

**MAHARISHI VIDYA MANDIR  
SENIOR SECONDARY SCHOOL  
POLACHERY**

**COMPUTER SCIENCE PROJECT**

**EMPLOYEE MANAGEMENT SYSTEM**

**NAME : S.MUGUNTHAN**  
**CLASS : XII-A1**  
**YEAR : 2020-2021**  
**ROLL NUMBER :**

<b>S.No.</b>	<b>TITLE</b>
1	BONAFIDE CERTIFICATE
2	ACKNOWLEDGEMENT
3	INTRODUCTION
4	PROBLEM DEFINITION
5	PROBLEM ANALYSIS
6	HARDWARE AND SOFTWARE REQUIREMENTS
7	FUTURE ENHANCEMENT
8	SOURCE CODE
9	OUTPUT
10	BIBLIOGRAPHY

## **BONAFIDE CERTIFICATE**

Certified to be the bonafide record of the Project work done by Master **S.MUGUNTHAN** of class **XII – A1** in **MAHARISHI VIDYA MANDIR SENIOR SECONDARY SCHOOL**, Polachery, Chennai 600 127 during the Year **2020-2021**.

Submitted for the **ALL INDIA SENIOR SCHOOL CERTIFICATE EXAMINATION** Conducted by **CBSE**, New Delhi, held in \_\_\_\_\_  
Month \_\_\_\_\_ Year at **MAHARISHI VIDYA MANDIR SENIOR SECONDARY SCHOOL**, Polachery, Chennai 600 127 by the candidate bearing the Registration Number \_\_\_\_\_.

DATE

INTERNAL EXAMINER

PRINCIPAL

EXTERNAL EXAMINER

SEAL

# **ACKNOWLEDGEMENT**

We wish to express our deep gratitude and sincere thanks to our Principal Ma'am **Mrs. M.U.Parimala Amudhan** and Associate Principal **Mrs. Sujatha Kasukela** and **our management**, for their encouragement and for all the facilities they provided for this project.

We extend our hearty thanks to **Mrs.B.Radhika**, Computer Science Teacher, who guided us to the successful completion of the project. We take this wonderful opportunity to express our deep sense of gratitude for her invaluable guidance, constant encouragement, constructive comments, sympathetic attitude and immense motivation which has sustained our efforts at all stages of this project work.

We would also like to thank our parents, for their blessings and the idea they gave us on creating this program, without them anything wouldn't have been possible.

We intently thank god for the blessings he showered on us so as to create this program.

# **INTRODUCTION**

Employee Management System Project is an GUI application that stores all the employee's information in a database. It is an application developed in Python GUI technology and database used is MYSQL.

Earlier systems were manual where there was no way of properly storing information. Employee records were stored manually which lead to errors. There was no proper way of tracking employee records. It was very difficult and required a lot of paperwork which makes the application time consuming and not secured.

This proposed system reduces the dependency on the manual system which could create errors. This system can easily help in tracking employee records. There is a search feature which allows getting records of a specific employee. There is an administrator which can add, edit, delete and save records in a database. It is an easy to use application and has a user-friendly interface. It is totally built at the administrative end which means that only the admin has access rights to add or change or modify any records.

The main aim of developing this application was to reduce the errors that occur in the manual system. One can search the details easily by just entering employee id. In earlier systems, there was not such a facility to do so. All the details are stored in an MYSQL database. It is easy to update any employee details. All the employee records are integrated and so this makes it user-friendly and easy to use application. It even stores who has logged in , logged in time and logout time in SQL. This is done using Tkinter and PIL module of Python.

# **PROBLEM**

# **DEFINITION**

The aim of our Python project is to make Employee Management System organized and computerized to add ,delete, update and view the details of employee and store login time and logout time along with login system such that only administrators can access it using Python GUI and MYSQL database.



# **PROBLEM**

# **ANALYSIS**

System analysis is the process of gathering and interpreting facts, diagnosing problem and using the information to recommended improvements to the system.

Our computerized EMPLOYEE MANAGEMENT SYSTEM is a python menu-driven GUI program which consists of different modules such as

➤ Login part:

- To allow only administrators to access the data.

➤ Main part:

- To create the new data  
To add the details of new employee into the system.
- To update the data  
To change the incorrect data given during creation of data using employee number.
- To view the data of single employee and all employee  
To view details of individual person by using his employee number or just employee number, name and designation of all employees.
- To delete the data  
To remove the details of individual person from all the records in the system using his employee number.




The proposed system is economically feasible because the project has been developed with the existing software and hardware.

# **HARDWARE AND**



# **SOFTWARE**

# **REQUIREMENTS**




## **HARDWARE REQUIREMENTS:-**

-  Mouse
-  Monitor
-  CPU

1. MEMORY SPACE - 1 TB
2. PROCESSOR - Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz  
1.80 GHz
3. RAM - 12.0 GB

-  UPS
-  Keyboard

## **SOFTWARE REQUIREMENTS:-**

-  OPERATING SYSTEM – WINDOWS 10
-  DEVELOPING LANGUAGE – PYTHON 3.0
-  DATABASE - MYSQL

# **FUTURE**

# **ENHANCEMENT**

The project has a very vast scope in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion.

The following are the future scope for the project:-

- ✚ Individual employee login to view their details alone.
- ✚ Entering details along with photo of employee.
- ✚ Operating GUI application using voice command.
- ✚ Employee attendance can be included.
- ✚ To add few more personal details during creation.

# **SOURCE CODE**

```

from tkinter import *
from tkinter import messagebox
from PIL import Image,ImageTk
import mysql.connector as m

def store():
    c=m.connect(host='localhost',user='root',password='2003',database='em')
    o=c.cursor()
    o.execute('select * from details')
    ed=o.fetchall()
    global l
    l={}
    for i in range(len(ed)):
        l[ed[i][0]]=ed[i][1]
    c.close()

def main1():
    store()
    main=Toplevel()
    main.title('Employee Management')
    main.maxsize(300,330)
    main.minsize(300,330)
    main.protocol ('WM_DELETE_WINDOW', (lambda: 'pass') ())
    login.protocol ('WM_DELETE_WINDOW', (lambda: 'pass') ())
    i=Image.open("C:\\Users\\Mugunthan\\n.jpg")
    i1=i.resize((300,330))
    i2=ImageTk.PhotoImage(i1)
    bi=Label(main,image=i2)
    bi.pack()
    I=Image.open("C:\\Users\\Mugunthan\\logo.jpg")
    I1=I.resize((100,100))
    I2=ImageTk.PhotoImage(I1)
    logo=Label(main,image=I2)
    logo.place(x=100,y=10)

```



```
head=Label(main,text='DAILY DAY NEWSPAPERS',font = "Helvetica 15
bold underline italic",fg='violet red')
head.place(x=20,y=120)
```

```
def ph_num(c,wh):    #validating phone number.
    a=c.isdigit()
    b=len(wh)<=10
    return a and b
```

```
ph_vad=main.register(ph_num)
```

```
def salary(c):    #validating salary
    return c.isdigit()
```

```
sal_vad=main.register(salary)
```

```
def create():
```

```
    crwin=Toplevel()
```

```
    crwin.title('CREATE DETAILS')
```

```
    crwin.maxsize(800,330)
```

```
    crwin.minsize(800,330)
```

```
    crwin.configure(background='pale turquoise')
```

```
    det=Label(crwin,text='Enter the details.',fg='dark green',bg='palevioletred1')
```

```
    det.place(x=350,y=10)
```

```
    name=Label(crwin,text='Name',fg='red')    #NAME
```

```
    name.place(x=20,y=50)
```

```
    def name_check(i):
```

```
        for j in namee.get():
```

```
            if j.isalpha() or j==' ' or j=='.':
```

```
                pass
```

```
            else:
```

```
                messagebox.showinfo('INFORMATION','Only alphabets,space and
dot is allowed.')
```

```
                namee.delete(0,'end')
```

```
                break
```

```
    namee=Entry(crwin,width=35)
```

```
    namee.bind("<FocusOut>",name_check)
```

```
namee.place(x=150,y=50)
```

```
dob=Label(crwin,text='Date of Birth',fg='red') #DATE OF BIRTH
```

```
dob.place(x=20,y=80)
```

```
day=Spinbox(crwin,from_=1,to=31,width=5)
```

```
day.place(x=150,y=80)
```

```
month=Spinbox(crwin,from_=1,to=12,width=5)
```

```
month.place(x=200,y=80)
```

```
year=Spinbox(crwin,from_=1950,to=2020,width=5)
```

```
year.place(x=250,y=80)
```

```
gen=Label(crwin,text='Gender',fg='red') #GENDER
```

```
gen.place(x=20,y=110)
```

```
var = IntVar()
```

```
M= Radiobutton(crwin,text="Male",variable=var,value=1,fg='chocolate3')
```

```
M.place(x=150,y=110)
```

```
F= Radiobutton(crwin,text="Female",variable=var,value=2,fg='chocolate3')
```

```
F.place(x=150,y=140)
```

```
fn=Label(crwin,text='Father name',fg='red') #FATHER NAME
```

```
fn.place(x=20,y=170)
```

```
def f_name_check(i):
```

```
    for j in fne.get():
```

```
        if j.isalpha() or j==' ' or j=='.':
```

```
            pass
```

```
        else:
```

```
            messagebox.showinfo('INFORMATION','Only alphabets,space and  
dot is allowed.')
```

```
            fne.delete(0,'end')
```

```
            break
```

```
fne=Entry(crwin,width=35)
```

```
fne.bind("<FocusOut>",f_name_check)
```

```
fne.place(x=150,y=170)
```

```
mn=Label(crwin,text='Mother name',fg='red') #MOTHER NAME
```

```

mn.place(x=20,y=200)
def m_name_check(i):
    for j in mne.get():
        if j.isalpha() or j==' ' or j=='.':
            pass
        else:
            messagebox.showinfo('INFORMATION','Only alphabets,space and
dot is allowed.')
            mne.delete(0,'end')
            break
mne=Entry(crwin,width=35)
mne.bind("<FocusOut>",m_name_check)
mne.place(x=150,y=200)

pn=Label(crwin,text='Phone number',fg='red') #PHONE NUMBER
pn.place(x=410,y=50)
def do_p(i):    #phone number check
    if pne.get()!=":
        if len(pne.get())!=10:
            pne.delete(0,'end')
            messagebox.showinfo('INFORMATION','You entered less than 10
digits for phone number.')

pne=Entry(crwin,width=20,validate="all",validatecommand=(ph_vad,'%S','%P'))
pne.bind("<FocusOut>", do_p)
pne.place(x=550,y=50)

em=Label(crwin,text='Email ID',fg='red') #E-MAIL ID
em.place(x=410,y=80)
eme=Entry(crwin,width=35)
eme.place(x=550,y=80)

ad=Label(crwin,text='Address',fg='red') #ADDRESS
ad.place(x=410,y=110)
ade=Entry(crwin,width=35)

```

```

ade.place(x=550,y=110)

doj=Label(crw,text='Date of Joining',fg='red') #DATE OF JOINING
doj.place(x=410,y=140)
day1=Spinbox(crw,from_=1,to=31,width=5)
day1.place(x=550,y=140)
month1=Spinbox(crw,from_=1,to=12,width=5)
month1.place(x=600,y=140)
year1=Spinbox(crw,from_=1950,to=2020,width=5)
year1.place(x=650,y=140)

de=Label(crw,text='Designation code',fg='red') #DESIGNATION
de.place(x=410,y=170)
dee=Spinbox(crw,from_=1,to=12,width=7)
dee.place(x=550,y=170)
def cr():
    crw=Toplevel()
    crw.title(' ')
    crw.maxsize(300,300)
    crw.minsize(300,300)
    h=Label(crw,text='Code with their designation.',fg='sienna4')
    h.place(x=30,y=10)
    desn=['Publisher','Editor in chief','Managing Director','Assistant Managing
Director',
        'audio/visual production specalist','News Editor','Sub Editor',
        'Photo Editor','Reporter','Video Grapher','Lawyer','Worker in printing
section']
    y1=30
    for i in range(1,13):
        st=str(i)+' - '+desn[i-1]
        pri=Label(crw,text=st,fg='dark violet')
        pri.place(x=20,y=y1)
        y1+=20
    crw.mainloop()

```

```

dcr=Button(crwin,text='Code
reference',command=cr,bg='khaki1',fg='maroon')
dcr.place(x=630,y=165)

sa=Label(crwin,text='Salary',fg='red') #SALARY
sa.place(x=410,y=200)
def do_s(i):    #salary check
    if sae.get()!=":
        if int(sae.get())<10000:
            sae.delete(0,'end')
            messagebox.showinfo('INFORMATION','You entered less than
minimum salary(Rs.10000).')
sae=Entry(crwin,width=25,validate="key", validatecommand=(sal_vad,'%S'))
sae.bind("<FocusOut>", do_s)
sae.place(x=550,y=200)
def sc():
    store()
    global empno
    empno=""
    dv={'1':'PB','2':'EC','3':'MD','4':'AM','5':'PS','6':'NE',
        '7':'SE','8':'PE','9':'RE','10':'VG','11':'LW','12':'PW'}
    global na
    na=namee.get()
    db=year.get()+'-'+month.get()+'-'+day.get()
    g=var.get()
    if g==1:
        g1='Male'
    elif g==2:
        g1='Female'
    else:
        g1=""
    fa=fne.get()
    mo=mne.get()
    ema=eme.get()
    add=ade.get()

```

```

dj=year1.get()+'-'+month1.get()+'-'+day1.get()
dc=dee.get()
empno+=dv[dc]
emp=empno
for i in range(1,10000):
    empno+=str(i)
    if empno not in l:
        break
    else:
        empno=emp
c=m.connect(host='localhost',user='root',password='2003',database='em')
o=c.cursor()
q="insert into details values(%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"
o.execute('select * from details')
ed=o.fetchall()
cond= g1!=" and namee.get()!=" and fne.get()!=" and mne.get()!=" and
pne.get()!=" and eme.get()!=" and ade.get()!=" and sae.get()!="
def check(q,cond):
    if cond:
        decision= messagebox.askquestion ('Yes or No','Are you sure you
want to add the data?')
        if decision=='yes':
            d=(empno,na,db,g1,fa,mo,pne.get(),ema,add,dj,dc,sae.get())
            o.execute(q,d)
            messagebox.showinfo('INFORMATION','Data added.\nThe
employee Number of {n} is {e}'.format(n=na,e=empno))
            l=[namee,fne,mne,pne,eme,ade,sae]
            for j in l:
                j.delete(0,'end')
        else:
            messagebox.showinfo('INFORMATION','Some of details not filled.')
    check(q,cond)
c.commit()
c.close()

```

```
sub=Button(crwin,text='CREATE',command=sc,fg='dark
green',bg='khaki',width=7)
sub.place(x=350,y=250)
def ebc():
    crwin.destroy()
    eb=Button(crwin,text='EXIT',command=ebc,fg='dark
green',bg='khaki',width=7)
    eb.place(x=350,y=290)
    crwin.mainloop()
```

```
createbutton=Button(main,text='CREATE
DETAILS',command=create,bg='olivedrab1',fg='mediumorchid4',width=15,height
=1)
createbutton.place(x=30,y=170)
```

```
def update():
    upwin=Toplevel()
    upwin.title('UPDATE DETAILS')
    upwin.maxsize(300,330)
    upwin.minsize(300,330)
    upwin.configure(background='aquamarine2')
    text=Label(upwin,text='Employee number:',fg='slateblue1')
    text.place(x=20,y=20)
    ene=Entry(upwin,width=20)
    ene.place(x=150,y=20)
    text1=Label(upwin,text='Select columns.',fg='slateblue1')
    text1.place(x=20,y=50)
    li = Listbox(upwin, selectmode = "multiple",height=11)
    li.place(x=150,y=50)
    lic=['Name','Date of birth','Gender' ,'Father Name','Mother Name','Phone
Number',
        'Email Id','Address','Date of joining','Designation','Salary']
    for i in range(len(lic)):
```

```

        li.insert(i+1,lic[i])
store()
def uc():
    inp=ene.get()
    global se
    se=li.curselection()
    if inp=="":
        messagebox.showinfo('INFORMATION','Employee number is empty.')
    elif inp not in l:
        messagebox.showinfo('INFORMATION','Data with this employee
number({e}) does not exist.'.format(e=inp))
    elif se==():
        messagebox.showinfo('INFORMATION','No columns selected.')
    elif len(se)>5:
        messagebox.showinfo('INFORMATION','Can update only 5 columns at
a time.')
    else:
        ene.delete(0,'end')
        uw=Toplevel()
        uw.title('UPDATE')
        uw.maxsize(400,250)
        uw.minsize(400,250)
        uw.configure(background='pale green')
        y1=20

        if 0 in se:
            uname=Label(uw,text='Name',fg='red')
            uname.place(x=20,y=y1)
            def uname_check(i):
                for j in unamee.get():
                    if j.isalpha() or j==' ' or j=='.':
                        pass
                    else:
                        messagebox.showinfo('INFORMATION','Only
alphabets,space and dot is allowed.')

```



```

        unamee.delete(0,'end')
        break
    unamee=Entry(uw,width=35)
    unamee.bind("<FocusOut>",uname_check)
    unamee.place(x=150,y=y1)
    y1+=30

if 1 in se:
    udob=Label(uw,text='Date of Birth',fg='red')
    udob.place(x=20,y=y1)
    uday=Spinbox(uw,from_=1,to=31,width=5)
    uday.place(x=150,y=y1)
    umonth=Spinbox(uw,from_=1,to=12,width=5)
    umonth.place(x=200,y=y1)
    uyear=Spinbox(uw,from_=1950,to=2020,width=5)
    uyear.place(x=250,y=y1)
    y1+=30

if 2 in se:
    ugen=Label(uw,text='Gender',fg='red')
    ugen.place(x=20,y=y1)
    var = IntVar()
    uM=
Radiobutton(uw,text="Male",variable=var,value=1,fg='chocolate3')
    uM.place(x=150,y=y1)
    y1+=30
    uF=
Radiobutton(uw,text="Female",variable=var,value=2,fg='chocolate3')
    uF.place(x=150,y=y1)
    y1+=30

if 3 in se:
    ufn=Label(uw,text='Father name',fg='red')
    ufn.place(x=20,y=y1)
    def f_upd_name_check(i):

```

```

        for j in ufne.get():
            if j.isalpha() or j==' ' or j=='.':
                pass
            else:
                messagebox.showinfo('INFORMATION','Only
alphabets,space and dot is allowed.')
                ufne.delete(0,'end')
                break
        ufne=Entry(uw,width=35)
        ufne.bind("<FocusOut>",f_upd_name_check)
        ufne.place(x=150,y=y1)
        y1+=30

if 4 in se:
    umn=Label(uw,text='Mother name',fg='red')
    umn.place(x=20,y=y1)
    def m_upd_name_check(i):
        for j in umne.get():
            if j.isalpha() or j==' ' or j=='.':
                pass
            else:
                messagebox.showinfo('INFORMATION','Only
alphabets,space and dot is allowed.')
                umne.delete(0,'end')
                break
        umne=Entry(uw,width=35)
        umne.bind("<FocusOut>",m_upd_name_check)
        umne.place(x=150,y=y1)
        y1+=30

if 5 in se:
    upn=Label(uw,text='Phone number',fg='red')
    upn.place(x=20,y=y1)
    def do_p(i):      #phone number check
        if upne.get()!=":

```

```
        if len(upne.get())!=10:
            upne.delete(0,'end')
            messagebox.showinfo(", 'You entered less than 10 digits for  
phone number.')
```

```
upne=Entry(uw,width=20,validate="all",validatecommand=(ph_vad,'%S','%P'))
upne.bind("<FocusOut>", do_p)
upne.place(x=150,y=y1)
y1+=30
```

```
if 6 in se:
    uem=Label(uw,text='Email ID',fg='red')
    uem.place(x=20,y=y1)
    ueme=Entry(uw,width=35)
    ueme.place(x=150,y=y1)
    y1+=30
```

```
if 7 in se:
    uad=Label(uw,text='Address',fg='red')
    uad.place(x=20,y=y1)
    uade=Entry(uw,width=35)
    uade.place(x=150,y=y1)
    y1+=30
```

```
if 8 in se:
    udoj=Label(uw,text='Date of Joining',fg='red')
    udoj.place(x=20,y=y1)
    uday1=Spinbox(uw,from_=1,to=31,width=5)
    uday1.place(x=150,y=y1)
    umonth1=Spinbox(uw,from_=1,to=12,width=5)
    umonth1.place(x=200,y=y1)
    uyear1=Spinbox(uw,from_=1950,to=2020,width=5)
    uyear1.place(x=250,y=y1)
    y1+=30
```

```
if 9 in se:
```

```
    ude=Label(uw,text='Designation code',fg='red')
```

```
    ude.place(x=20,y=y1)
```

```
    udee=Spinbox(uw,from_=1,to=12,width=7)
```

```
    udee.place(x=150,y=y1)
```

```
    y1+=30
```

```
if 10 in se:
```

```
    usa=Label(uw,text='Salary',fg='red')
```

```
    usa.place(x=20,y=y1)
```

```
    def do_s(i):      #salary check
```

```
        if usae.get()!=":
```

```
            if int(usae.get())<10000:
```

```
                usae.delete(0,'end')
```

```
                messagebox.showinfo(", 'You entered less than minimum
```

```
salary(Rs.10000).')
```

```
                usae=Entry(uw,width=25,validate="key",
```

```
validatecommand=(sal_vad,'%S'))
```

```
                usae.bind("<FocusOut>", do_s)
```

```
                usae.place(x=150,y=y1)
```

```
                y1+=30
```

```
def up_c():
```

```
    upda=()
```

```
    error=[]
```

```
    if 0 in se:
```

```
        una=unamee.get()
```

```
        if una!=":
```

```
            upda+=(una,)
```

```
        else:
```

```
            error+=['Name']
```

```
    if 1 in se:
```

```
        udb=uyear.get()+'-'+umonth.get()+'-'+uday.get()
```

```
        upda+=(udb,)
```

```
    if 2 in se:
```

```
ug=var.get()
if ug==1:
    ug1='Male'
    upda+=(ug1,)
elif ug==2:
    ug1='Female'
    upda+=(ug1,)
else:
    error+=['Gender']
if 3 in se:
    ufa=ufne.get()
    if ufa!="":
        upda+=(ufa,)
    else:
        error+=['Father Name']
if 4 in se:
    umo=umne.get()
    if umo!="":
        upda+=(umo,)
    else:
        error+=['Mother Name']
if 5 in se:
    upn=upne.get()
    if upn!="":
        upda+=(upn,)
    else:
        error+=['Phone Number']
if 6 in se:
    uema=ueme.get()
    if uema!="":
        upda+=(uema,)
    else:
        error+=['Email ID']
if 7 in se:
    uadd=uade.get()
```

```

        if uadd!="":
            upda+=(uadd,)
        else:
            error+=['Address']
    if 8 in se:
        udj=uyear1.get()+'-'+umonth1.get()+'-'+uday1.get()
        upda+=(udj,)
    if 9 in se:
        udc=udee.get()
        upda+=(udc,)
    if 10 in se:
        us=usae.get()
        if us!="":
            upda+=(us,)
        else:
            error+=['Salary']

```

```

c=m.connect(host='localhost',user='root',password='2003',database='em')
o=c.cursor()
q='update details set '
ori={0:'name',1:'dob',2:'gender',3:'father',4:'mother',5:'phone',
      6:'email',7:'Address',8:'doj',9:'desigco',10:'salary'}
for i in range(len(se)):
    q+=ori[se[i]]+'=%s'
    if i<len(se)-1:
        q+=", "
    if i==len(se)-1:
        q+=' where empno=%s'
upda+=(inp.upper(),)
def ucheck(q,upda):
    decision= messagebox.askquestion ('Yes or No','Is all data to be
updated is correct?')
    if decision=='yes':
        uw.destroy()
        o.execute(q,upda)

```

```

        c.commit()
        messagebox.showinfo('INFORMATION','Data updated')
    if len(se)==len(upda)-1:
        ucheck(q,upda)
    else:
        s='Check the following.\n'
        for i in range(len(error)):
            s+=str(i+1)+' '+error[i]+' \n'
        messagebox.showinfo('INFORMATION',s)
    c.close()
    ub=Button(uw,text='UPDATE',fg='black',command=up_c)
    ub.place(x=150,y=y1)
    uw.mainloop()

```

```

probu=Button(upwin,text='Proceed',command=uc,fg='black',bg='coral1',width=10,
height=2)

```

```

    probu.place(x=110,y=250)

```

```

    def ebc():

```

```

        upwin.destroy()

```

```

    ebu=Button(upwin,text='Exit',command=ebc,fg='black',bg='coral1',width=10)

```

```

    ebu.place(x=110,y=300)

```

```

    upwin.mainloop()

```

```

    updatebutton=Button(main,text='UPDATE
DETAILS',command=update,bg='olivedrab1',fg='mediumorchid4',width=15,height
=1)

```

```

    updatebutton.place(x=160,y=200)

```

```

def delete():

```

```

    dewin=Toplevel()

```

```

    dewin.title('DELETE DETAILS')

```

```

    dewin.maxsize(300,210)

```

```

    dewin.minsize(300,210)

```

```

    dewin.configure(background='khaki1')

```

```

store()
text=Label(dewin,text='Employee number',bg='khaki1',fg='red',font =
"Helvetica 13 bold italic")
text.place(x=90,y=30)
empe=Entry(dewin,width=40)
empe.place(x=30,y=70)
def dc():
    inp=empe.get()
    if inp=="":
        messagebox.showinfo('INFORMATION','Entry box is empty.')
    elif inp not in l:
        messagebox.showinfo('INFORMATION','Employee number( {a} ) does
not exist.'.format(a=inp))
    else:
        decision= messagebox.askquestion ('Yes or No','Are you sure you want
to delete the data of %s(%s) ?'%(l[inp],inp))
        if decision=='yes':

c=m.connect(host='localhost',user='root',password='2003',database='em')
    o=c.cursor()
    o.execute('delete from details where empno=%s',(inp,))
    c.commit()
    c.close()
    empe.delete(0,'end')

subb=Button(dewin,text='Delete',width=10,height=2,command=dc,bg='black',fg='
gold')
subb.place(x=110,y=120)
def ex():
    dewin.destroy()
exb=Button(dewin,text='Exit',width=10,command=ex,bg='black',fg='gold')
exb.place(x=110,y=170)
dewin.mainloop()

```



```
deletebutton=Button(main,text='DELETE  
DETAILS',command=delete,bg='olivedrab1',fg='mediumorchid4',width=15,height  
=1)
```

```
deletebutton.place(x=160,y=260)
```

```
def view():
```

```
    viewwin=Toplevel()
```

```
    viewwin.title('VIEW DETAILS')
```

```
    viewwin.maxsize(300,250)
```

```
    viewwin.minsize(300,250)
```

```
    viewwin.configure(background='pale green')
```

```
    text=Label(viewwin,text='Enter the Employee number',bg='pale  
green',fg='red',font = "Helvetica 13 bold italic")
```

```
    text.place(x=40,y=30)
```

```
    empe=Entry(viewwin,width=40)
```

```
    empe.place(x=30,y=80)
```

```
    def sc():
```

```
        if empe.get()=="":
```

```
            messagebox.showinfo('INFORMATION','Entry box is empty.')
```

```
        else:
```

```
            c=m.connect(host='localhost',user='root',password='2003',database='em')
```

```
            o=c.cursor()
```

```
            q='select * from details where empno=%s'
```

```
            inp=empe.get()
```

```
            d=(inp,)
```

```
            o.execute(q,d)
```

```
            data=o.fetchall()
```

```
            c.close()
```

```
            empe.delete(0,'end')
```

```
            if len(data)!=0:
```

```
                vtl=Toplevel()
```

```
                vtl.title('  ')
```

```
                vtl.maxsize(500,330)
```

```
                vtl.minsize(500,330)
```

```

        vtl.configure(background='light salmon')
        t=Label(vtl,text="\nDetails",fg='black',bg='light salmon',font =
"Helvetica 13 bold italic")
        t.pack()
        l=['Employee Number      :','Name\t\t      :','Date of birth\t      :',
          'Gender\t\t      :','Father Name\t      :',
          'Mother Name\t      :','Phone Number\t      :',
          'Email Id\t\t      :','Address\t\t      :','Date of joining\t      :',
          'Designation\t      :','Salary\t\t      : Rs.']
        de={'1':'Publisher','2':'Editor in chief','3':'Managing
Director','4':'Assistant Managing Director',
          '5':'audio/visual production specialist','6':'News Editor','7':'Sub
Editor',
          '8':'Photo Editor','9':'Reporter','10':'Video Grapher','11':'Lawyer',
          '12':'Worker in printing section'}
        y1=50
        for h in range(len(l)):
            if h==10:
                da1=l[h]+de[str(data[0][h]])+'\n'
            else:
                da1=l[h]+str(data[0][h])+'\n'
            dp=Label(vtl,text=da1,fg='midnight blue',bg='light salmon')
            dp.place(x=10,y=y1)
            y1+=20
        def eb():
            vtl.destroy()

        exb=Button(vtl,text='Exit',width=10,command=eb,bg='black',fg='gold')
        exb.place(x=230,y=300)
        vtl.mainloop()
    else:
        messagebox.showinfo('INFORMATION','This Employee number( {e}
) does not exist.'.format(e=inp))

```

```

subb=Button(viewwin,text='Submit',width=10,height=2,command=sc,bg='black',fg='gold')
subb.place(x=110,y=120)
def vawc():
    vaw=Toplevel()
    vaw.title('ALL DETAILS')
    vaw.maxsize(400,300)
    vaw.minsize(400,300)
    vaw.configure(background='khaki1')
    de={'1':'Publisher','2':'Editor in chief','3':'Managing Director','4':'Assistant
Managing Director',
'5':'audio/visual production specialist','6':'News Editor','7':'Sub Editor',
'8':'Photo Editor','9':'Reporter','10':'Video Grapher','11':'Lawyer',
'12':'Worker in printing section'}
    c=m.connect(host='localhost',user='root',password='2003',database='em')
    o=c.cursor()
    q='select empno,name,desigco from details'
    o.execute(q)
    data=o.fetchall()
    y1=20
    if len(data)==0:
        messagebox.showinfo('INFORMATION','No data exist.')
    else:
        for i in range(len(data)):
            s=str(i+1)+' '+data[i][0]+' >> '+data[i][1]+' >> '+de[data[i][2]]
            text=Label(vaw,text=s,bg='khaki1',fg='royalblue4')
            text.place(x=20,y=y1)
            y1+=20
    c.close()
def eb():
    vaw.destroy()
exb=Button(vaw,text='Exit',width=10,command=eb,bg='black',fg='gold')
exb.place(x=180,y=270)
vaw.mainloop()

```

```
vawb=Button(viewwin,text='All
Details',width=10,height=2,command=vawc,bg='black',fg='gold')
vawb.place(x=110,y=170)
def eb():
    viewwin.destroy()
exb=Button(viewwin,text='Exit',width=10,command=eb,bg='black',fg='gold')
exb.place(x=110,y=220)
viewwin.mainloop()
```

```
viewbutton=Button(main,text='VIEW
DETAILS',command=view,bg='olivedrab1',fg='mediumorchid4',width=15,height=
1)
```

```
viewbutton.place(x=30,y=230)
def ex():
    c=m.connect(host='localhost',user='root',password='2003',database='em')
    o=c.cursor()
    q='update log_de set lo_outime=curtime() where lo_outime is null'
    o.execute('update log_de set lo_outime=curtime() where lo_outime is null')
    c.commit()
    c.close()
    main.destroy()
    login.destroy()
```

```
exitbutton=Button(main,text='LOGOUT',command=ex,bg='olivedrab1',fg='mediu
morchid4',width=15,height=1)
```

```
exitbutton.place(x=30,y=290)
main.mainloop()
```

```
login=Tk()
login.title('Login')
login.maxsize(300,120)
login.minsize(300,120)
login.configure(background='khaki1')
lo_id=Label(login,text='Login ID',bg='khaki1',fg='red')
lo_id.place(x=20,y=20)
```

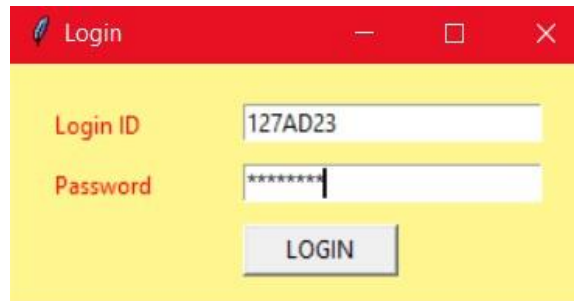
```

lo_e=Entry(login,width=25)
lo_e.place(x=120,y=20)
pwd=Label(login,text='Password',bg='khaki1',fg='red')
pwd.place(x=20,y=50)
pw_e=Entry(login,width=25,show='*')
pw_e.place(x=120,y=50)
def lo_c():
    global log
    log=lo_e.get()
    password=pw_e.get()
    if lo_e.get()=="":
        if pw_e.get()=="":
            messagebox.showinfo('INFORMATION','Login Id and Password is
empty.')
        else:
            messagebox.showinfo('INFORMATION','Login Id is empty.')
    elif pw_e.get()=="":
        messagebox.showinfo('INFORMATION','Password is empty.')
    else:
        c=m.connect(host='localhost',user='root',password='2003',database='em')
        o=c.cursor()
        o.execute('select * from login')
        ed=o.fetchall()
        t=(log,password)
        if t in ed:
            to_d=[lo_id,lo_e,pwd,pw_e,lo_bu]
            for i in to_d:
                i.destroy()
            ou=Label(login,text='You have logged in.',bg='khaki1',fg='red',font =
"bold")
            ou.place(x=30,y=30)
            login.iconify()
            q='insert into
log_de(log_id,lo_date,lo_intime)values(%s,curdate(),curtime())'
            o.execute(q,(log,))

```

```
        c.commit()
        c.close()
        main1()
    else:
        messagebox.showinfo('INFORMATION','Check the login id or password.')
lo_bu=Button(login,text='LOGIN',command=lo_c,width=10)
lo_bu.place(x=120,y=80)
login.mainloop()
```

**OUTPUT**



**LOGIN SCREEN**



**MAIN SCREEN**



**CHANGED LOGIN SCREEN  
AFTER BUTTON IS PRESSED**



The 'CREATE DETAILS' window has a red title bar and a light blue background. It contains a form with the following fields and values:

Enter the details.	
Name	M.Abi
Phone number	9238160174
Date of Birth	3 / 7 / 1975
Email ID	xyz@gmail.com
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female
Address	31, Gandhi nagar, Chennai-600183
Date of Joining	5 / 12 / 1999
Father name	M.Kumar
Designation code	4
Mother name	J.Lakshmi
Salary	25000

At the bottom, there are two buttons: 'CREATE' and 'EXIT'.

**CREATE IS PRESSED IN MAIN SCREEN**

The 'CODE REFERENCE' window has a red title bar and a light gray background. It displays a list of designations with their corresponding codes:

- 1 - Publisher
- 2 - Editor in chief
- 3 - Managing Director
- 4 - Assistant Managing Director
- 5 - audio/visual production specialist
- 6 - News Editor
- 7 - Sub Editor
- 8 - Photo Editor
- 9 - Reporter
- 10 - Video Grapher
- 11 - Lawyer
- 12 - Worker in printing section

**CODE REFERENCE  
WINDOW**

The 'INFORMATION' window has a red title bar and a white background. It displays a message with an information icon:

Data added.  
The employee Number of M.Abi is AM1.

At the bottom right, there is an 'OK' button.

**AUTO GENERATED  
EMPLOYEE NUMBER**

UPDATE DETAILS

Employee number: AM1

Select columns.

- Name
- Date of birth
- Gender
- Father Name
- Mother Name
- Phone Number
- Email Id
- Address
- Date of joining
- Designation
- Salary

Proceed

Exit

UPDATE SCREEN

UPDATE

Date of Birth 3 4 1975

Designation code 4

Salary 30000

UPDATE

WINDOW TO ENTER DATA TO BE UPDATED

INFORMATION

Data updated

OK

UPDATE  
CONFIRMATION

DELETE DETAILS

Employee number

AM1

Delete

Exit

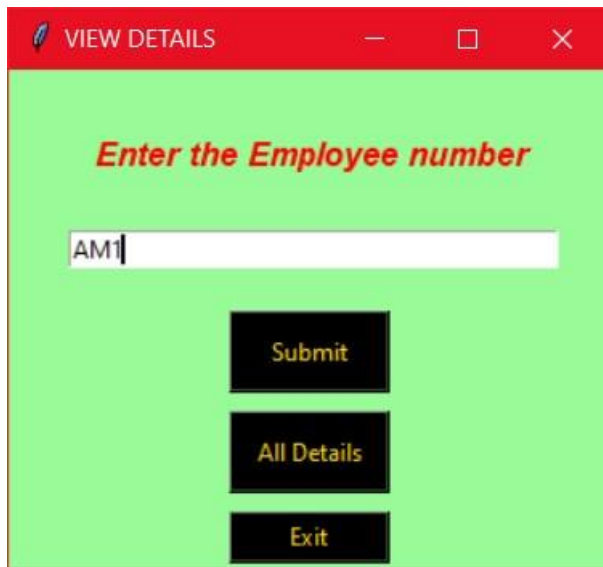
DELETE SCREEN

Yes or No

Are you sure you want to delete the data of M.Abi(AM1) ?

Yes No

CONFIRMATION SCREEN FOR  
DELETING DATA



VIEW DETAILS

*Enter the Employee number*

AM1

Submit

All Details

Exit

This screenshot shows a window titled 'VIEW DETAILS' with a light green background. It prompts the user to 'Enter the Employee number' in red italicized text. A text input field contains 'AM1'. Below the input field are three black buttons with yellow text: 'Submit', 'All Details', and 'Exit'.

VIEW SCREEN



ALL DETAILS

1. AM1 >> M.Abi >> Assistant Managing Director
2. E01 >> a >> Assistant Managing Director
3. E02 >> a >> Editor in chief
4. EA03 >> q >> Publisher
5. EA05 >> d >> Publisher
6. EE03 >> g >> Publisher
7. MD1 >> h >> Managing Director
8. MD2 >> naveen >> Managing Director
9. VG1 >> g >> Video Grapher

Exit

This screenshot shows a window titled 'ALL DETAILS' with a light yellow background. It displays a list of nine employees with their IDs, names, and designations. At the bottom right is a black button with yellow text labeled 'Exit'.

ALL DETAILS SCREEN



**Details**

Employee Number	: AM1
Name	: M.Abi
Date of birth	: 1975-07-03
Gender	: Male
Father Name	: M.Kumar
Mother Name	: J.Lakshmi
Phone Number	: 9238160174
Email Id	: xyz@gmail.com
Address	: 31,Gandhi nagar,Chennai-600183
Date of joining	: 1999-12-05
Designation	: Assistant Managing Director
Salary	: Rs.25000

Exit

This screenshot shows a window titled 'Details' with a light orange background. It displays the details for the employee with ID 'AM1'. The details are listed in a table format. At the bottom center is a black button with yellow text labeled 'Exit'.

DETAILS OF INDIVIDUAL EMPLOYEE NUMBER  
AS ENTERED IN VIEW SCREEN

# **BIBLIOGRAPHY**

 [www.tutorialspoint.com](http://www.tutorialspoint.com)


 [www.geeksforgeeks.com](http://www.geeksforgeeks.com)

 [www.javatpoint.com](http://www.javatpoint.com)

 [www.realpython.com](http://www.realpython.com)

 [www.datacamp.com](http://www.datacamp.com)

 Computer science with Python with sumita aurora

 You Tube

THANK YOU