

DMS PROJECT REPORT

Project Done By: Utsav Agarwal, Yash Agarwal

Section B

ICAS

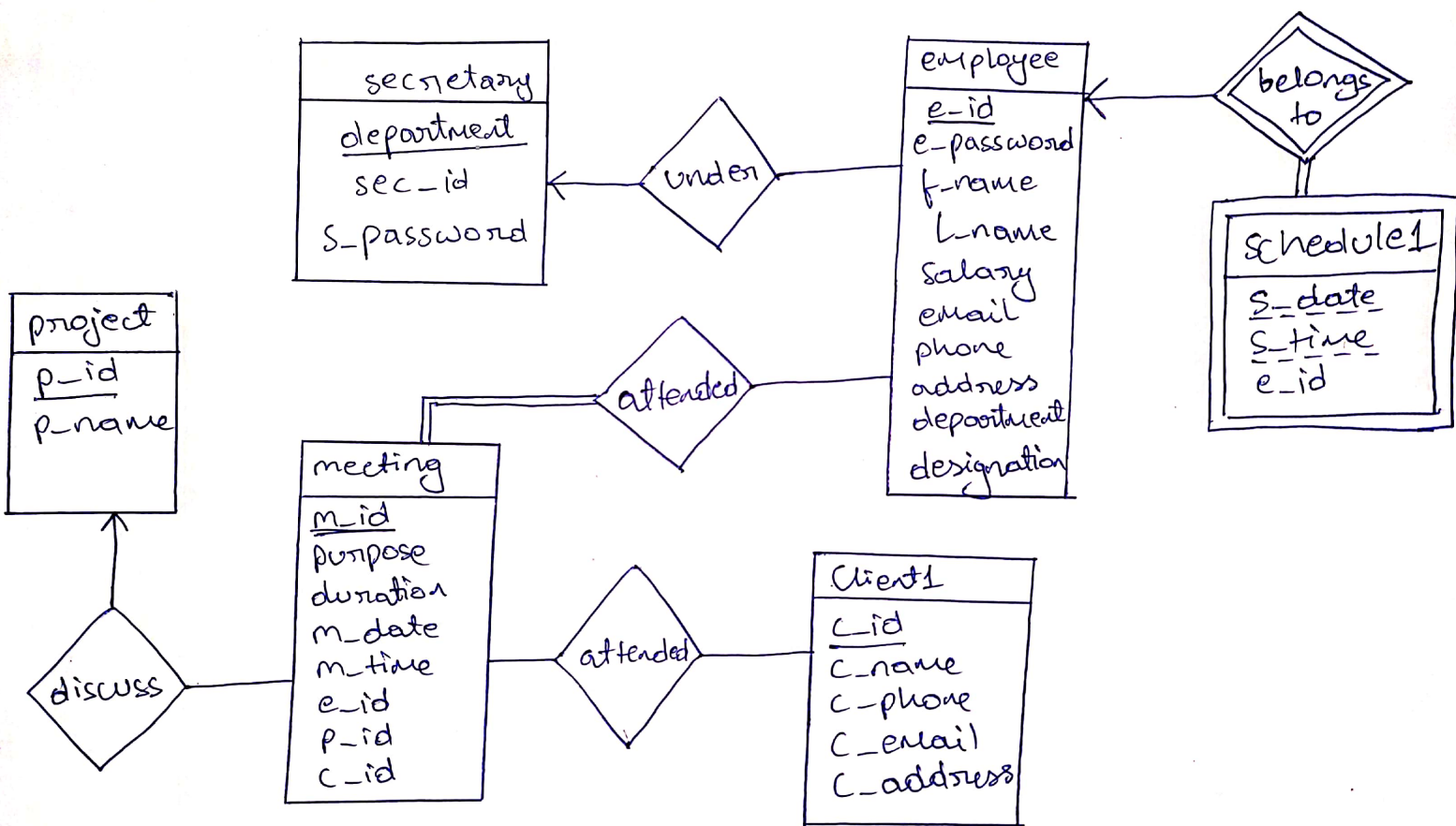
Registration Numbers: 181627116, 181627114

Subject: Database Management Systems

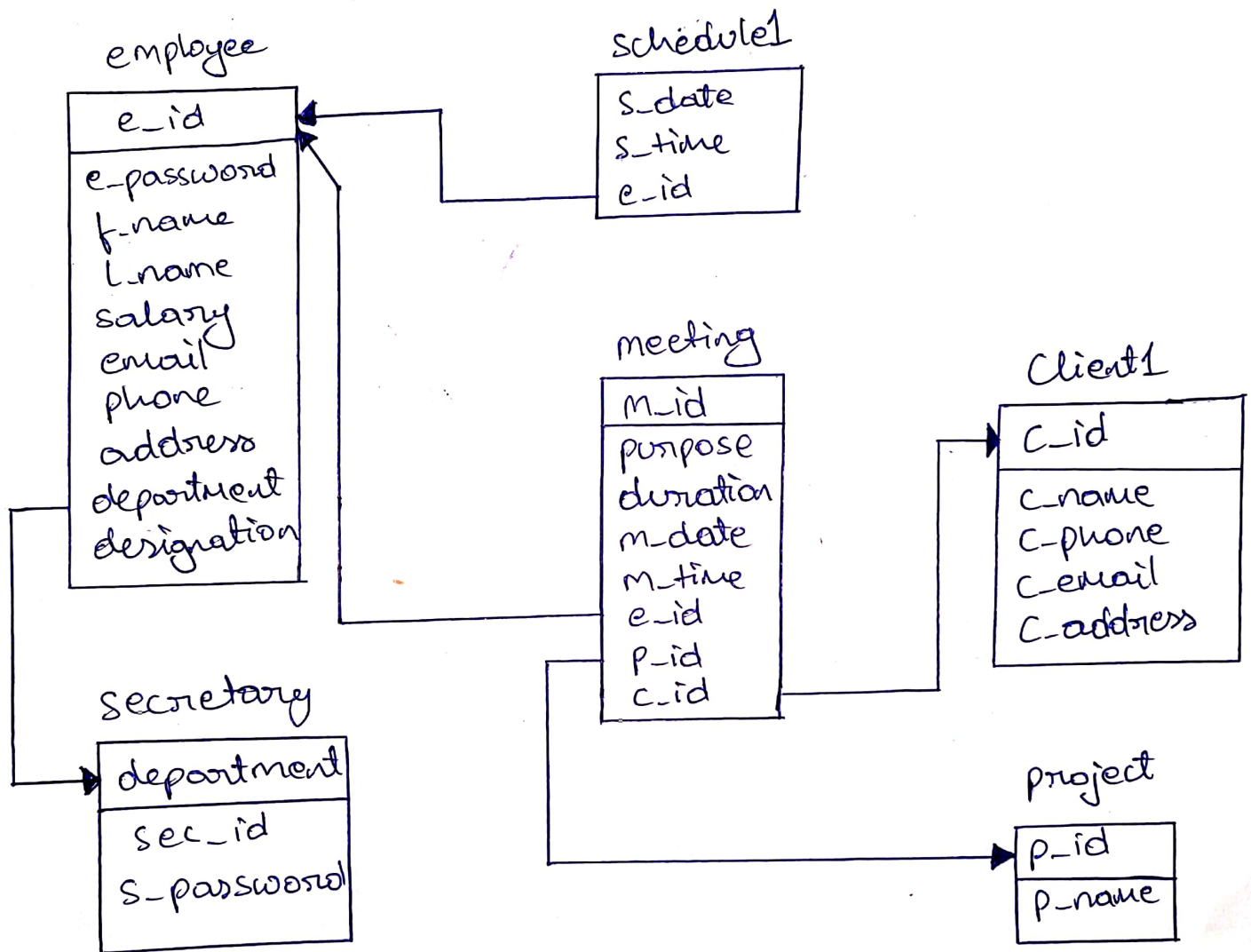
Languages Used:

- MySql
- Python

ER DIAGRAM



SCHEMA DIAGRAM



MySQL code

```
use db2;
```

```
create table secretary(sec_id numeric(7,0), s_password varchar(20), department varchar(20),  
                        primary key(department));
```

```
create table employee( e_id numeric(5,0) , e_password varchar(20), f_name varchar(10),  
                        l_name varchar(10), salary numeric(10,0), email  
varchar(40),  
                        phone numeric(10,0), address varchar(50), department varchar(20),  
                        designation varchar(25), primary key(e_id),  
                        foreign key(department) references secretary(department) on delete set null);
```

```
create table Client1(c_id numeric(5,0), c_name varchar(30), c_phone numeric(10,0),  
                     c_email varchar(40), c_address varchar(50), primary  
key(c_id));
```

```
create table project(p_name varchar(30), p_id numeric(5,0), primary key(p_id));
```

```
create table schedule1(s_date date, s_time time, e_id numeric(5,0));
```

```
create table meeting(m_id numeric(6,0), purpose varchar(50), duration varchar(7), e_id  
numeric(5,0),  
                     m_date date, m_time time, p_id numeric(5,0), c_id  
numeric(5,0), primary key(m_id),  
                     foreign key(p_id) references project(p_id), foreign key(c_id) references Client1(c_id),  
                     foreign key(e_id) references employee(e_id) on delete set null);
```

```
insert into secretary values(1234567 , 'Asdfgh' , 'Power');
```

```
insert into secretary values(1234568 , 'Bsd fgh' , 'Public Relations');
```

```
insert into secretary values(1234569 , 'Csdfigh' , 'Business');

insert into employee values(11111 , 'asd12' , 'Chandan' , 'Prabhu' , 100000.0 , 'cpr@tms.com' ,
9876543210 , 'New Delhi' , 'Business' , 'Clerk');

insert into employee values(11112 , 'asd13' , 'Asif' , 'Khan' , 109600.0 , 'akh@tms.com' , 9876777210
, 'Mumbai' , 'Public Relations' , 'Executive Director');

insert into employee values(11113 , 'asd14' , 'Bhushan' , 'Kumar' , 365000.0 , 'bku@tms.com' ,
9111543210 , 'Chennai' , 'Public Relations' , 'Clerk');

insert into Client1 values(25251 , 'Raj Kumar' , 9297374545 , 'rjk@gmail.com' , 'Gurgaon');

insert into Client1 values(25252 , 'Priya Prakash' , 9296664545 , 'prpk@gmail.com' , 'Faridabad');

insert into Client1 values(25253 , 'Jethalal' , 7371212456 , 'jeth@gmail.com' , 'Gokuldham');

insert into project values('Heavy Building Material' , 55110);

insert into project values('Bridge of Bombay' , 55111);

insert into project values('Dharavi Sports Complex' , 55112);

insert into schedule1 values(2018-11-27 , '10:00:00' , 11112);

insert into schedule1 values(2018-11-27 , '14:30:00' , 11113);

insert into schedule1 values(2018-11-28 , '09:25:00' , 11111);

insert into meeting values(112230 , 'Discussing Budget' , '2 hours' , 11111 , 2018-11-28 , '09:25:00' ,
NULL , 25253);

insert into meeting values(112231 , 'Nominating Project Head' , '1 hour' , 11112 , 2018-11-27 ,
'14:30:00' , 55111 , NULL);

insert into meeting values(112231 , 'Nominating Project Head' , '1 hour' , 11113 , 2018-11-27 ,
'14:30:00' , 55111 , NULL);
```

FRONT END (GUI) – CODE

(python 3.7.4)

```
import pymysql
import time
import logging
import threading
from datetime import datetime
from tkinter import *
import tkinter as tk

db= pymysql.connect('localhost','Utsav','1234','db2')
cr=db.cursor()

parent=Tk()

label1=Label(parent,text="Emp ID")
label1.grid(row=2,column=1)

label2=Label(parent,text="Password")
label2.grid(row=3,column=1)

TextField1=Entry(parent)
TextField1.grid(row=2,column=4)

TextField2=Entry(parent)
```

```
TextField2.grid(row=3,column=4)
```

```
label3=Label(parent,text="")
```

```
label3.grid(row = 6, column= 2)
```

```
#####
```

```
##LOGIN##
```

```
def login(lab,entry1,entry2):
```

```
    e_id=entry1.get()
```

```
    res2=entry2.get()
```

```
    cr=db.cursor()
```

```
    cr.execute('select e_password from employee where e_id = %d;' % int(e_id))
```

```
    for i in cr:
```

```
        if(i[0]==res2):
```

```
            mainBox(e_id)
```

```
        else:
```

```
            lab.configure(text='Incorrect Emp ID / password')
```

```
button1=Button(parent,text="login", command = lambda: login(label3,TextField1,TextField2))
```

```
button1.grid(row=8,column=1)
```

```
#####
```

```
##SEC LOGIN##
```

```
def secLogin():
```

```
    secLog=Toplevel(parent)
```

```
label1=Label(secLog,text="Sec ID")
```

```
label1.grid(row=2,column=1)
```

```
label2=Label(secLog,text="Password")
```

```
label2.grid(row=3,column=1)
```

```
TextField1=Entry(secLog)
```

```
TextField1.grid(row=2,column=4)
```

```
TextField2=Entry(secLog)
```

```
TextField2.grid(row=3,column=4)
```

```
label3=Label(secLog,text="")
```

```
label3.grid(row = 6, column= 2)
```

```
but=Button(secLog, text="login", command= lambda:readInp(TextField1,TextField2,secLog))
```

```
but.grid(row = 5, column = 2)
```

```
secLog.mainloop()
```

```
def readInp(tF1,tF2,secLog):
```

```
    s_id=tF1.get()
```

```
    s_password=tF2.get()
```

```
    cr=db.cursor()
```

```
    cr.execute('select s_password from secretary where sec_id = %d;%int(s_id))
```

```
    for i in cr:
```

```
        if i[0]==s_password:
```

```
            secMain(secLog)
```

```
        else:
```

```
            lab.configure(text='Incorrect Sec ID / password')
```



```
button2=Button(parent,text="Secretary Login", command= secLogin)
button2.grid(row=8,column=4)
```

```
#####
##SEC MAIN##
```

```
def secMain(secLog):
    secLog.destroy()
```

```
secMain = Toplevel(parent)
```

```
mainBut1=Button(secMain,text='Check Schedule',command=lambda:checkSchedule())
mainBut1.grid(row=1,column=1)
```

```
mainBut2=Button(secMain,text='Schedule Meeting',command=lambda:scheduleMeeting())
mainBut2.grid(row=1,column=2)
```

```
mainBut3=Button(secMain,text='Cancel Meeting',command=lambda:cancel())
mainBut3.grid(row=2,column=2)
```

```
mainBut5=Button(secMain,text='Logout',command=lambda: logout(secMain))
mainBut5.grid(row=3,column=5)
```

```
mainBut4=Button(secMain,text='Create Exec Account',command= newBox)
```

```
mainBut4.grid(row=1,column=3)
```

```
mainBut6=Button(secMain,text='Schedule Exec task',command= taskSch)
```

```
mainBut6.grid(row=1,column=5)
```

```
lab = Label(secMain, text = "* Always logout before quitting! *")
```

```
lab.grid(row=3, column=2)
```

```
secMain.mainloop()
```

```
def logout(box):
```

```
    box.destroy()
```

```
    parent.deiconify()
```

```
def cancel():
```

```
    cancelDet=Toplevel(parent)
```

```
    lab=Label(cancelDet, text = "Meeting ID")
```

```
    lab.grid(row=1,column=2)
```

```
    txtField=Entry(cancelDet)
```

```
    txtField.grid(row = 1, column = 4)
```

```
    canBut=Button(cancelDet,text="confirm",command= lambda: cr.execute("delete from meeting  
where m_id = %d;" ,(int(txtField.get()))))
```

```
    canBut.grid(row = 3, column = 3)
```

```
    cr=db.cursor()
```

```
def taskSch():
```

```
    taskBox=Toplevel(parent)
```

```
    txtField1=Entry(taskBox)
```

```
txtField1.grid(row = 1, column = 2)
txtField2=Entry(taskBox)
txtField2.grid(row = 2, column = 2)
txtField3=Entry(taskBox)
txtField3.grid(row = 3, column = 2)
```

```
lab1= Label(taskBox, text = 'Date')
lab1.grid(row = 1, column =1)
lab2= Label(taskBox, text = 'Time')
lab2.grid(row = 2, column =1)
lab3= Label(taskBox, text = 'Emp ID')
lab3.grid(row = 3, column =1)
```

```
taskBut=Button(taskBox,text='Confirm',command= lambda: taskSet(txtField1,txtField2,txtField3))
taskBut.grid(row = 5, column= 2)
taskBox.mainloop()
```

```
def taskSet(entry1,entry2,entry3):
```

```
    s_date=datetime.strptime(entry6.get(),'%y-%m-%d').date()
    s_time=datetime.strptime(entry7.get(),'%H:%M:%S').time()
    e_id=entry3.get()
```

```
    cr=db.cursor()
```

```
    cr.execute("insert into schedule1(s_date,s_time,e_id) values (%s,%s,%d);",(s_date,s_time,e_id))
```

```
def checkSchedule():
```

```

schedule=Toplevel(parent)
schLab1=Label(schedule,text="")
schLab1.grid(row=1,column=0)
cr=db.cursor()
cr.execute('delete from schedule1 where s_date < CURDATE();')
cr.execute('delete from meeting where m_date < CURDATE();')

schLab1.configure(text='You have '+str(cr.execute('select * from schedule1, meeting where
schedule1.s_date = CURDATE() and meeting.m_date=curdate();'))+' meetings and appointments
today' )

#scrollbar = tk.Scrollbar(schedule,orient='vertical')

Lb = Listbox(schedule, width=50, height=20)

x=0
for i in cr:
    x=x+1
    Lb.insert(x,i)
Lb.grid()

scBut1=Button(schedule,text="view all meetings",command = lambda: allEmpMeetings(Lb))
scBut1.grid(row=3, column=0)

scBut2=Button(schedule,text="view today's meetings",command = lambda:
todayEmpMeetings(Lb))
scBut2.grid(row=5, column=0)

mainBut2=Button(schedule,text='Schedule Meeting',command=lambda:scheduleEmpMeeting(Lb))
mainBut2.grid(row=4,column=0)

schedule.mainloop()

```

```

def allMeetings(Lb):
    Lb.delete(0,'end')
    cr.execute('select * from schedule1;')
    x=0
    for i in cr:
        x=x+1
        Lb.insert(x,i)

```

```
def todayMeetings(Lb):  
    Lb.delete(0,'end')  
    cr.execute('select * from schedule1 where s_date = CURDATE();')  
    x=0  
    for i in cr:  
        x=x+1  
        Lb.insert(x,i)
```

```
def scheduleMeeting():  
    secSchMeeting=Tk()  
    txtField1=Entry(secSchMeeting)  
    txtField1.grid(row = 2, column = 2)  
    txtField2=Entry(secSchMeeting)  
    txtField2.grid(row = 4, column = 2)  
    txtField3=Entry(secSchMeeting)  
    txtField3.grid(row = 6, column = 2)  
    txtField4=Entry(secSchMeeting)  
    txtField4.grid(row = 8, column = 2)  
    txtField5=Entry(secSchMeeting)  
    txtField5.grid(row = 10, column = 2)  
    txtField6=Entry(secSchMeeting)  
    txtField6.grid(row = 12, column = 2)  
    txtField7=Entry(secSchMeeting)  
    txtField7.grid(row = 14, column = 2)
```

```
lab1= Label(secSchMeeting, text = 'Meeting ID')  
lab1.grid(row = 2, column =1)  
lab2= Label(secSchMeeting, text = 'Project ID')
```

```

lab2.grid(row = 4, column =1)

lab3= Label(secSchMeeting, text = 'Client ID')

lab3.grid(row = 6, column =1)

lab4= Label(secSchMeeting, text = 'Purpose')

lab4.grid(row = 8, column =1)

lab5= Label(secSchMeeting, text = 'Duration(mins)')

lab5.grid(row = 10, column =1)

lab6= Label(secSchMeeting, text = 'Date')

lab6.grid(row = 12, column =1)

lab7= Label(secSchMeeting, text = 'Time')

lab7.grid(row = 14, column =1)

```

```

but1= Button(secSchMeeting, text = 'make appointment',command =
lambda:makeAppointment(secSchMeeting,txtField1,txtField2,txtField3,txtField4,txtField5,txtField6,t
xtField7))

```

```

but1.grid(row= 16, column = 1)

secSchMeeting.mainloop()

```

```

def makeAppointment(box,entry1,entry2,entry3,entry4,entry5,entry6,entry7):

```

```

    m_id=int(entry1.get())
    p_id=int(entry2.get())
    c_id=int(entry3.get())
    purpose=entry4.get()
    duration=int(entry5.get())
    m_date=datetime.strptime(entry6.get(),'%Y-%m-%d').date()
    m_time=datetime.strptime(entry7.get(),'%H:%M:%S').time()

```

```
cr.execute('insert into meeting(m_id,p_id,c_id,purpose,duration,m_date,m_time)
values(%d,%d,%d,%s,%d,%s,%s) ;', (m_id,p_id,c_id,purpose,duration,m_date,m_time)) ## <<l add
client in sql and code here
```

```
x=Toplevel(parent)

lab = Label(x, text='Meeting Scheduled')

lab.grid(row = 1 , column = 2)

butConf=Button(x,text = 'ok', command = lambda :ok(x))

butConf.grid(row=2 , column =2)

x.mainloop()
```

```
def ok(x):
```

```
    x.destroy()
```

```
#####
```

```
##EMP SIGNUP##
```

```
def newBox():
```

```
    box = Toplevel(parent)

    lab1=Label(box,text="E ID")

    lab1.grid(row=2,column=1)

    lab2=Label(box,text="Password")

    lab2.grid(row=3,column=1)

    lab3=Label(box,text="First Name")
```

```
lab3.grid(row=4,column=1)
lab4=Label(box,text="Last Name")
lab4.grid(row=5,column=1)
lab5=Label(box,text="Salary")
lab5.grid(row=6,column=1)
lab6=Label(box,text="Email")
lab6.grid(row=7,column=1)
lab7=Label(box,text="Phone No.")
lab7.grid(row=8,column=1)
lab8=Label(box,text="Address")
lab8.grid(row=9,column=1)
lab9=Label(box,text="Department")
lab9.grid(row=10,column=1)
lab10=Label(box,text="Designation")
lab10.grid(row=11,column=1)
lab11=Label(box,text="")
lab11.grid(row=13,column=2)
```

```
Text1=Entry(box)
Text1.grid(row=2,column=4)
Text2=Entry(box)
Text2.grid(row=3,column=4)
Text3=Entry(box)
Text3.grid(row=4,column=4)
Text4=Entry(box)
Text4.grid(row=5,column=4)
Text5=Entry(box)
Text5.grid(row=6,column=4)
Text6=Entry(box)
Text6.grid(row=7,column=4)
Text7=Entry(box)
```



```
Text7.grid(row=8,column=4)
```

```
Text8=Entry(box)
```

```
Text8.grid(row=9,column=4)
```

```
Text9=Entry(box)
```

```
Text9.grid(row=10,column=4)
```

```
Text10=Entry(box)
```

```
Text10.grid(row=11,column=4)
```

```
but=Button(box,text="confirm", command=  
lambda:readInpE(box,Text1,Text2,Text3,Text4,Text5,Text6,Text7,Text8,Text9,Text10,lab11))
```

```
but.grid(row=13,column=3)
```

```
box.mainloop()
```

```
def readInpE(box,entry1,entry2,entry3,entry4,entry5,entry6,entry7,entry8,entry9,entry10,lab11):
```

```
    e_id=int(entry1.get())
```

```
    e_password=str(entry2.get())
```

```
    f_name=str(entry3.get())
```

```
    l_name=str(entry4.get())
```

```
    salary=int(entry5.get())
```

```
    email=str(entry6.get())
```

```
    phone=int(entry7.get())
```

```
    address=str(entry8.get())
```

```
    department=str(entry9.get())
```

```
    designation=str(entry10.get())
```

```
    cr=db.cursor()
```

```
    cr.execute('insert into
```

```
Employee(e_id,e_password,f_name,l_name,salary,email,phone,address,department,designation)
```

```
values(%d,%s,%s,%s,%d,%s,%d,%s,%s,%s);',(e_id,e_password,f_name,l_name,salary,email,phone,ad  
dress,department,designation))
```

```
    lab11.configure(text='Registered')
```

```
    db.commit()
```

```
#####
```

```
##EMP MAIN##
```

```
def mainBox(e_id):
```

```
    parent.withdraw()
```

```
    mainBox = Toplevel(parent)
```

```
    mainBut1=Button(mainBox,text='Check Schedule',command=lambda:checkEmpSchedule(e_id))
```

```
    mainBut1.grid(row=1,column=1)
```

```
    mainBut2=Button(mainBox,text='Schedule  
Meeting',command=lambda:scheduleEmpMeeting(e_id))
```

```
    mainBut2.grid(row=1,column=2)
```

```
    mainBut2=Button(mainBox,text='Schedule Task',command=lambda:taskSch())
```

```
    mainBut2.grid(row=1,column=3)
```

```
    mainBut3=Button(mainBox,text='Cancel Meeting',command=lambda:cancelEmp(e_id))
```

```
    mainBut3.grid(row=2,column=2)
```

```
    mainBut4=Button(mainBox,text='Logout',command=lambda:Logout(mainBox))
```

```
    mainBut4.grid(row=2,column=3)
```

```
    lab = Label(mainBox, text = "* Always logout before quitting! *")
```

```
    lab.grid(row=3, column=2)
```

```
    mainBox.mainloop()
```

```

def Logout(box):
    box.destroy()
    parent.deiconify()

def checkEmpSchedule(e_id):
    schedule=Toplevel(parent)
    schLab1=Label(schedule,text="")
    schLab1.grid(row=1,column=0)
    cr=db.cursor()
    cr.execute('delete from schedule1 where s_date < CURDATE();')
    schLab1.configure(text='You have '+str(cr.execute('select * from schedule1, meeting where
    schedule1.s_date = CURDATE() and meeting.m_date=curdate() and meeting.e_id = %d and
    schedule1.e_id = %d;',(e_id,e_id)))+ ' meetings today')

    Lb = Listbox(schedule, width=50, height=20)
    x=0
    for i in cr:
        x=x+1
        Lb.insert(x,i)
    Lb.grid()
    scBut1=Button(schedule,text="view all meetings",command = lambda: allEmpMeetings(e_id,Lb))
    scBut1.grid(row=3, column=0)
    scBut2=Button(schedule,text="view today's meetings",command = lambda:
    todayEmpMeetings(e_id,Lb))
    scBut2.grid(row=5, column=0)
    mainBut2=Button(schedule,text='Schedule
    Meeting',command=lambda:scheduleEmpMeeting(e_id,Lb))
    mainBut2.grid(row=4,column=0)
    schedule.mainloop()

def allEmpMeetings(e_id,Lb):

```

```

Lb.delete(0,'end')
cr.execute('select * from schedule1 where e_id= %d;',(e_id))
x=0
for i in cr:
    x=x+1
    Lb.insert(x,i)

```

```

def todayEmpMeetings(e_id,Lb):
    Lb.delete(0,'end')
    cr.execute('select * from schedule1 where s_date = CURDATE() and e_id= %d;',(e_id))
    x=0
    for i in cr:
        x=x+1
        Lb.insert(x,i)

```

```

def cancelEmp(e_id):
    cancelDet=Toplevel(parent)
    lab=Label(cancelDet, text = "Meeting ID")
    lab.grid(row=1,column=2)
    txtField=Entry(cancelDet)
    txtField.grid(row = 1, column = 4)
    canBut=Button(cancelDet,text="confirm",command= lambda: cr.execute("delete from meeting
where m_id = %d and e_id = %d;",(int(txtField.get()),e_id)))
    canBut.grid(row = 3, column = 3)
    cr=db.cursor()

```

```

def scheduleEmpMeeting(e_id):
    secSchMeeting=Tk()
    txtField1=Entry(secSchMeeting)
    txtField1.grid(row = 2, column = 2)
    txtField2=Entry(secSchMeeting)
    txtField2.grid(row = 4, column = 2)
    txtField3=Entry(secSchMeeting)
    txtField3.grid(row = 6, column = 2)
    txtField4=Entry(secSchMeeting)
    txtField4.grid(row = 8, column = 2)
    txtField5=Entry(secSchMeeting)
    txtField5.grid(row = 10, column = 2)
    txtField6=Entry(secSchMeeting)
    txtField6.grid(row = 12, column = 2)
    txtField7=Entry(secSchMeeting)
    txtField7.grid(row = 14, column = 2)

    lab1= Label(secSchMeeting, text = 'Meeting ID')
    lab1.grid(row = 2, column =1)
    lab2= Label(secSchMeeting, text = 'Project ID')
    lab2.grid(row = 4, column =1)
    lab3= Label(secSchMeeting, text = 'Client ID')
    lab3.grid(row = 6, column =1)
    lab4= Label(secSchMeeting, text = 'Purpose')
    lab4.grid(row = 8, column =1)
    lab5= Label(secSchMeeting, text = 'Duration(mins)')
    lab5.grid(row = 10, column =1)
    lab6= Label(secSchMeeting, text = 'Date')
    lab6.grid(row = 12, column =1)

```

```

lab7= Label(secSchMeeting, text = 'Time')

lab7.grid(row = 14, column =1)


but1= Button(secSchMeeting, text = 'make appointment',command =
lambda:makeAppointment(secSchMeeting,txtField1,txtField2,txtField3,txtField4,txtField5,txtField6,t
xtField7,e_id))

but1.grid(row= 16, column = 1)

secSchMeeting.mainloop()


def makeEmpAppointment(box,entry1,entry2,entry3,entry4,entry5,entry6,entry7,e_id):
    m_id=int(entry1.get())
    p_id=int(entry2.get())
    c_id=int(entry3.get())
    purpose=entry4.get()
    duration=int(entry5.get())
    m_date=datetime.strptime(entry6.get(),'%y-%m-%d').date()
    m_time=datetime.strptime(entry7.get(),'%H:%M:%S').time()

    cr.execute('insert into meeting(m_id,p_id,c_id,e_id,purpose,duration,m_date,m_time)
values(%d,%d,%d,%d,%s,%d,%s,%s) ;', (m_id,p_id,c_id,e_id,purpose,duration,m_date,m_time)) ##
<<I add client in sql and code here

x=Toplevel(parent)

lab = Label(x, text='Meeting Scheduled')

lab.grid(row = 1 , column = 2)

butConf=Button(x,text = 'ok', command = lambda :ok(x))

butConf.grid(row=2 , column =2)

x.mainloop()

parent.mainloop()

```

Queries utilized

```
select e_password from employee where e_id = %d;

select s_password from secretary where sec_id = %d;

delete from meeting where m_id = %d;

insert into schedule1(s_date,s_time,e_id) values (%s,%s,%d);

select * from schedule1, meeting where schedule1.s_date = CURDATE() and
meeting.m_date=CURDATE() and meeting.e_id=%d and meeting.e_id=%d;

select * from schedule1 where s_date = CURDATE();

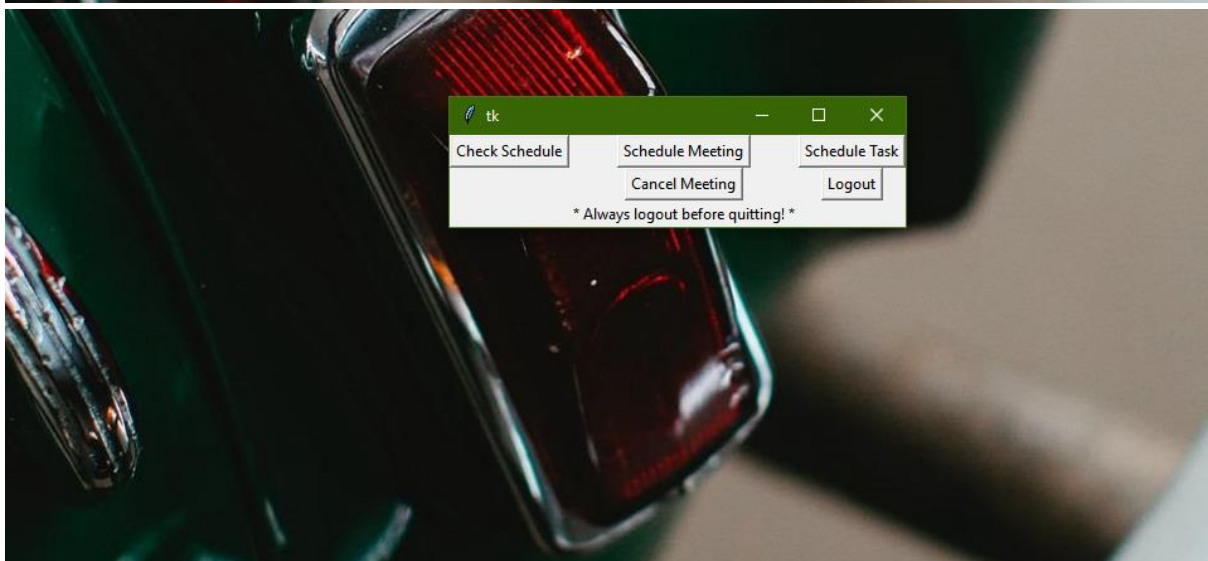
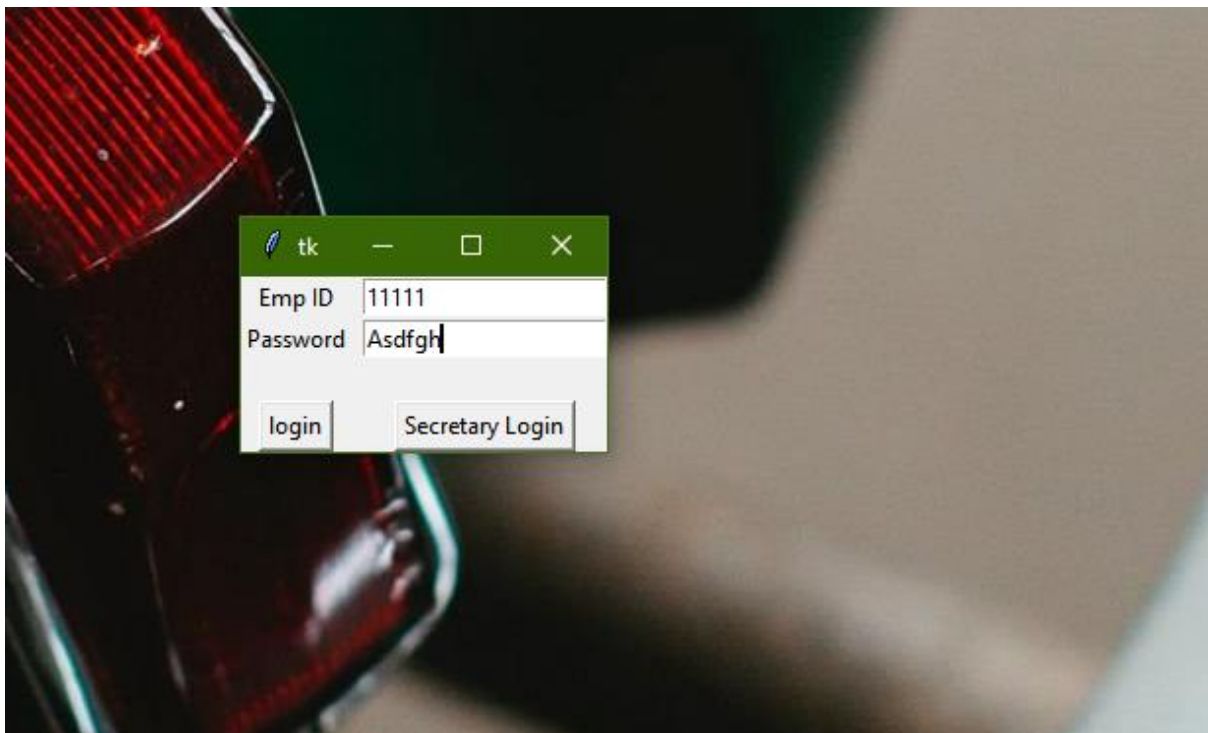
select * from schedule1 where s_date = CURDATE() and e_id= %d;

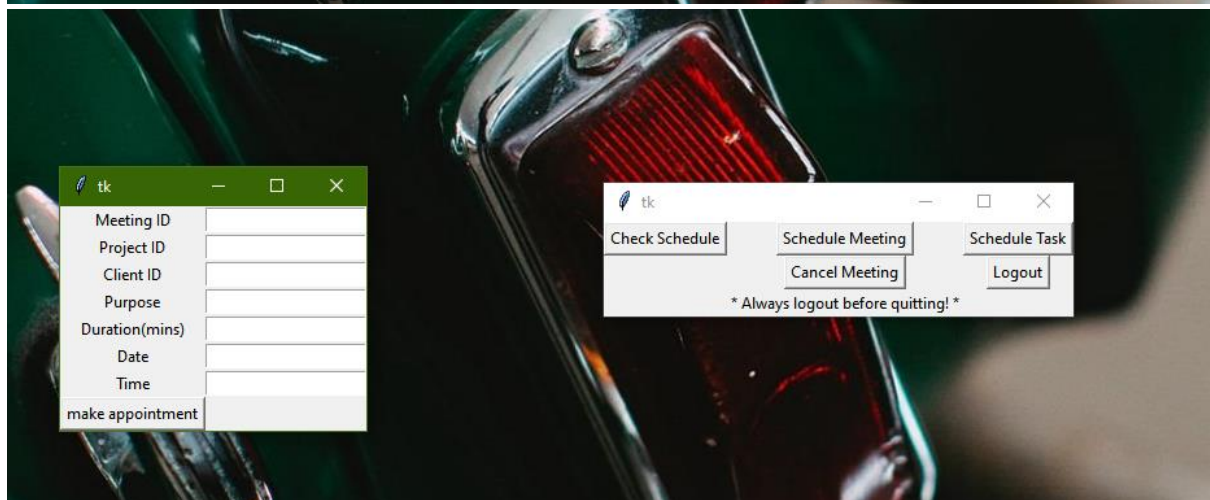
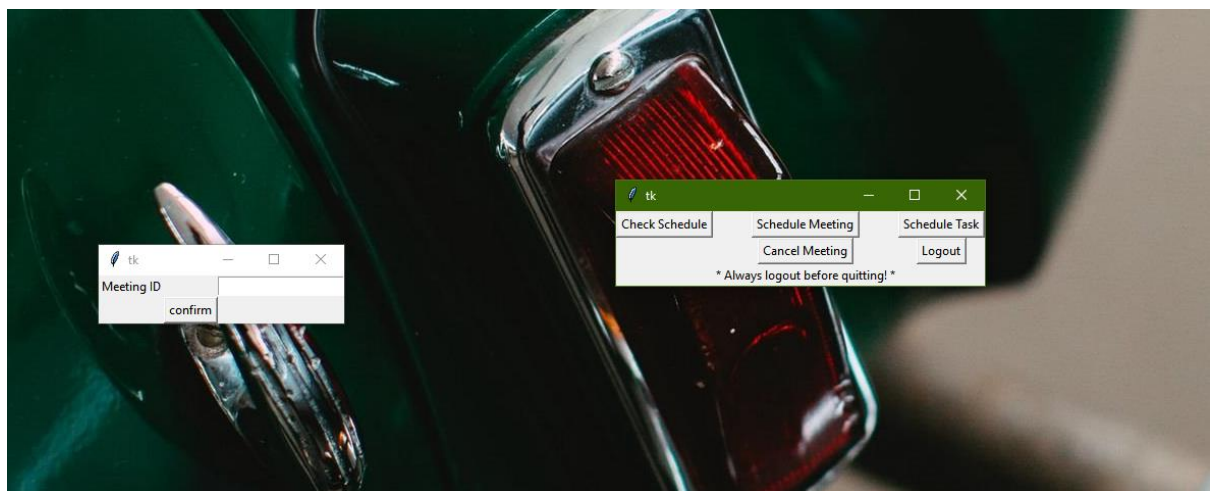
delete from meeting where m_id = %d and e_id = %d;

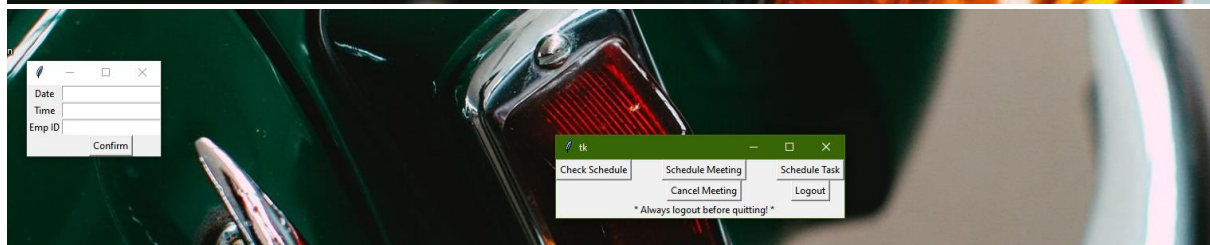
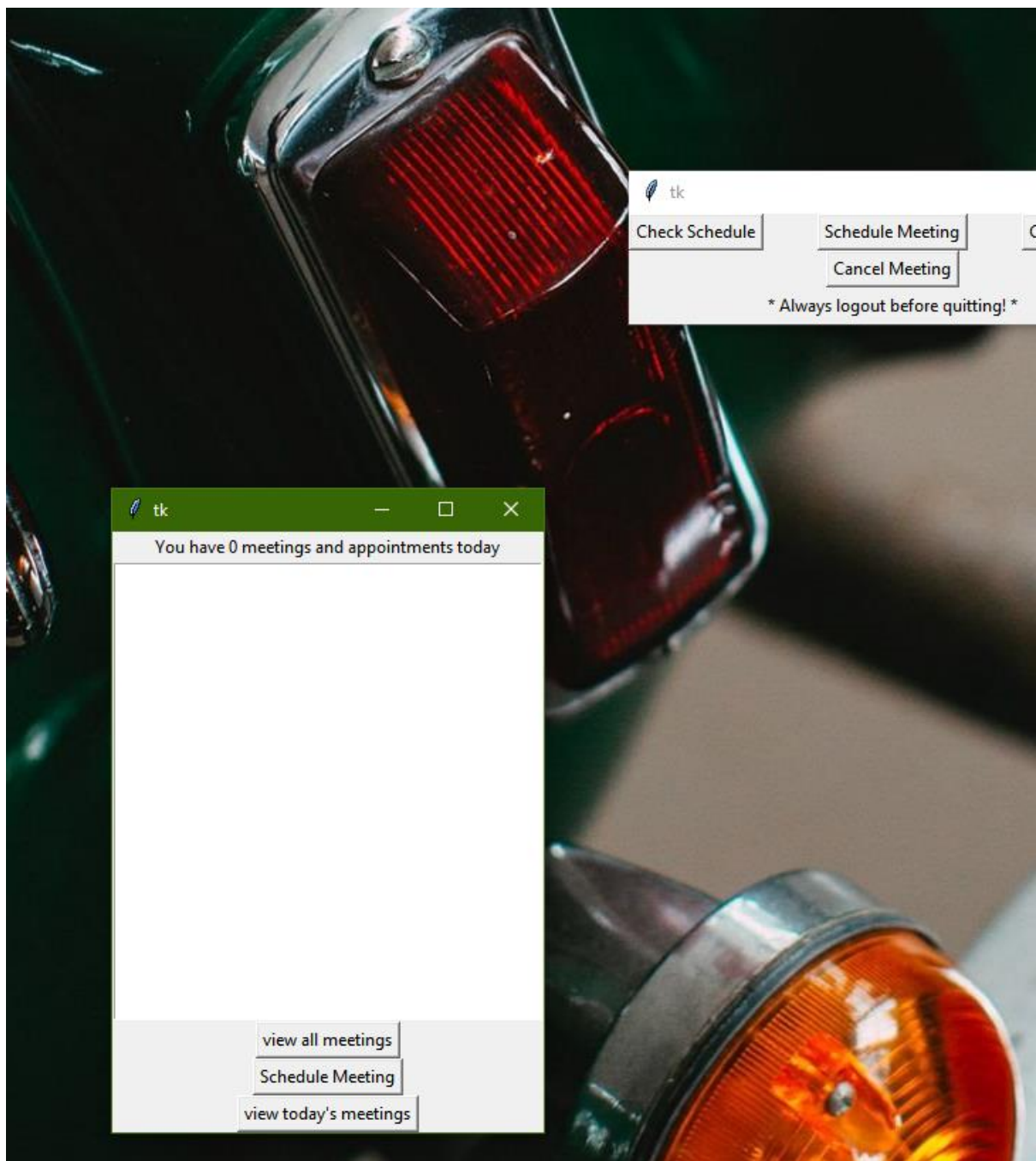
insert into meeting(m_id,p_id,c_id,e_id,purpose,duration,m_date,m_time)
values(%d,%d,%d,%d,%s,%d,%s,%s) ;
```

WORKING PROOF

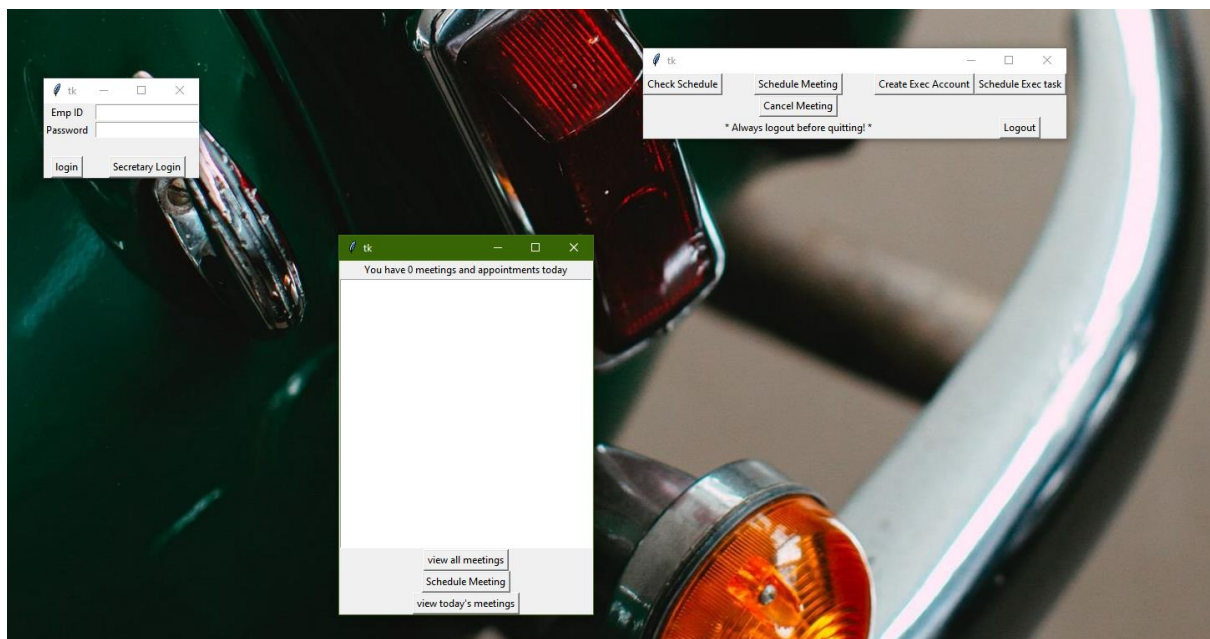
(employee)

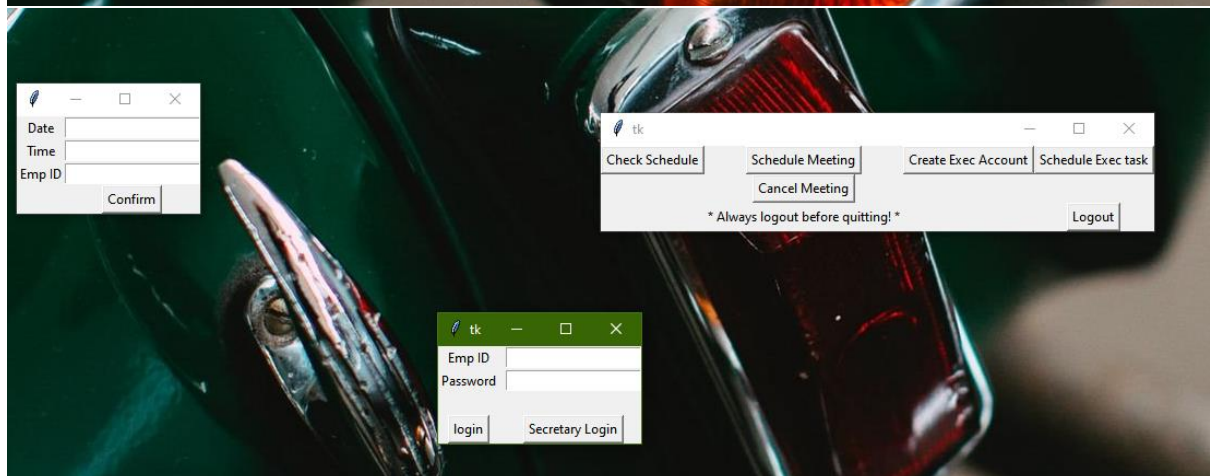
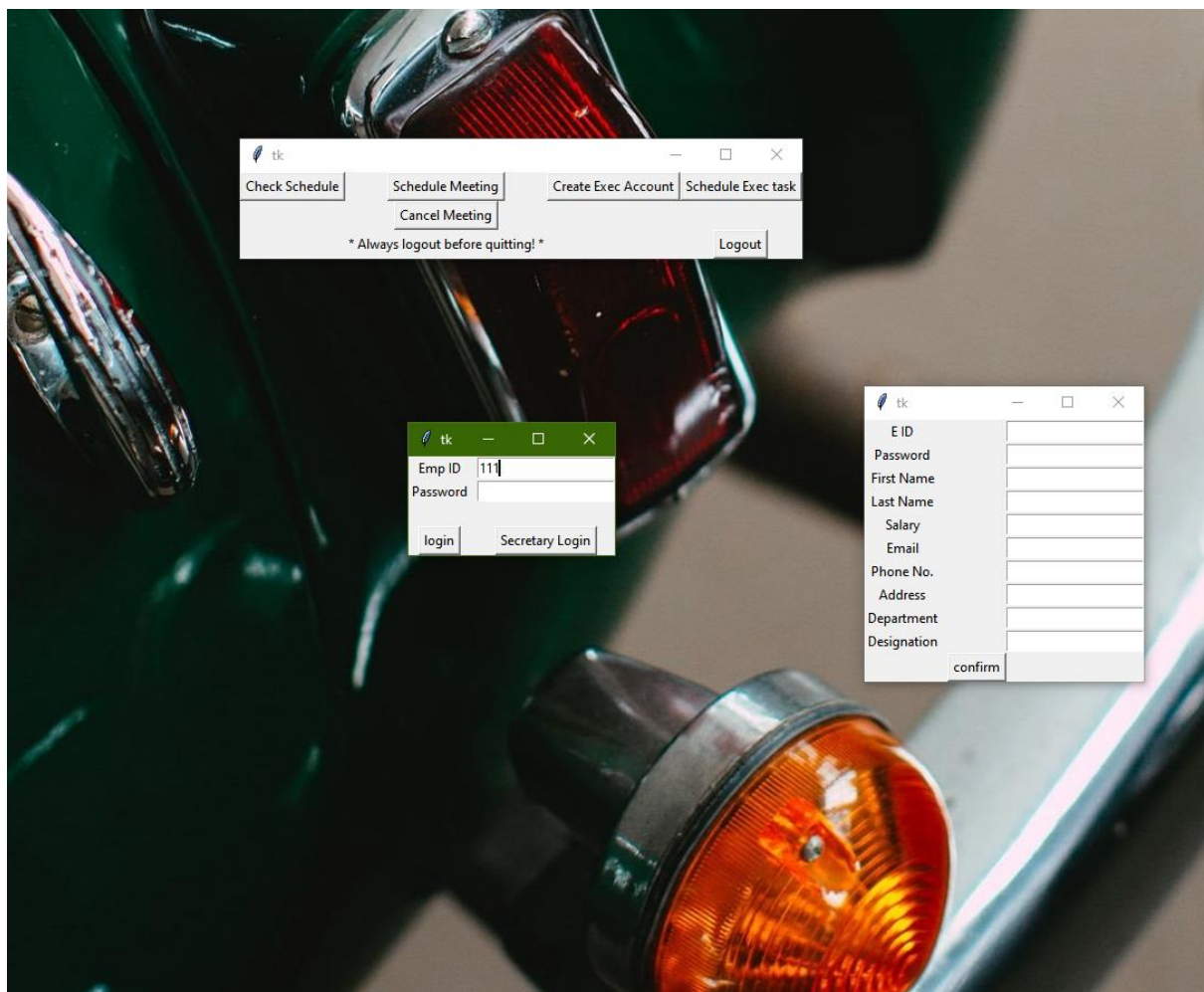


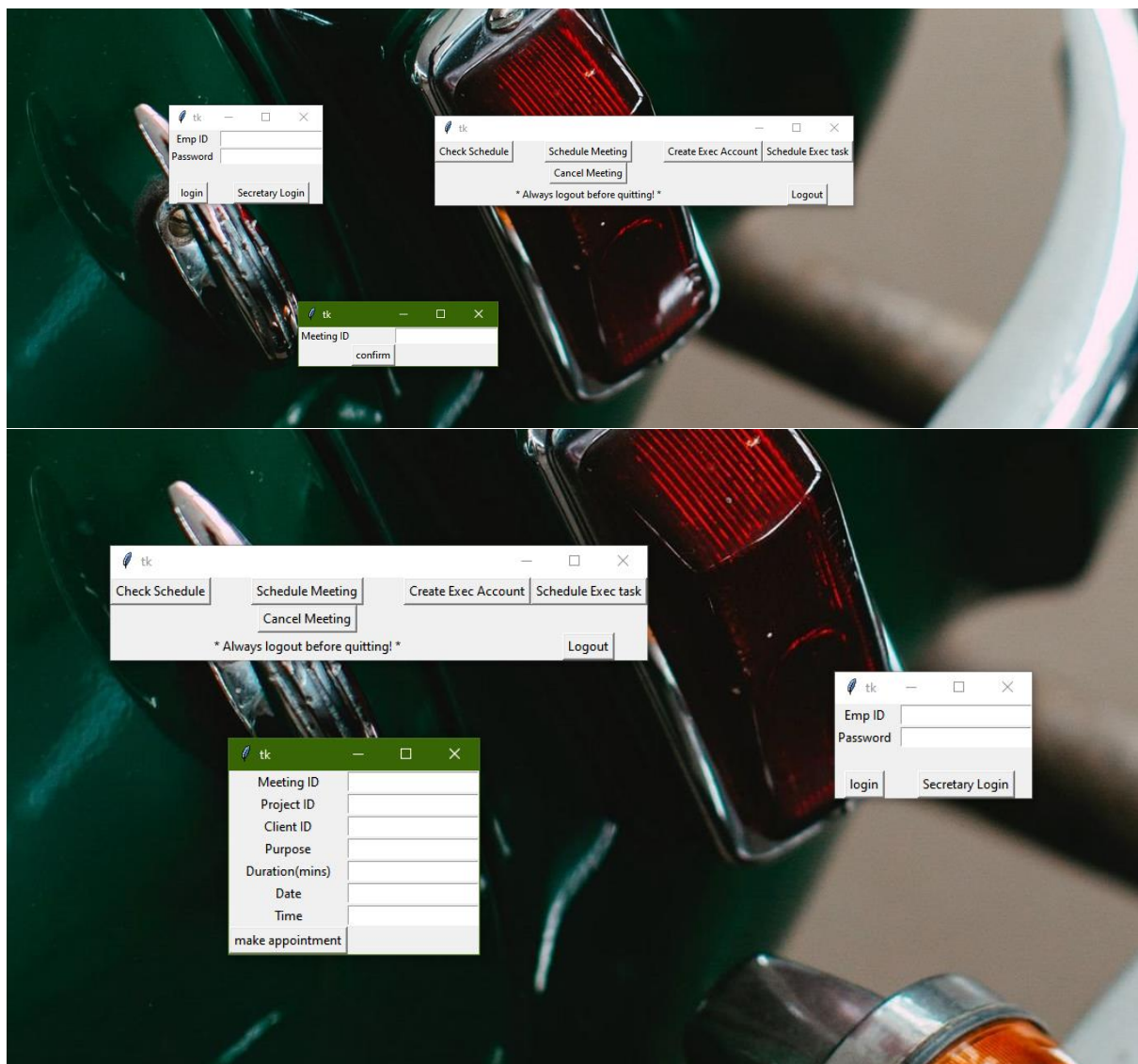


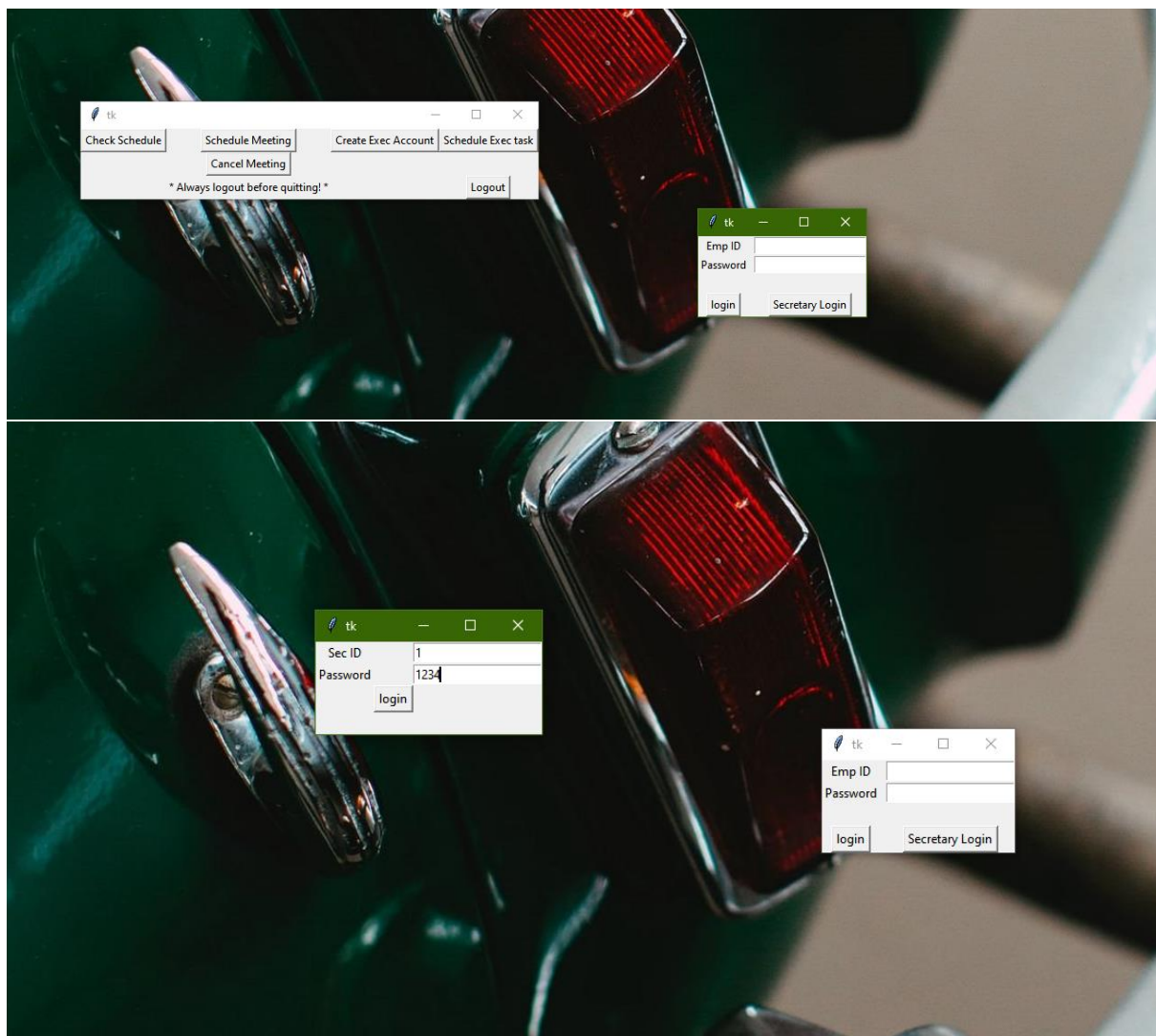


(secretary)









Limit to 1000 rows

```

1 • use db2;
2 • desc meeting;

```

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	Field	Type	Null	Key	Default	Extra
▶	m_id	decimal(6,0)	NO	PRI	NULL	
	purpose	varchar(50)	YES		NULL	
	duration	varchar(7)	YES		NULL	
	e_id	decimal(5,0)	YES	MUL	NULL	
	m_date	date	YES		NULL	
	m_time	time	YES		NULL	
	p_id	decimal(5,0)	YES	MUL	NULL	
	c_id	decimal(5,0)	YES	MUL	NULL	

Limit to 1000 rows

```

1 • use db2;
2 • desc employee;

```

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	Field	Type	Null	Key	Default	Extra
▶	e_id	decimal(5,0)	NO	PRI	NULL	
	e_password	varchar(20)	YES		NULL	
	f_name	varchar(10)	YES		NULL	
	l_name	varchar(10)	YES		NULL	
	salary	decimal(10,0)	YES		NULL	
	email	varchar(40)	YES		NULL	
	phone	decimal(10,0)	YES		NULL	
	address	varchar(50)	YES		NULL	
	department	varchar(20)	YES	MUL	NULL	
	designation	varchar(25)	YES		NULL	

Limit to 1000 rows

```
1 • use db2;  
2 • desc schedule1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

Field	Type	Null	Key	Default	Extra
s_date	date	YES		NULL	
s_time	time	YES		NULL	
e_id	decimal(5,0)	YES		NULL	

Limit to 1000 rows

```
1 • use db2;  
2 • desc Client1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

Field	Type	Null	Key	Default	Extra
c_id	decimal(5,0)	NO	PRI	NULL	
c_name	varchar(30)	YES		NULL	
c_phone	decimal(10,0)	YES		NULL	
c_email	varchar(40)	YES		NULL	
c_address	varchar(50)	YES		NULL	