470)
mainloop()
```

Delete Salary:



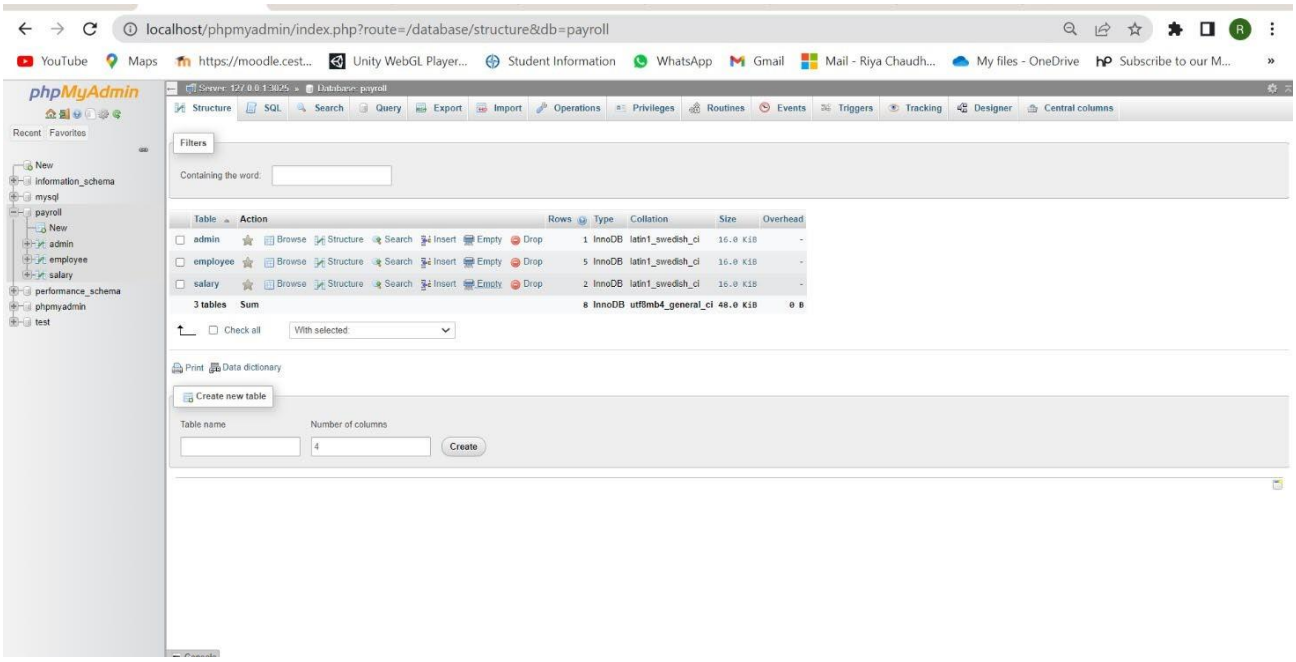

```
from tkinter import *
from PIL import ImageTk, Image
import pymysql
import pymysql.cursors


def search_record():
    conn = pymysql.connect(host='localhost', user='root', db='payroll')
    a = conn.cursor()
    args = int(e12.get())
    sql = "SELECT * FROM salary where employeeno=%d" % (args)
    try:
        numrows = a.execute(sql)
        results = a.fetchall()
        for row in results:
            empno = row[0]
            name = row[1]
            month = row[2]
            ndays = row[3]
            nleaves = row[4]
            otime = row[5]
            salary = row[6]
            spday = row[7]
            deduct = row[8]
            etime = row[9]
            gpay = row[10]
            e1.insert(0, empno)
            e2.insert(1, name)
            e101.insert(2, month)
            e4.insert(3, ndays)
            e5.insert(4, nleaves)
            e6.insert(5, otime)
            e7.insert(6, salary)
            e8.insert(7, spday)
            e9.insert(8, deduct)
            e10.insert(9, etime)
            e11.insert(10, gpay)
```

```
    except:
        print("ERROR")
        a.close()
        conn.close()


def delete_record():
    conn = pymysql.connect(host='localhost', user='root', db='payroll')
    a = conn.cursor()
    args = int(e12.get())
    delstmt = "DELETE FROM salary where employeeno=%d" % (args)
    L1.config(text="Salary Deleted", fg="#3a4c40", font=("times", 20))
    a.execute(delstmt)
    print("RECORD DELETED")
    conn.commit()
    a.close()
    conn.close()


master = Tk()
master.title("payroll")
img = Image.open('a1.jpg')
bg = ImageTk.PhotoImage(img)
master.geometry("750x450")
label3 = Label(master, image=bg)
label3.place(x=0, y=0)

label = Label(master, text="DELETE SALARY", fg="#3a4c40", font=("times", 20))
label.place(x=300, y=25)
label1 = Label(master, text="Employee No.", fg="#3a4c40", font=("times", 15))
label1.place(x=30, y=80)
label2 = Label(master, text="Name", fg="#3a4c40", font=("times", 15))
label2.place(x=30, y=110)
label3 = Label(master, text="Month", fg="#3a4c40", font=("times", 15))
label3.place(x=30, y=140)
label4 = Label(master, text="No. Of Days", fg="#3a4c40", font=("times", 15))
label4.place(x=30, y=170)
label5 = Label(master, text="N. Of Leaves", fg="#3a4c40", font=("times", 15))
label5.place(x=30, y=200)

label6 = Label(master, text="Overtime(in hrs.)", fg="#3a4c40", font=("times",
15))
label6.place(x=30, y=230)
label7 = Label(master, text="Salary", fg="#3a4c40", font=("times", 15))
label7.place(x=30, y=260)
label8 = Label(master, text="Salary/day", fg="#3a4c40", font=("times", 15))
label8.place(x=30, y=290)
label9 = Label(master, text="Deduction", fg="#3a4c40", font=("times", 15))
label9.place(x=30, y=320)

label10 = Label(master, text="Extra time", fg="#3a4c40", font=("times", 15))
label10.place(x=30, y=350)
label11 = Label(master, text="Net Pay", fg="#3a4c40", font=("times", 15))
label11.place(x=30, y=380)
label12 = Label(master, text="Search Employee No.", fg="#3a4c40",
font=("times", 15))
label12.place(x=30, y=410)
e1 = Entry(master, width=40, bd=4, fg="white", bg="#3a4c40", font="italic",
selectbackground="cyan")
e2 = Entry(master, width=40, bd=4, fg="white", bg="#3a4c40", font="italic",
selectbackground="cyan")
```

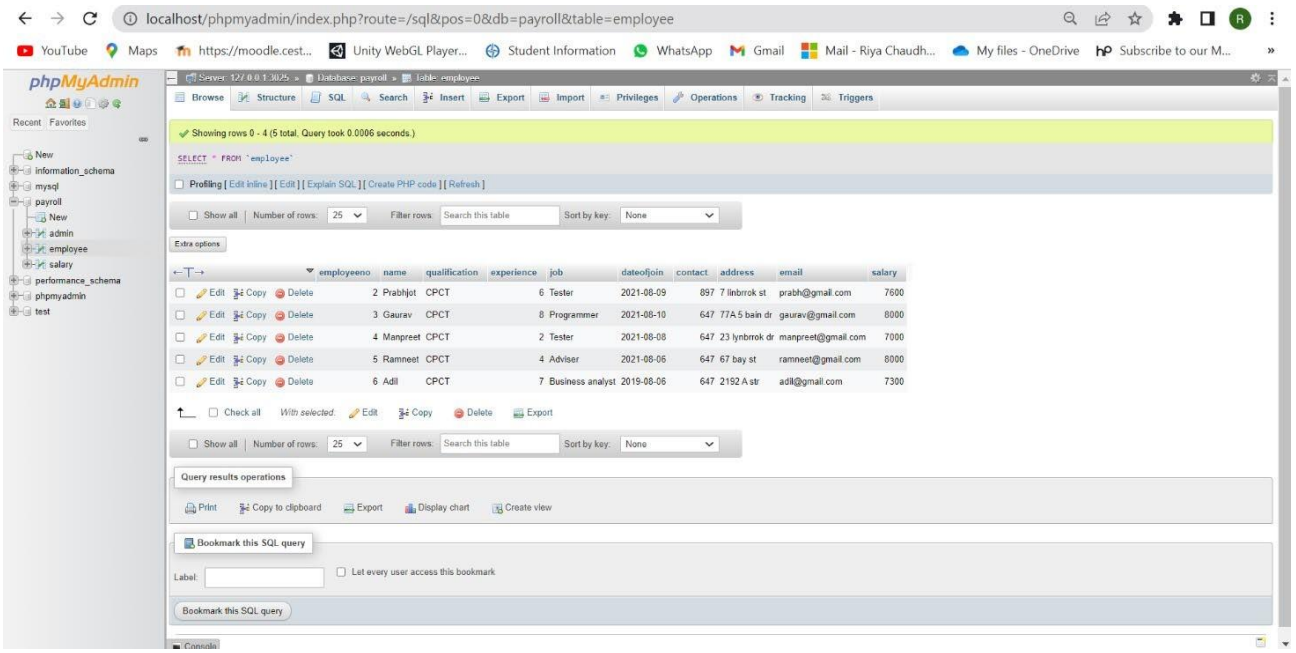
```
e4 = Entry(master, width=40, bd=4, fg="white", bg="#3a4c40", font="italic",
selectbackground="cyan")
e5 = Entry(master, width=40, bd=4, fg="white", bg="#3a4c40", font="italic",
selectbackground="cyan")
e6 = Entry(master, width=40, bd=4, fg="white", bg="#3a4c40", font="italic",
selectbackground="cyan")
e7 = Entry(master, width=40, bd=4, fg="white", bg="#3a4c40", font="italic",
selectbackground="cyan")
e8 = Entry(master, width=40, bd=4, fg="white", bg="#3a4c40", font="italic",
selectbackground="cyan")
e9 = Entry(master, width=40, bd=4, fg="white", bg="#3a4c40", font="italic",
selectbackground="cyan")
e10 = Entry(master, width=40, bd=4, fg="white", bg="#3a4c40", font="italic",
selectbackground="cyan")
e11 = Entry(master, width=40, bd=4, fg="white", bg="#3a4c40", font="italic",
selectbackground="cyan")
e12 = Entry(master, width=40, bd=4, fg="white", bg="#3a4c40", font="italic",
selectbackground="cyan")

e1.place(x=200, y=80)
e2.place(x=200, y=110)
e101 = Entry(master, width=40, bd=4, fg="white", bg="#3a4c40", font="italic",
selectbackground="cyan")
e101.place(x=200, y=140)
e4.place(x=200, y=170)
e5.place(x=200, y=200)
e6.place(x=200, y=230)
e7.place(x=200, y=260)
e8.place(x=200, y=290)
e9.place(x=200, y=320)
e10.place(x=200, y=350)
e11.place(x=200, y=380)
e12.place(x=200, y=410)
Button(master, text="Delete", fg="#3a4c40", command=delete_record).place(x=30,
y=440)
Button(master, text="Search", fg="#3a4c40", command=search_record).place(x=200,
y=440)
Button(master, text="Close", fg="#3a4c40", command=master.quit).place(x=370,
y=440)

L1 = Label(master, text="")
L1.place(x=400, y=470)

mainloop()
```

Database

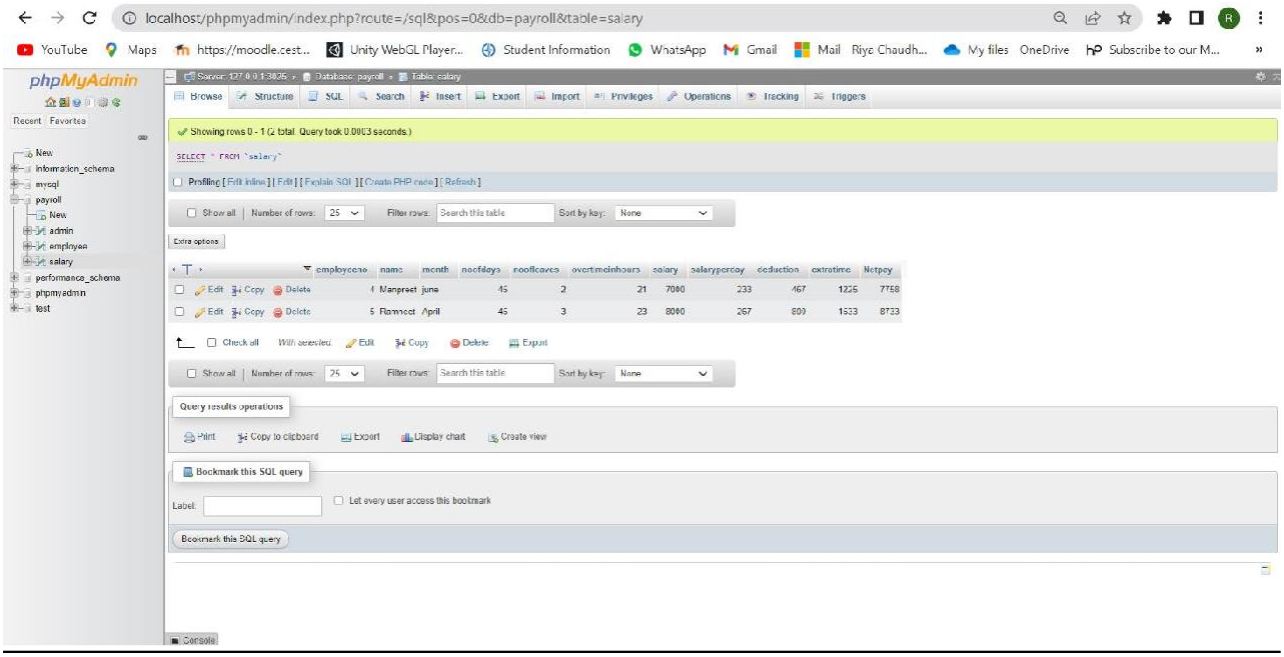
Admin:



Employee:



Salary:




```
-- phpMyAdmin SQL Dump
-- version 3.5.2.2
-- http://www.phpmyadmin.net
--
-- Host: 127.0.0.1
-- Generation Time: Nov 28, 2022 at 03:14 PM
-- Server version: 5.5.27
-- PHP Version: 5.4.7

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Database: `payroll`
--

---- --------------------------------------------------------------------------------


--
-- Table structure for table `admin`
--

CREATE TABLE IF NOT EXISTS `admin` (
  `uid` varchar(20) NOT NULL,
  `pass` varchar(20) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `admin`
--

INSERT INTO `admin` (`uid`, `pass`) VALUES
('admin', '123');

---- --------------------------------------------------------------------------------


--
-- Table structure for table `employee`
--

CREATE TABLE IF NOT EXISTS `employee` (
  `employeeno` int(20) NOT NULL,
  `name` varchar(20) NOT NULL,
  `qualification` varchar(50) NOT NULL,
  `experience` varchar(50) NOT NULL,
  `job` varchar(20) NOT NULL,
  `dateofjoin` varchar(50) NOT NULL,
  `contact` int(20) NOT NULL,
  `address` varchar(30) NOT NULL,
  `email` varchar(20) NOT NULL,
  `salary` int(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `employee`
--

INSERT INTO `employee` (`employeeno`, `name`, `qualification`, `experience`,
`job`, `dateofjoin`, `contact`, `address`, `email`, `salary`) VALUES
(1221, 'rahul', 'graduate', '1 yr', 'clerk', '20 october 2022', 2147483647,
'khanna', 'rahul@gmail.com', 12000),
(121, 'ragahv', 'graduate', '2 yrs', 'clerk', '21 nov 2022', 543435646, 'khanna',
'raghav@gmail.com', 15000),
(99, 'Kamal', 'MCA', '12', 'Accountant', '12-12-2021', 2147483647, 'khann', 'aa',
120000);

---- --------------------------------------------------------------------------------


--
-- Table structure for table `salary`
--

CREATE TABLE IF NOT EXISTS `salary` (
  `employeeno` int(10) NOT NULL,
```

```
  `name` varchar(30) NOT NULL,
  `month` varchar(30) NOT NULL,
  `noofdays` int(10) NOT NULL,
  `noofleaves` int(10) NOT NULL,
  `overtimeinhours` int(15) NOT NULL,
  `salary` int(10) NOT NULL,
  `salaryperday` int(10) NOT NULL,
  `deduction` int(10) NOT NULL,
  `extratime` int(10) NOT NULL,
  `grosspay` int(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dumping data for table `salary`
--

INSERT INTO `salary` (`employeeno`, `name`, `month`, `noofdays`, `noofleaves`,
`overtimeinhours`, `salary`, `salaryperday`, `deduction`, `extratime`, `grosspay`)
VALUES
(101, 'shivay', 'january', 28, 2, 20, 30000, 1000, 2000, 1000, 31000),
(22222, 'ghf', 'February', 22, 8, 12, 56000, 1867, 14933, 5600, 46667),
(99, 'Kamal', '', 30, 12, 12, 120000, 4000, 48000, 12000, 84000);

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

1 . Describe how this application can be deployed in a cloud environment.

In order to deploy our application, we will use Admin API.

Before deploying, we, the one who is the owner of the cloud should make the app engine application but ensure that the privileges should be fulfilled by the user account. Most importantly, permission should be given to the cloud build in order to do so.

Furthermore, downloading, installing and initializing the gloud CLI is the mandatory step. In order to configure the following command should be run

gcloud.config set proxy/type [PROXY_TYPE]

For deploying the app, following command should be run:

Gcloud app deploy [CONFIGURATION_FILES]

2. Describe how you would implement security features for your app in the cloud.

The ways to implement the security for the app in cloud environment are as follows:

1.) Managing the access to cloud applications and user behavior: Limiting the number of users as well as implementing multi-factor authentication. Moreover, using the strong policy obviously containing 14 characters such as lower case, upper case and special character. Furthermore, failed attempts of the login to the cloud should be limited.

2.) Cloud governance policies: multi-factor authentication should be must with the defined roles to know who and why is the person accessing the app. Other policies include monitoring the usage etc.

3.) Malware threat protection: The advanced components used by the attackers make this protection quite difficult. In order to protect, create a BYOD protection policy which helps to secure upload. Even, upgrade a cloud-specific protective layer to the cloud application so that the infrastructure is secured.

3.)If you need to make your application serverless how it can be done.

Answer 3: In order to build the serverless application, we can use AWS Lambda, Amazon API Gateway, Amazon DynamoDB, Amazon Cognito.

This can be done in the following order:

Firstly, hosting a static website to configure the AWS Amplify. Eventually, managing the users by creating amazon cognito user pool is mandatory step. Furthermore, building a serverless backend is utter most important to handle requests. Then, use amazon API gateway. Finally, terminate the resources.

CONCLUSION

The creation of this system required a wide range of research to review, update and learn new and old programming concepts of Python and modules.

To understand the functioning of a Payroll system, we use local and existing banks as a reference so that we can abstract reality in code lines.

It is always challenging to use new technologies, but even so, before starting this project, we looked for the most current tools on the market and used them in our project. This cleared many doubts, but we solved all our doubts by reading the official documentation.

REFERENCES

Create mysql database login page in Python using Tkinter. GeeksforGeeks. (2022, October 11). Retrieved December 4, 2022, from https://www.geeksforgeeks.org/create-mysql-database-login-page-in-python-using-tkinter/

Real Python. (2022, July 18). *Python GUI programming with Tkinter*. Real Python. Retrieved December 4, 2022, from https://realpython.com/python-gui-tkinter/

Google. (n.d.). *Deploying a python app | App Engine Standard Environment for Python 2 | google cloud*. Google. Retrieved December 5, 2022, from https://cloud.google.com/appengine/docs/legacy/standard/python/tools/uploadinganapp

10 best practices for application security in the cloud. Cypress Data Defense RSS. (n.d.). Retrieved December 5, 2022, from https://www.cypressdatadefense.com/blog/application-security-in-cloud/