MODULE CODE & MODULE TITLE

STW121COM – Introduction to Computing

ASSESSMENT TYPE

Individual Coursework

YEAR AND SEMESTER

2019-2020 SEPTEMBER/NOVEMBER

STUDENT NAME: SWAPNIL THAPA

COVENTRY ID: 10176782

STUDENT ID: 190105

ASSIGNMENT DUE DATE: 21st February, 2020

ASSIGNMENT SUBMISSION DATE: 21st February, 2020

Table of Content

List of Figures:

1. Introduction

The project is about to make a GUI (Graphical User Interface) based python program for Employee Management System which consist of login form, user registration form, selection form, employee form and department form where a user has to enter the data in GUI interfaces that is created with the help of python code. Registered user can view the employee details with the help of view button present in employee registration form. The GUI module Tkinter which is built-in module in python have been implemented in the code. User entered data are stored in a file which is available in the folder where the program has been saved.

(tutorialspoint, 2020)

2.Screenshots of User Interface and Code

User Login Form Interface

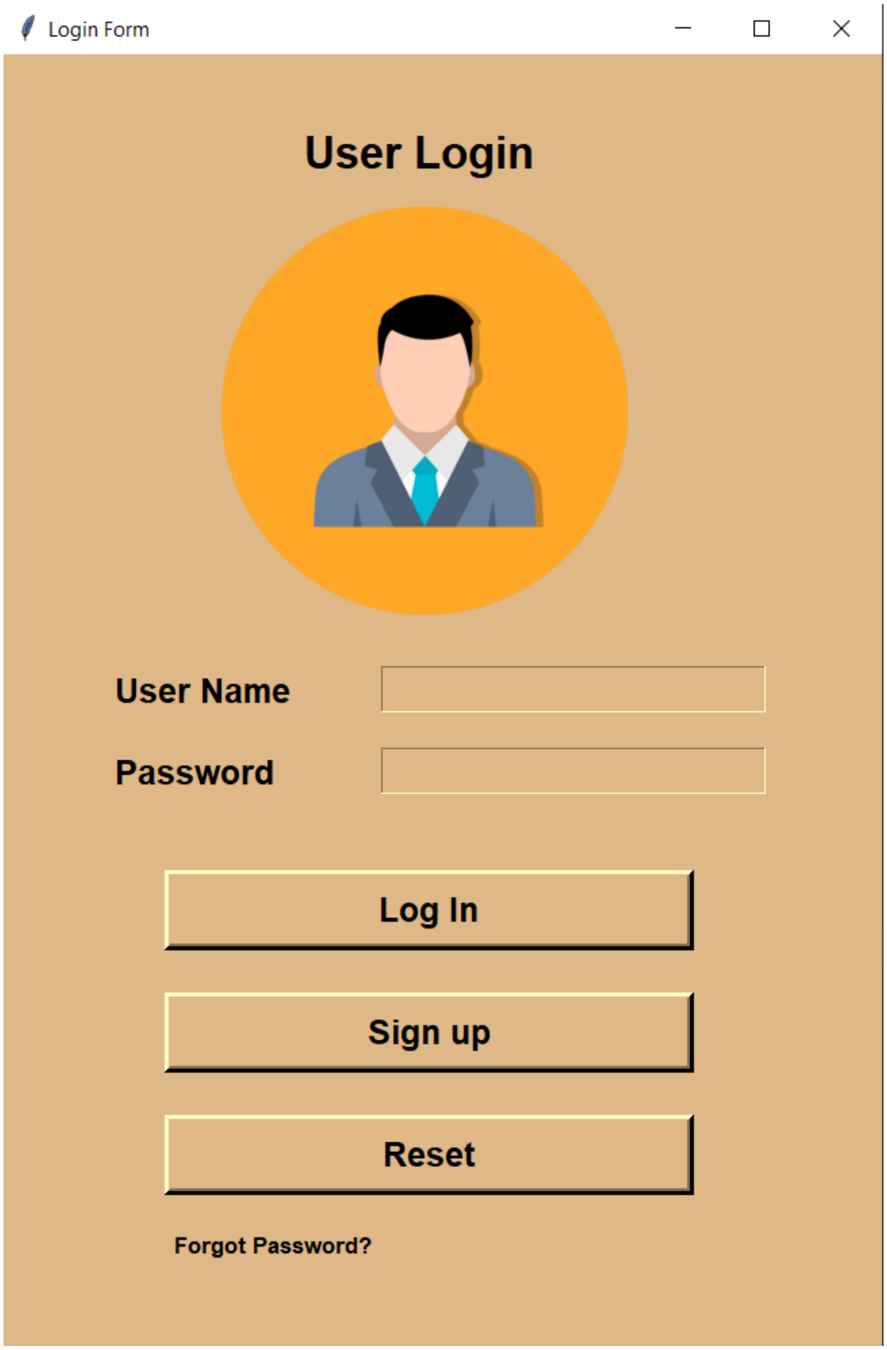
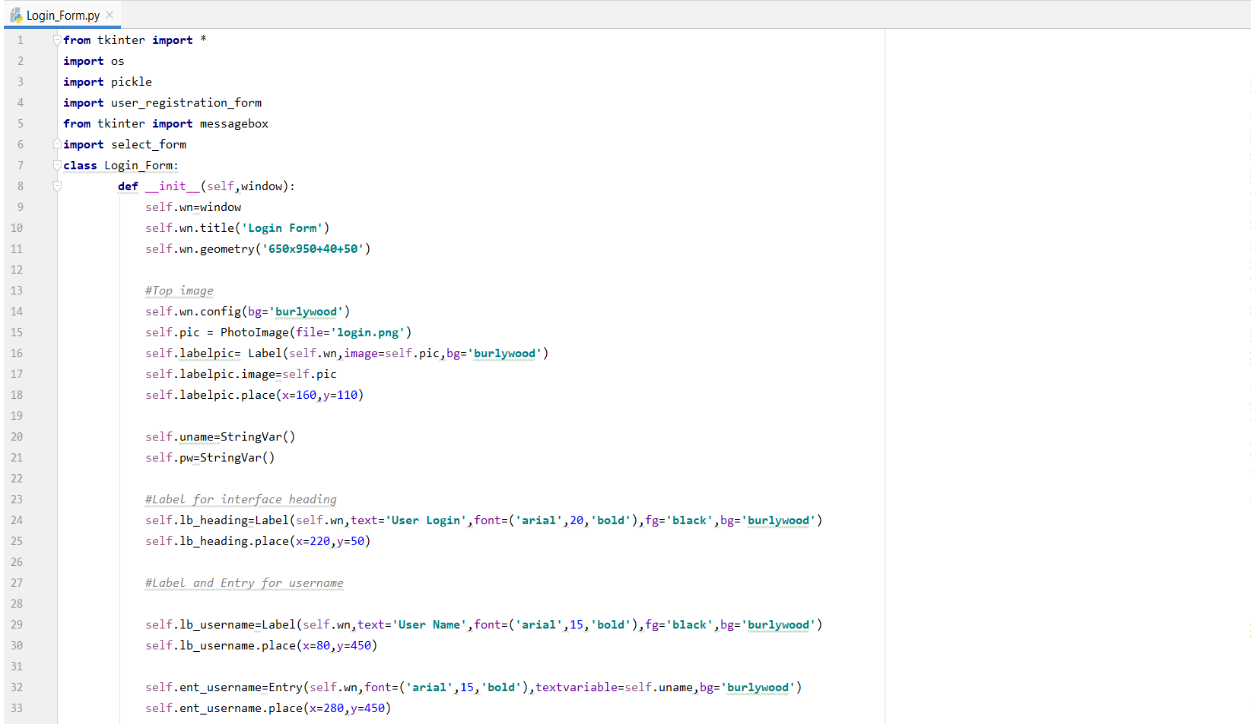
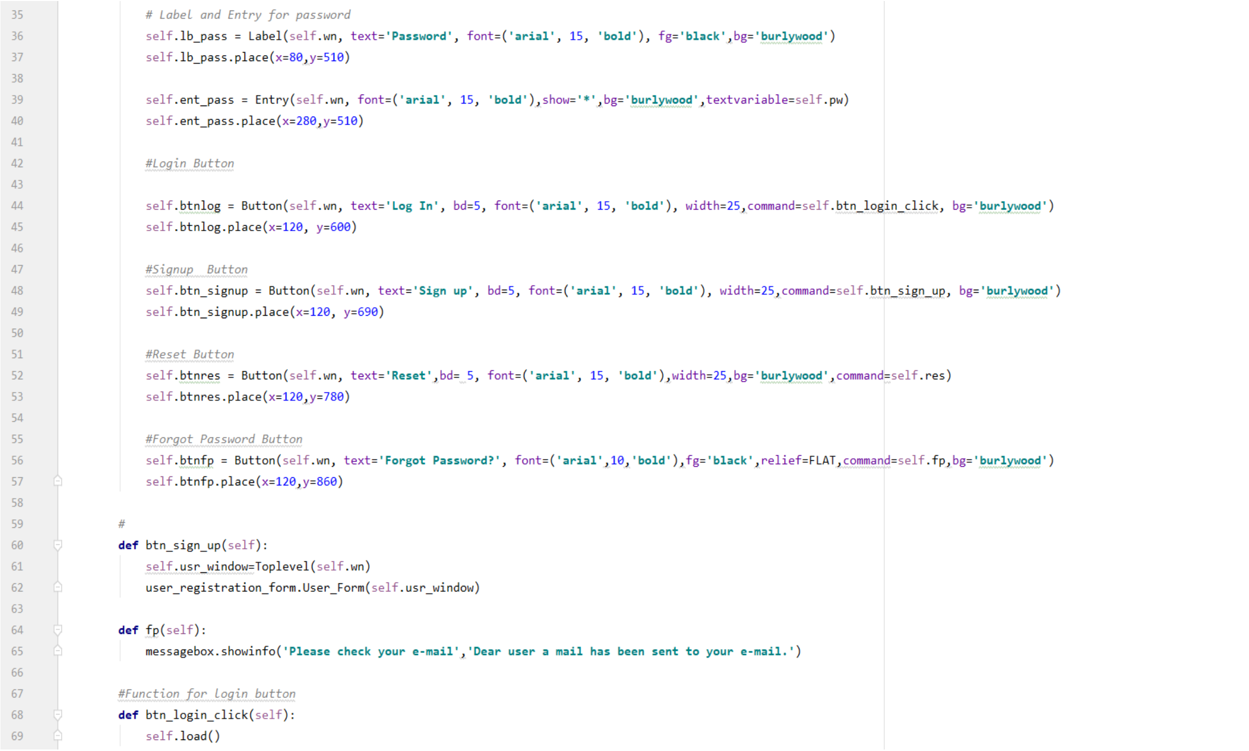


Figure 1: User Login Interface

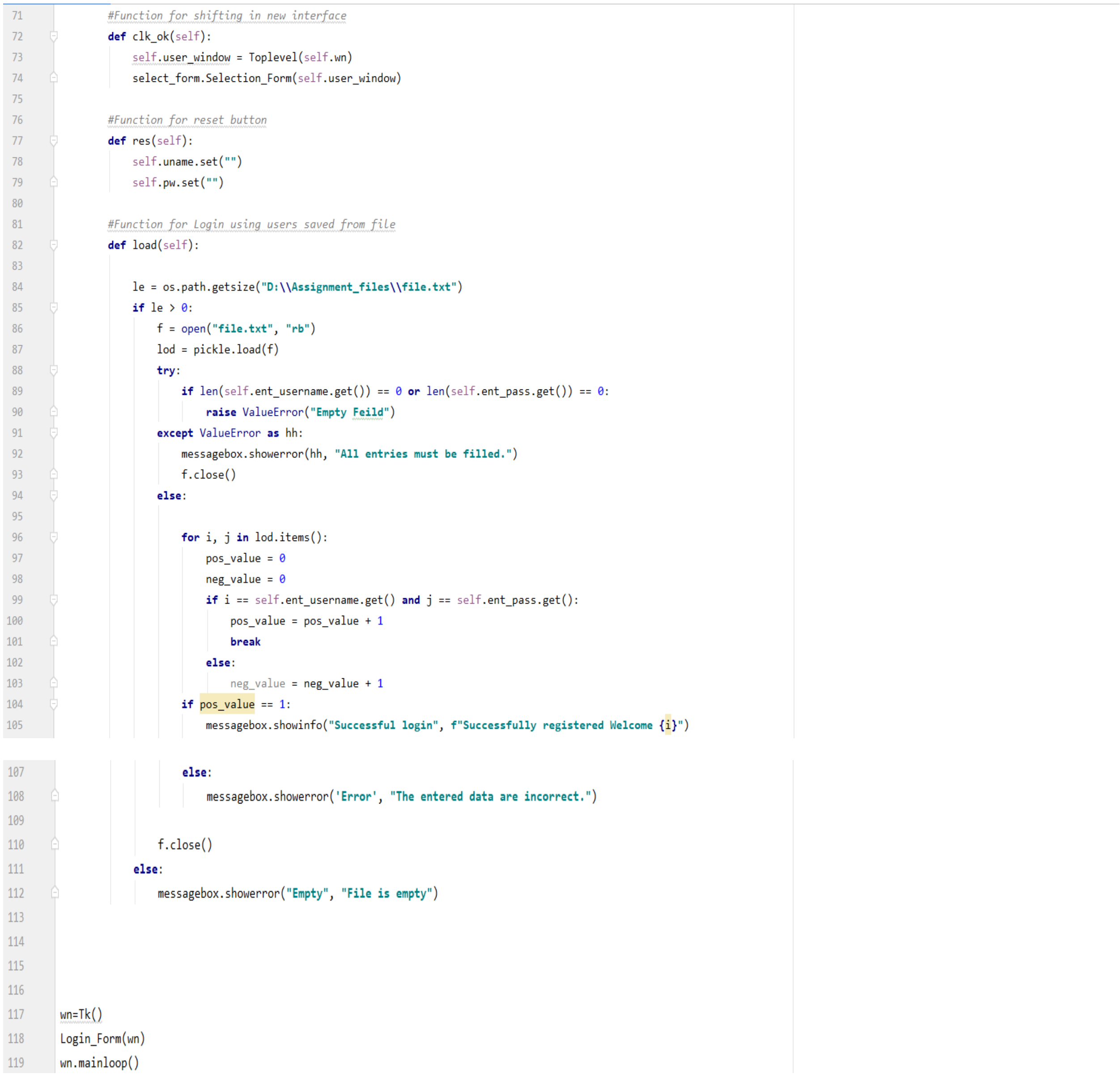
Login form interface code



```
from tkinter import *
import os
import pickle
import user_registration_form
from tkinter import messagebox
import select_form
class Login_Form:
        def __init__(self,window):
            self.wn=window
            self.wn.title('Login Form')
            self.wn.geometry('650x950+40+50')

            #Top image
            self.wn.config(bg='burlywood')
            self.pic = PhotoImage(file='login.png')
            self.labelpic= Label(self.wn,image=self.pic,bg='burlywood')
            self.labelpic.image=self.pic
            self.labelpic.place(x=160,y=110)

            self.uname=StringVar()
            self.pw=StringVar()

            #Label for interface heading
            self.lb_heading=Label(self.wn,text='User Login',font=('arial',20,'bold'),fg='black',bg='burlywood')
            self.lb_heading.place(x=220,y=50)

            #Label and Entry for username

            self.lb_username=Label(self.wn,text='User Name',font=('arial',15,'bold'),fg='black',bg='burlywood')
            self.lb_username.place(x=80,y=450)

            self.ent_username=Entry(self.wn,font=('arial',15,'bold'),textvariable=self.uname,bg='burlywood')
            self.ent_username.place(x=280,y=450)
```



```
            # Label and Entry for password
            self.lb_pass = Label(self.wn, text='Password', font=('arial', 15, 'bold'), fg='black',bg='burlywood')
            self.lb_pass.place(x=80,y=510)

            self.ent_pass = Entry(self.wn, font=('arial', 15, 'bold'),show='*',bg='burlywood',textvariable=self.pw)
            self.ent_pass.place(x=280,y=510)

            #Login Button

            self.btnlog = Button(self.wn, text='Log In', bd=5, font=('arial', 15, 'bold'), width=25,command=self.btn_login_click, bg='burlywood')
            self.btnlog.place(x=120, y=600)

            #Signup  Button
            self.btn_signup = Button(self.wn, text='Sign up', bd=5, font=('arial', 15, 'bold'), width=25,command=self.btn_sign_up, bg='burlywood')
            self.btn_signup.place(x=120, y=690)

            #Reset Button
            self.btnres = Button(self.wn, text='Reset',bd= 5, font=('arial', 15, 'bold'),width=25,bg='burlywood',command=self.res)
            self.btnres.place(x=120,y=780)

            #Forgot Password Button
            self.btnfp = Button(self.wn, text='Forgot Password?', font=('arial',10,'bold'),fg='black',relief=FLAT,command=self.fp,bg='burlywood')
            self.btnfp.place(x=120,y=860)

        #
        def btn_sign_up(self):
            self.usr_window=Toplevel(self.wn)
            user_registration_form.User_Form(self.usr_window)

        def fp(self):
            messagebox.showinfo('Please check your e-mail','Dear user a mail has been sent to your e-mail.')

        #Function for login button
        def btn_login_click(self):
            self.load()
```

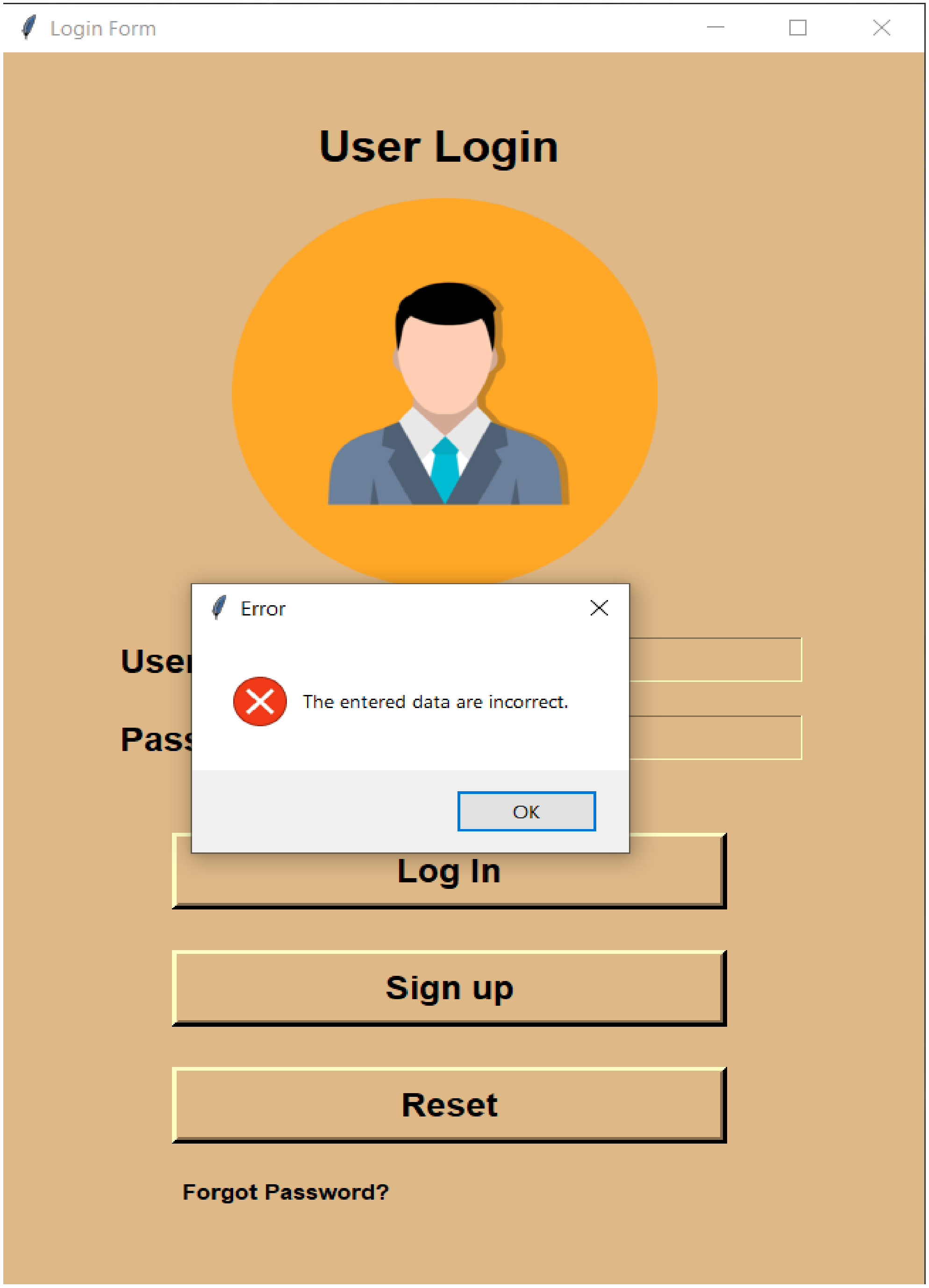


```
    #Function for shifting in new interface
    def clk_ok(self):
        self.user_window = Toplevel(self.wn)
        select_form.Selection_Form(self.user_window)

    #Function for reset button
    def res(self):
        self.uname.set("")
        self.pw.set("")

    #Function for Login using users saved from file
    def load(self):

        le = os.path.getsize("D:\\Assignment_files\\file.txt")
        if le > 0:
            f = open("file.txt", "rb")
            lod = pickle.load(f)
            try:
                if len(self.ent_username.get()) == 0 or len(self.ent_pass.get()) == 0:
                    raise ValueError("Empty Feild")
            except ValueError as hh:
                messagebox.showerror(hh, "All entries must be filled.")
                f.close()
            else:

                for i, j in lod.items():
                    pos_value = 0
                    neg_value = 0
                    if i == self.ent_username.get() and j == self.ent_pass.get():
                        pos_value = pos_value + 1
                        break
                    else:
                        neg_value = neg_value + 1
                if pos_value == 1:
                    messagebox.showinfo("Successful login", f"Successfully registered Welcome {i}")
```

```
                else:
                    messagebox.showerror('Error', "The entered data are incorrect.")

            f.close()
        else:
            messagebox.showerror("Empty", "File is empty")




wn=Tk()
Login_Form(wn)
wn.mainloop()
```

Figure 2: Code for User Login interface

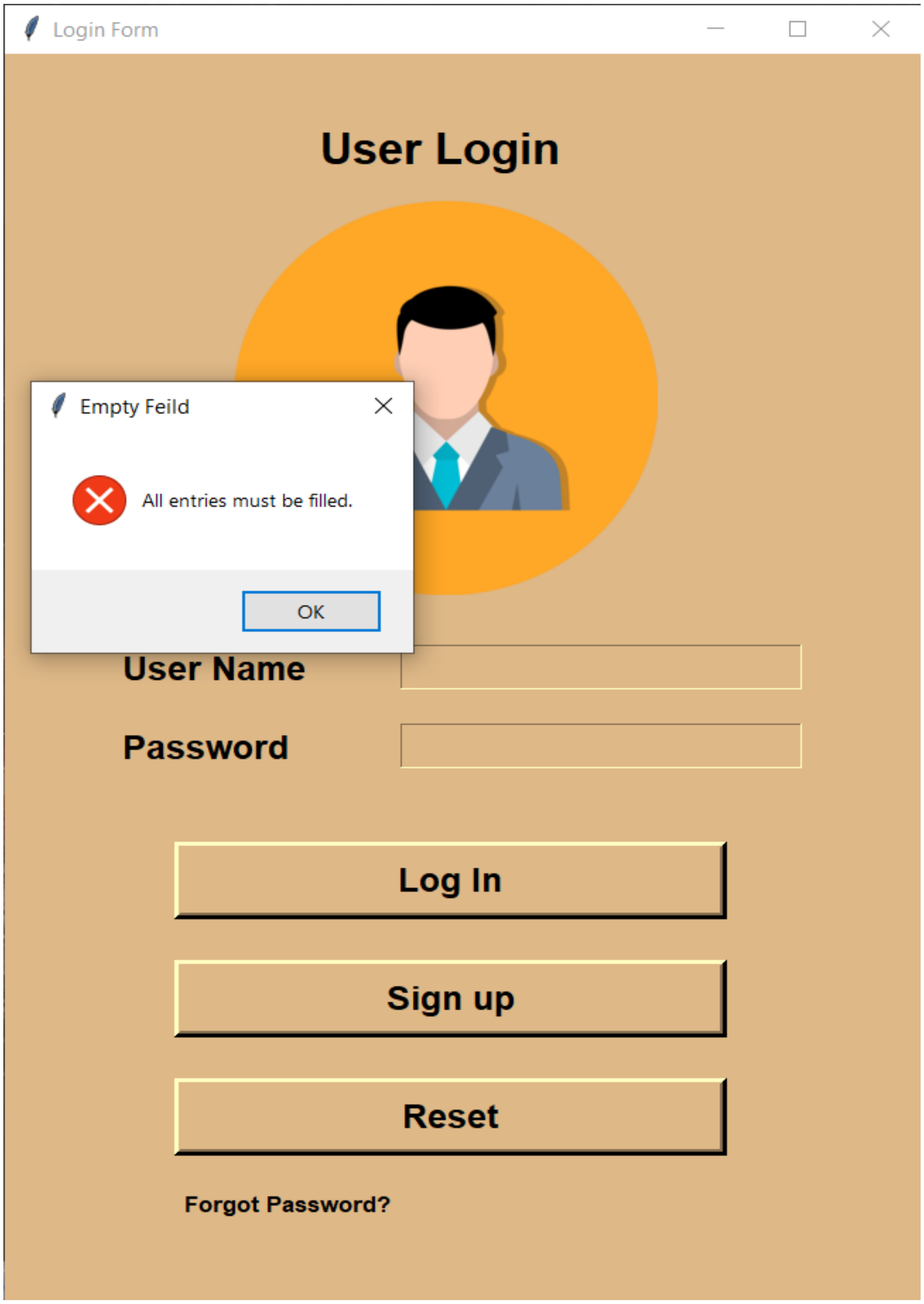
After Login button is clicked by wrong Username and Password



The entered data are incorrect.

Figure 3: Login button clicked filling wrong username and password

After Login button clicked filling no any entries

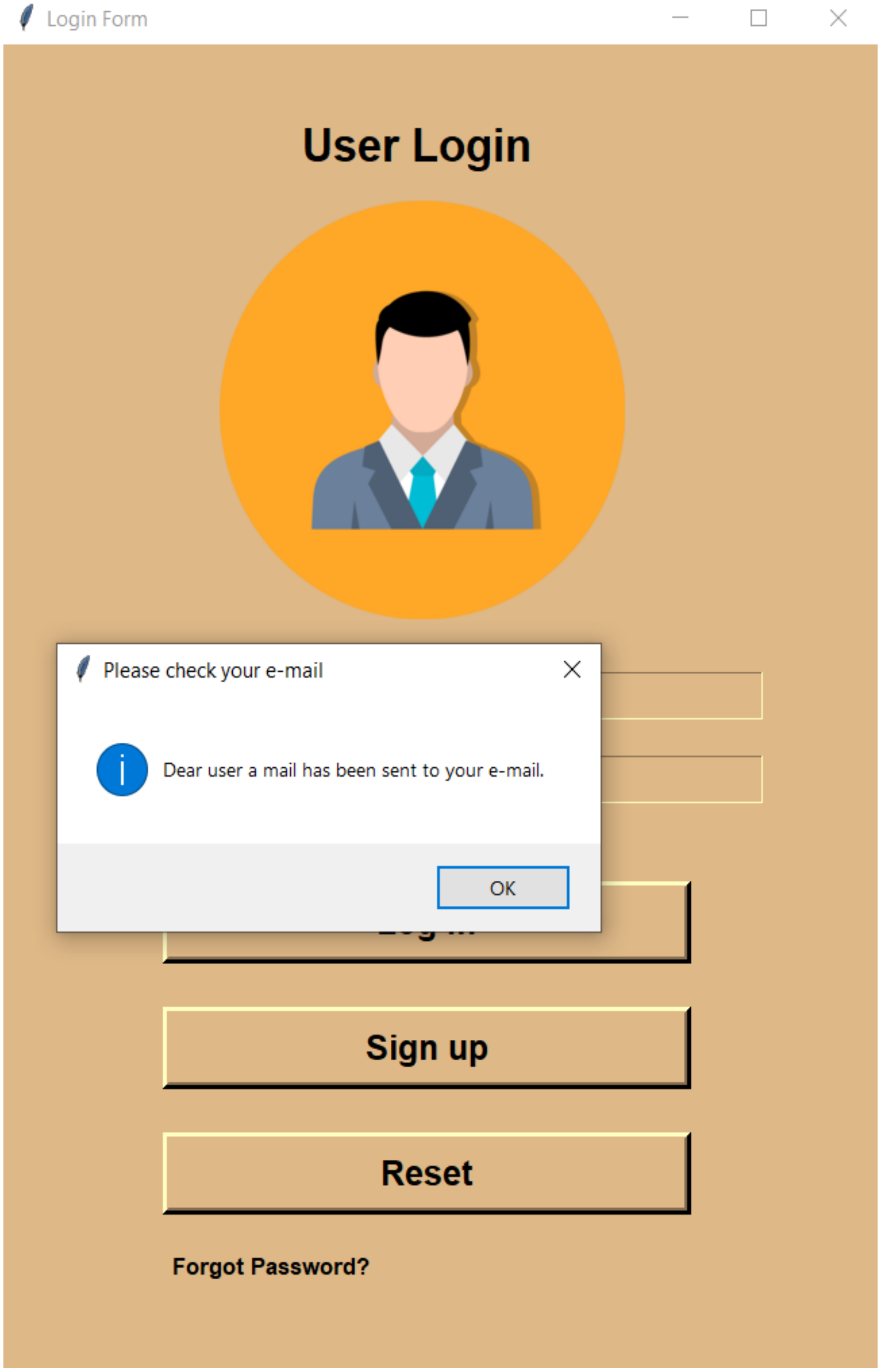


Empty Feild

All entries must be filled.

Figure 4: Login button clicked without entries

After Forgot password button is clicked

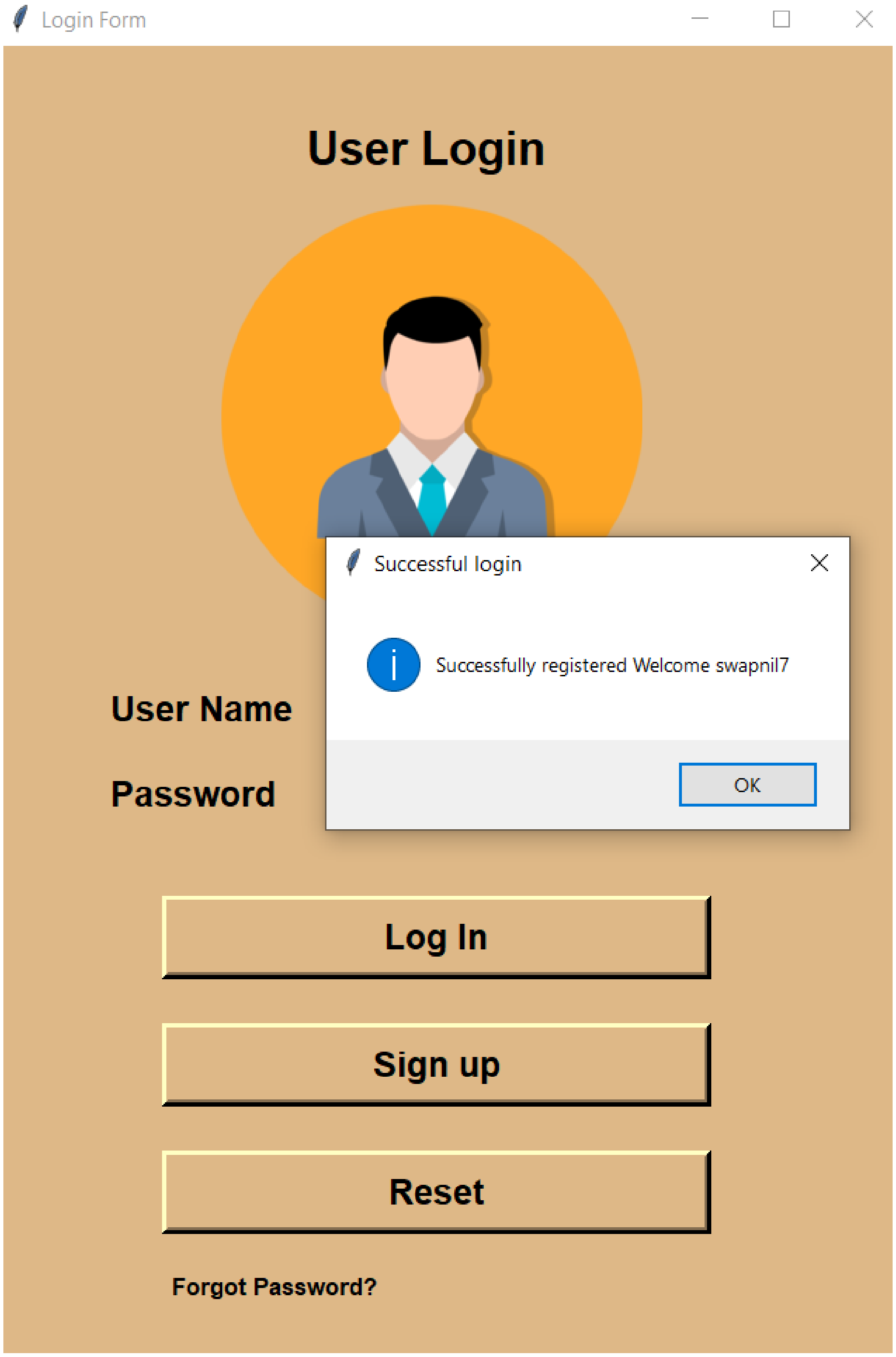


Please check your e-mail

Dear user a mail has been sent to your e-mail.

Figure 5: Forgot password button clicked

After clicking register button with registered username and password



Successfully registered Welcome swapnil7

Figure 6: Register button clicked with registered username and password

User Registration Form Interface

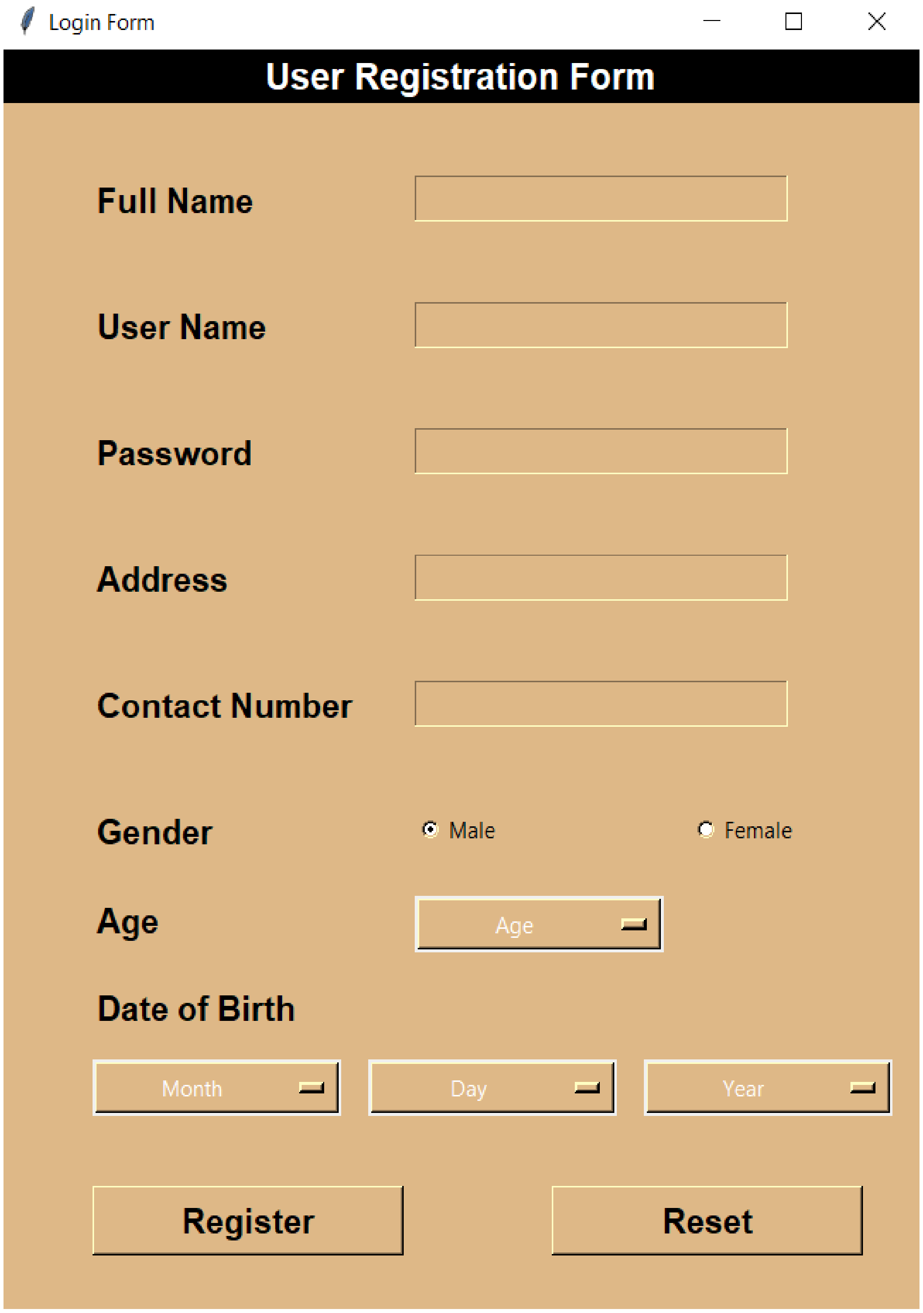
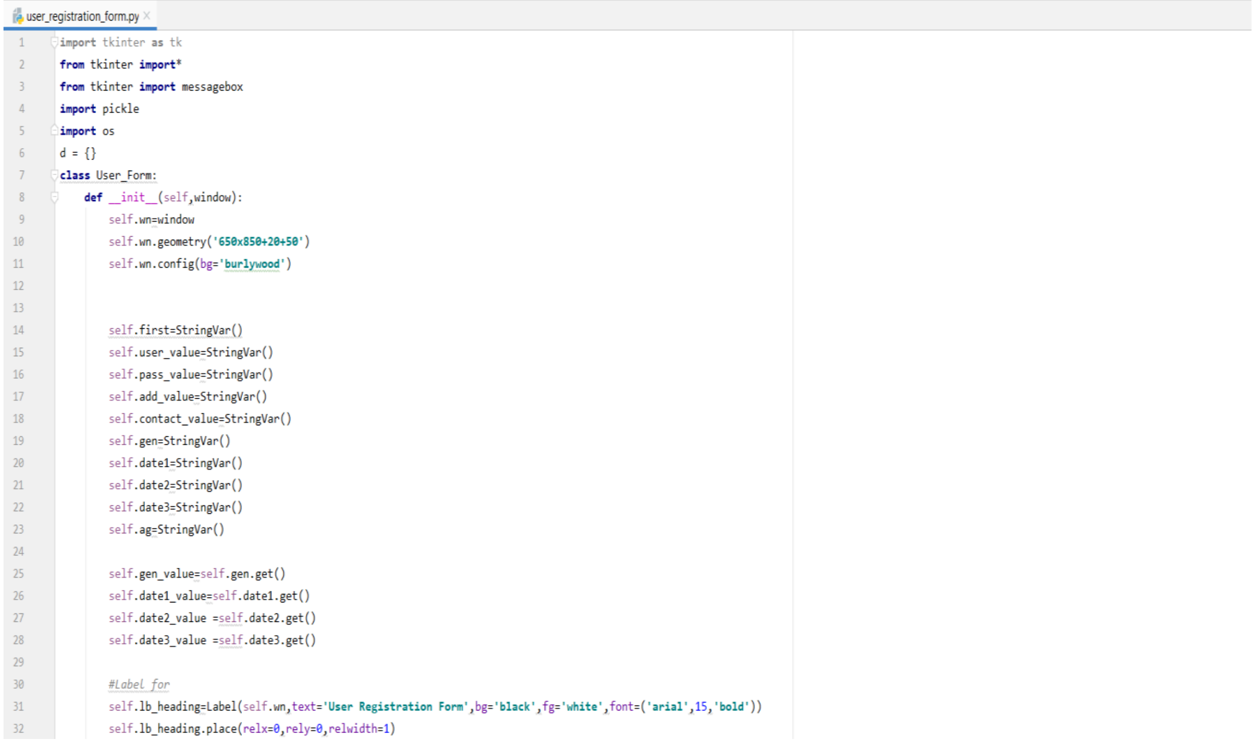


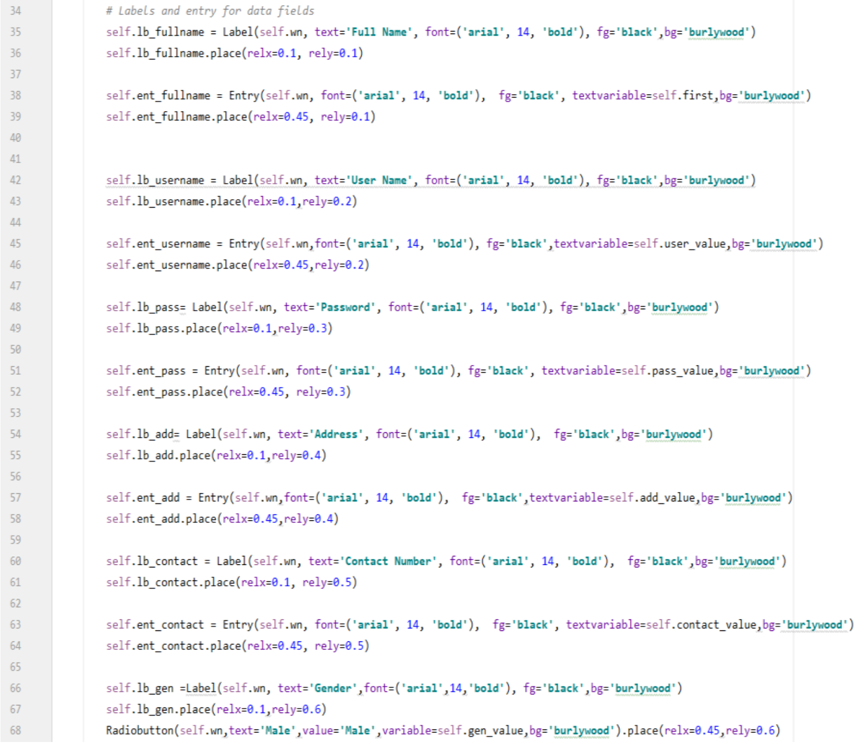
Figure 7:User registration interface

User registration interface code

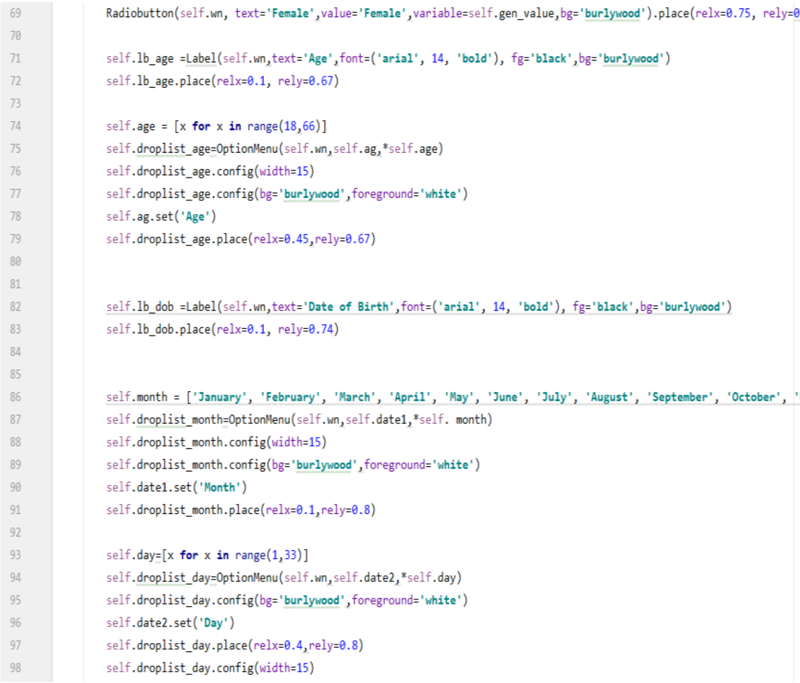


user_registration_form.py

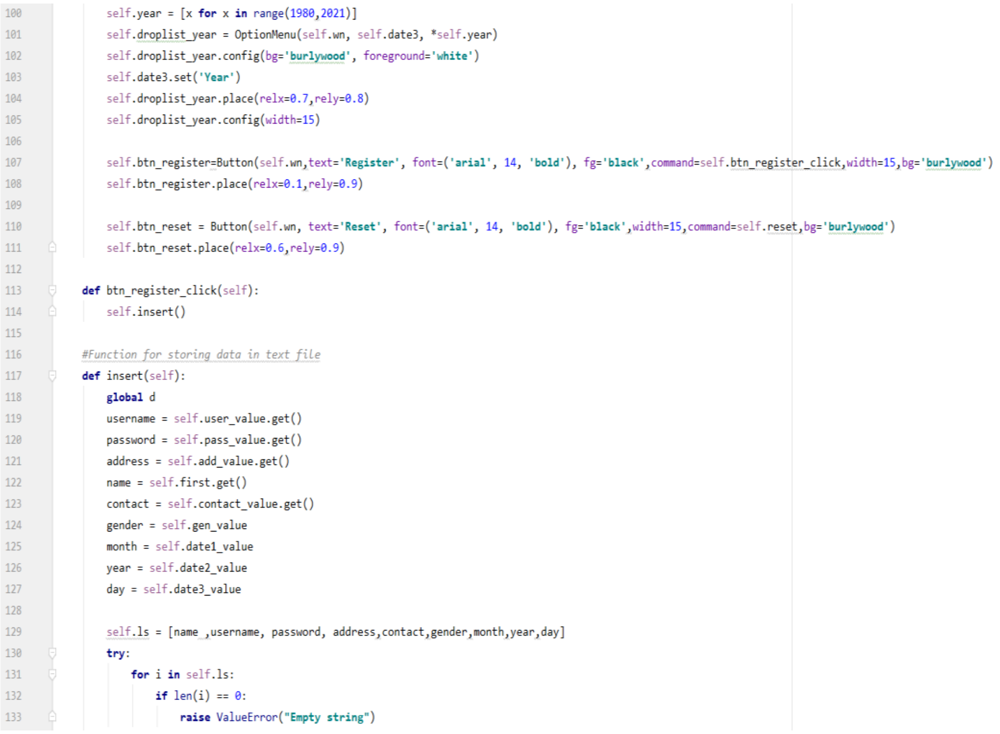
```
import tkinter as tk
from tkinter import*
from tkinter import messagebox
import pickle
import os
d = {}
class User_Form:
    def __init__(self,window):
        self.wn=window
        self.wn.geometry('650x850+20+50')
        self.wn.config(bg='burlywood')


        self.first=StringVar()
        self.user_value=StringVar()
        self.pass_value=StringVar()
        self.add_value=StringVar()
        self.contact_value=StringVar()
        self.gen=StringVar()
        self.date1=StringVar()
        self.date2=StringVar()
        self.date3=StringVar()
        self.ag=StringVar()

        self.gen_value=self.gen.get()
        self.date1_value=self.date1.get()
        self.date2_value =self.date2.get()
        self.date3_value =self.date3.get()

        #Label for
        self.lb_heading=Label(self.wn,text='User Registration Form',bg='black',fg='white',font=('arial',15,'bold'))
        self.lb_heading.place(relx=0,rely=0,relwidth=1)
```



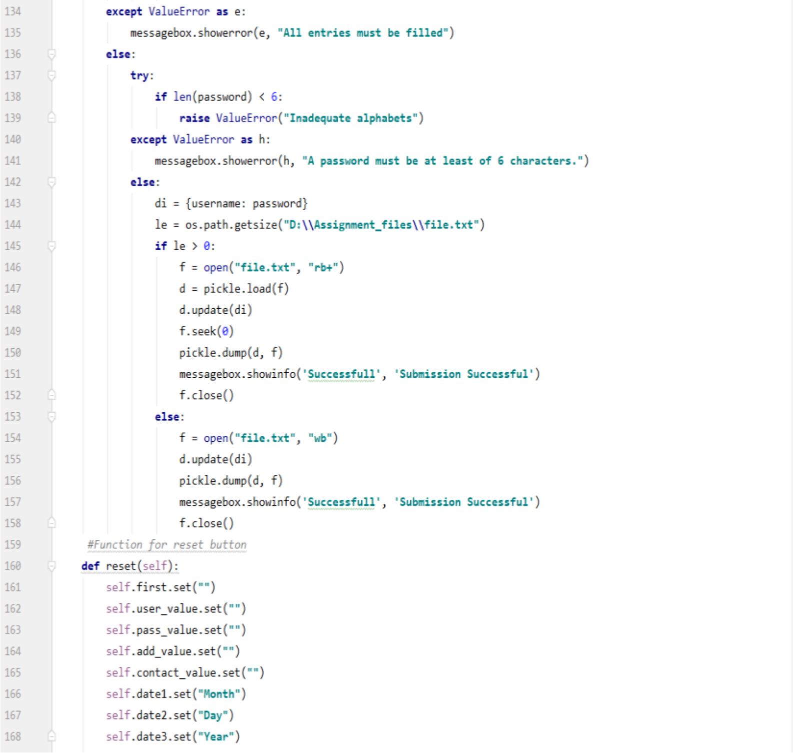
```
        # Labels and entry for data fields
        self.lb_fullname = Label(self.wn, text='Full Name', font=('arial', 14, 'bold'), fg='black',bg='burlywood')
        self.lb_fullname.place(relx=0.1, rely=0.1)

        self.ent_fullname = Entry(self.wn, font=('arial', 14, 'bold'),  fg='black', textvariable=self.first,bg='burlywood')
        self.ent_fullname.place(relx=0.45, rely=0.1)


        self.lb_username = Label(self.wn, text='User Name', font=('arial', 14, 'bold'), fg='black',bg='burlywood')
        self.lb_username.place(relx=0.1,rely=0.2)

        self.ent_username = Entry(self.wn,font=('arial', 14, 'bold'), fg='black',textvariable=self.user_value,bg='burlywood')
        self.ent_username.place(relx=0.45,rely=0.2)

        self.lb_pass= Label(self.wn, text='Password', font=('arial', 14, 'bold'), fg='black',bg='burlywood')
        self.lb_pass.place(relx=0.1,rely=0.3)

        self.ent_pass = Entry(self.wn, font=('arial', 14, 'bold'), fg='black', textvariable=self.pass_value,bg='burlywood')
        self.ent_pass.place(relx=0.45, rely=0.3)

        self.lb_add= Label(self.wn, text='Address', font=('arial', 14, 'bold'),  fg='black',bg='burlywood')
        self.lb_add.place(relx=0.1,rely=0.4)

        self.ent_add = Entry(self.wn,font=('arial', 14, 'bold'),  fg='black',textvariable=self.add_value,bg='burlywood')
        self.ent_add.place(relx=0.45,rely=0.4)

        self.lb_contact = Label(self.wn, text='Contact Number', font=('arial', 14, 'bold'),  fg='black',bg='burlywood')
        self.lb_contact.place(relx=0.1, rely=0.5)

        self.ent_contact = Entry(self.wn, font=('arial', 14, 'bold'),  fg='black', textvariable=self.contact_value,bg='burlywood')
        self.ent_contact.place(relx=0.45, rely=0.5)

        self.lb_gen =Label(self.wn, text='Gender',font=('arial',14,'bold'), fg='black',bg='burlywood')
        self.lb_gen.place(relx=0.1,rely=0.6)
        Radiobutton(self.wn,text='Male',value='Male',variable=self.gen_value,bg='burlywood').place(relx=0.45,rely=0.6)
```



```
        Radiobutton(self.wn, text='Female',value='Female',variable=self.gen_value,bg='burlywood').place(relx=0.75, rely=0.6)

        self.lb_age =Label(self.wn,text='Age',font=('arial', 14, 'bold'), fg='black',bg='burlywood')
        self.lb_age.place(relx=0.1, rely=0.67)

        self.age = [x for x in range(18,66)]
        self.droplist_age=OptionMenu(self.wn,self.ag,*self.age)
        self.droplist_age.config(width=15)
        self.droplist_age.config(bg='burlywood',foreground='white')
        self.ag.set('Age')
        self.droplist_age.place(relx=0.45,rely=0.67)


        self.lb_dob =Label(self.wn,text='Date of Birth',font=('arial', 14, 'bold'), fg='black',bg='burlywood')
        self.lb_dob.place(relx=0.1, rely=0.74)


        self.month = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']
        self.droplist_month=OptionMenu(self.wn,self.date1,*self. month)
        self.droplist_month.config(width=15)
        self.droplist_month.config(bg='burlywood',foreground='white')
        self.date1.set('Month')
        self.droplist_month.place(relx=0.1,rely=0.8)

        self.day=[x for x in range(1,33)]
        self.droplist_day=OptionMenu(self.wn,self.date2,*self.day)
        self.droplist_day.config(bg='burlywood',foreground='white')
        self.date2.set('Day')
        self.droplist_day.place(relx=0.4,rely=0.8)
        self.droplist_day.config(width=15)
```



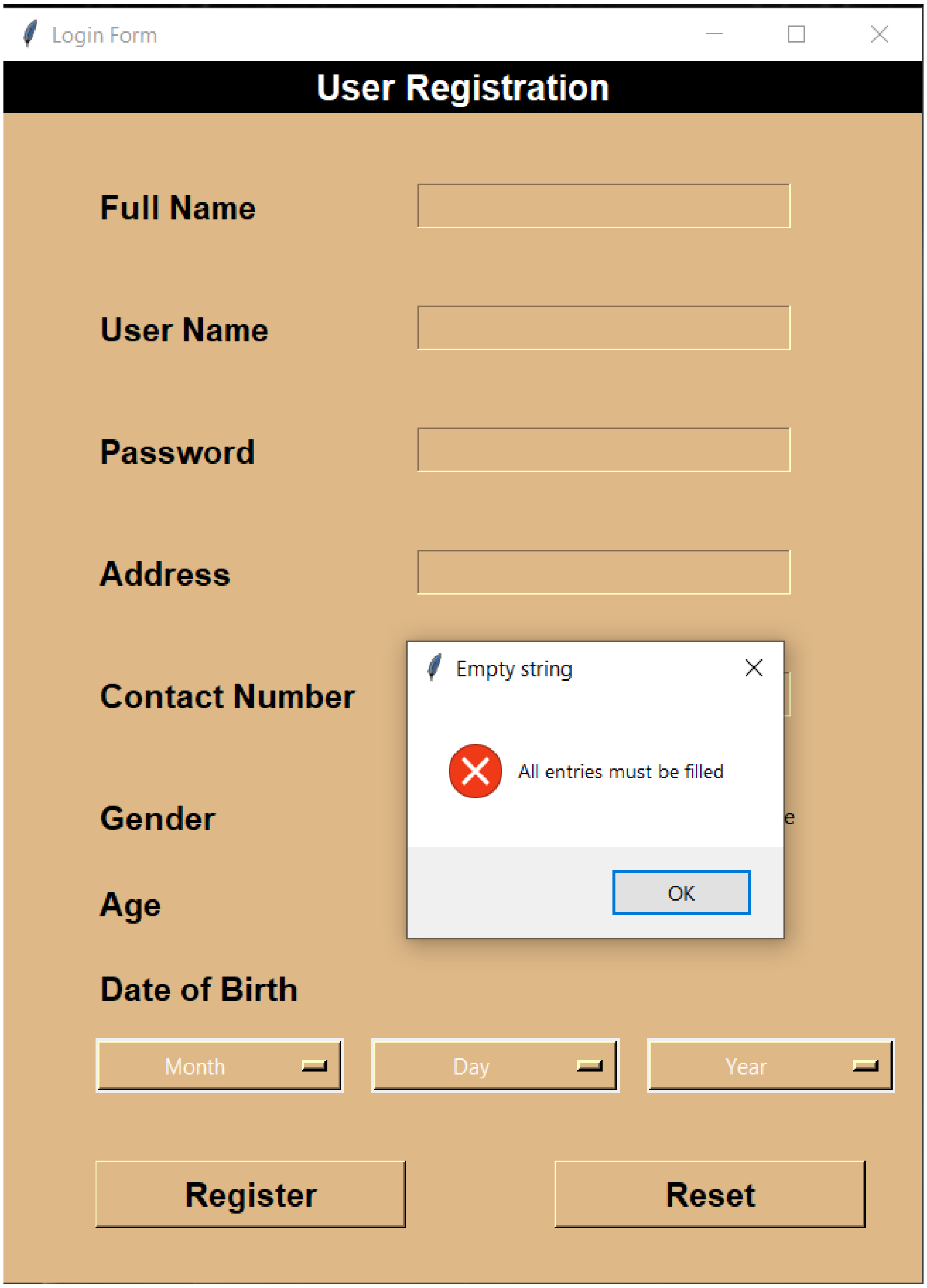
```
        self.year = [x for x in range(1980,2021)]
        self.droplist_year = OptionMenu(self.wn, self.date3, *self.year)
        self.droplist_year.config(bg='burlywood', foreground='white')
        self.date3.set('Year')
        self.droplist_year.place(relx=0.7,rely=0.8)
        self.droplist_year.config(width=15)

        self.btn_register=Button(self.wn,text='Register', font=('arial', 14, 'bold'), fg='black',command=self.btn_register_click,width=15,bg='burlywood')
        self.btn_register.place(relx=0.1,rely=0.9)

        self.btn_reset = Button(self.wn, text='Reset', font=('arial', 14, 'bold'), fg='black',width=15,command=self.reset,bg='burlywood')
        self.btn_reset.place(relx=0.6,rely=0.9)

    def btn_register_click(self):
        self.insert()

    #Function for storing data in text file
    def insert(self):
        global d
        username = self.user_value.get()
        password = self.pass_value.get()
        address = self.add_value.get()
        name = self.first.get()
        contact = self.contact_value.get()
        gender = self.gen_value
        month = self.date1_value
        year = self.date2_value
        day = self.date3_value

        self.ls = [name ,username, password, address,contact,gender,month,year,day]
        try:
            for i in self.ls:
                if len(i) == 0:
                    raise ValueError("Empty string")
```



```
        except ValueError as e:
            messagebox.showerror(e, "All entries must be filled")
        else:
            try:
                if len(password) < 6:
                    raise ValueError("Inadequate alphabets")
            except ValueError as h:
                messagebox.showerror(h, "A password must be at least of 6 characters.")
            else:
                di = {username: password}
                le = os.path.getsize("D:\\Assignment_files\\file.txt")
                if le > 0:
                    f = open("file.txt", "rb+")
                    d = pickle.load(f)
                    d.update(di)
                    f.seek(0)
                    pickle.dump(d, f)
                    messagebox.showinfo('Successfull', 'Submission Successful')
                    f.close()
                else:
                    f = open("file.txt", "wb")
                    d.update(di)
                    pickle.dump(d, f)
                    messagebox.showinfo('Successfull', 'Submission Successful')
                    f.close()
     #Function for reset button
    def reset(self):
        self.first.set("")
        self.user_value.set("")
        self.pass_value.set("")
        self.add_value.set("")
        self.contact_value.set("")
        self.date1.set("Month")
        self.date2.set("Day")
        self.date3.set("Year")
```

Figure 8: Code for user signup interface

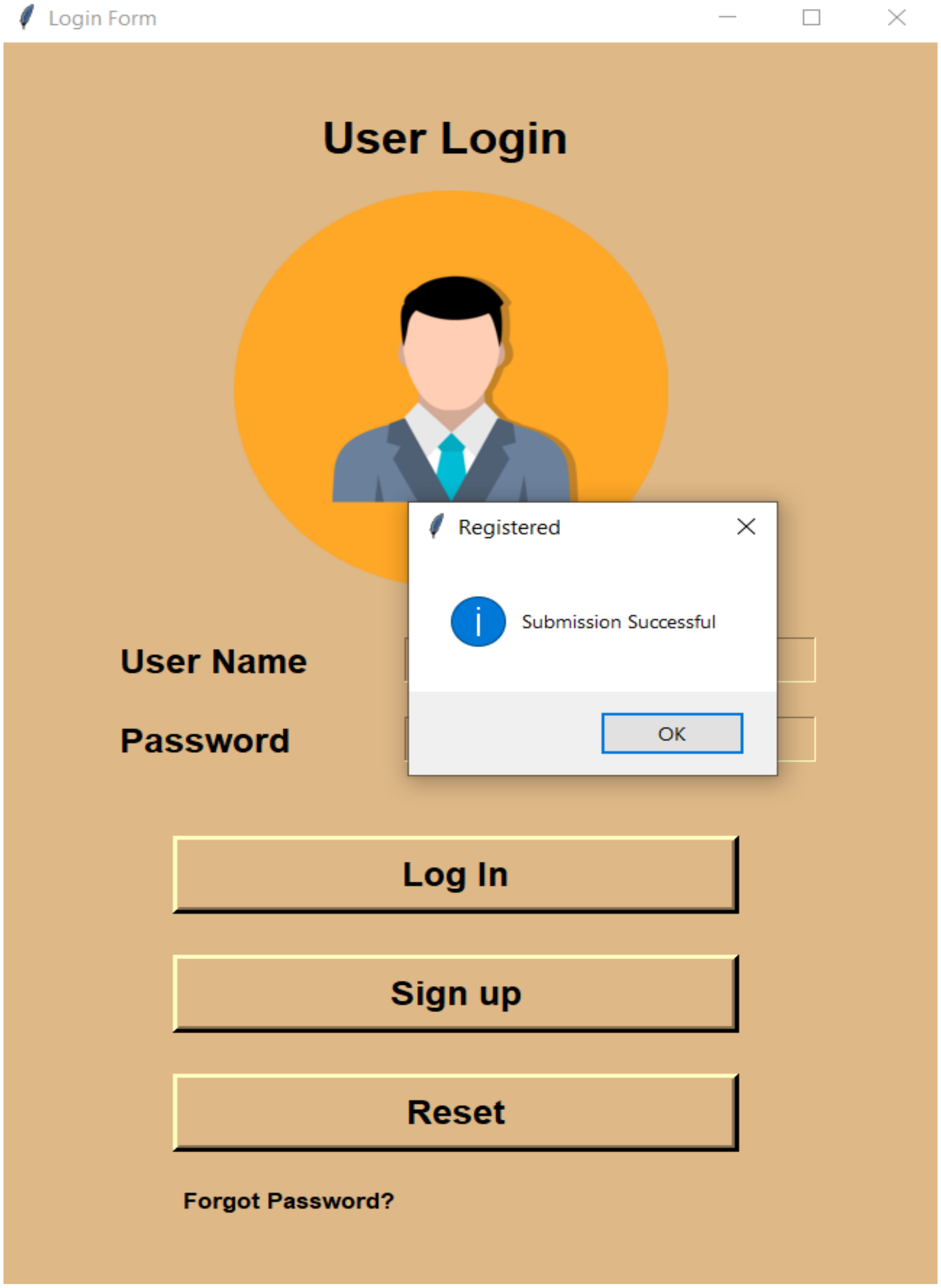
After pressing register button without any entries



All entries must be filled

Figure 9: Clicked register button without entries

After register button clicked filling all entries



Submission Successful

Figure 10: Login interface opened after clicking register

Selection form interface

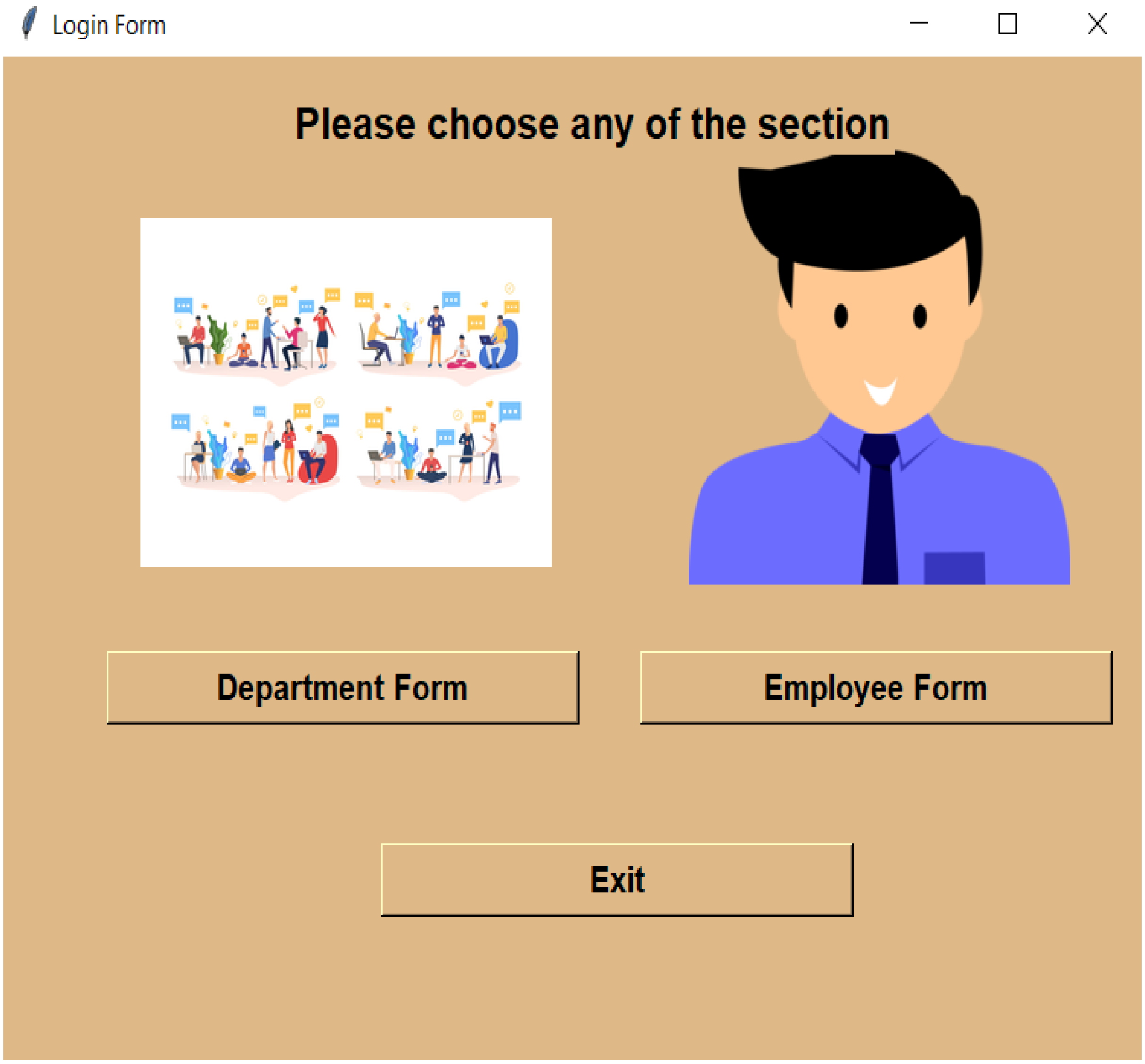
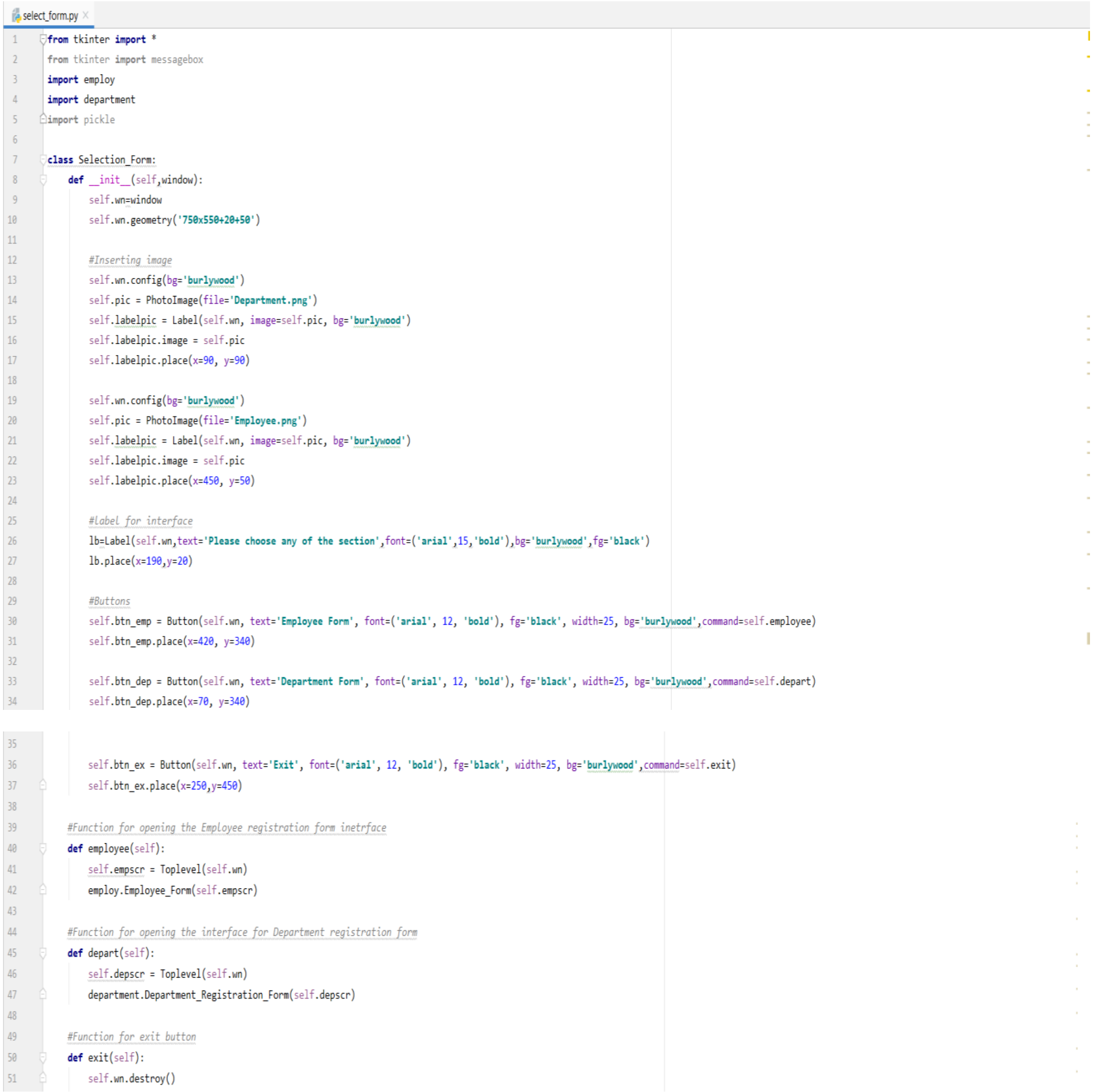


Figure 11:Selection form interface window

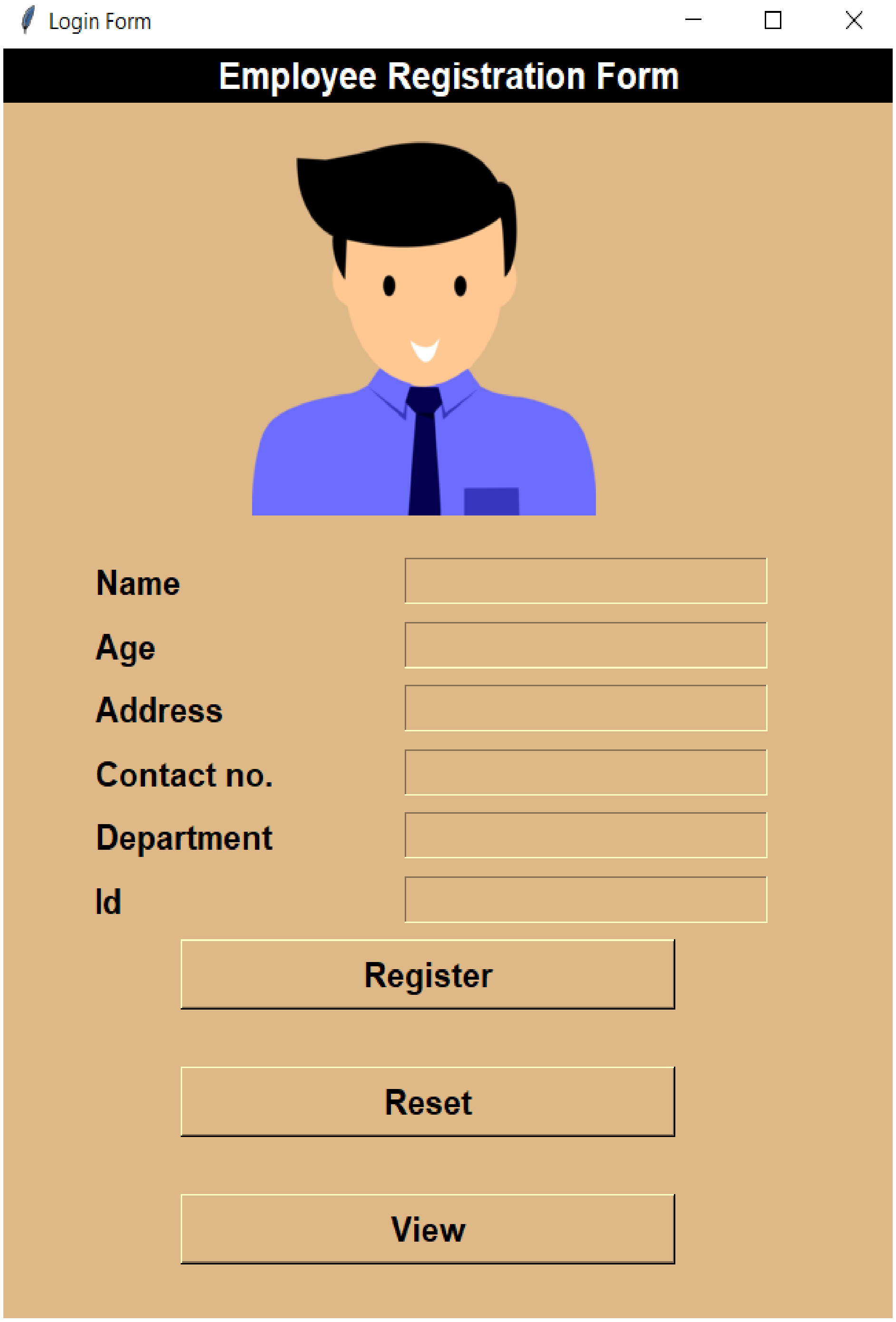
Selection form interface code



```python
from tkinter import *
from tkinter import messagebox
import employ
import department
import pickle

class Selection_Form:
    def __init__(self,window):
        self.wn=window
        self.wn.geometry('750x550+20+50')

        #Inserting image
        self.wn.config(bg='burlywood')
        self.pic = PhotoImage(file='Department.png')
        self.labelpic = Label(self.wn, image=self.pic, bg='burlywood')
        self.labelpic.image = self.pic
        self.labelpic.place(x=90, y=90)

        self.wn.config(bg='burlywood')
        self.pic = PhotoImage(file='Employee.png')
        self.labelpic = Label(self.wn, image=self.pic, bg='burlywood')
        self.labelpic.image = self.pic
        self.labelpic.place(x=450, y=50)

        #label for interface
        lb=Label(self.wn,text='Please choose any of the section',font=('arial',15,'bold'),bg='burlywood',fg='black')
        lb.place(x=190,y=20)

        #Buttons
        self.btn_emp = Button(self.wn, text='Employee Form', font=('arial', 12, 'bold'), fg='black', width=25, bg='burlywood',command=self.employee)
        self.btn_emp.place(x=420, y=340)

        self.btn_dep = Button(self.wn, text='Department Form', font=('arial', 12, 'bold'), fg='black', width=25, bg='burlywood',command=self.depart)
        self.btn_dep.place(x=70, y=340)


        self.btn_ex = Button(self.wn, text='Exit', font=('arial', 12, 'bold'), fg='black', width=25, bg='burlywood',command=self.exit)
        self.btn_ex.place(x=250,y=450)

    #Function for opening the Employee registration form inetrface
    def employee(self):
        self.empscr = Toplevel(self.wn)
        employ.Employee_Form(self.empscr)

    #Function for opening the interface for Department registration form
    def depart(self):
        self.depscr = Toplevel(self.wn)
        department.Department_Registration_Form(self.depscr)

    #Function for exit button
    def exit(self):
        self.wn.destroy()
```

Figure 12: Code for selection form interface

Employee registration interface

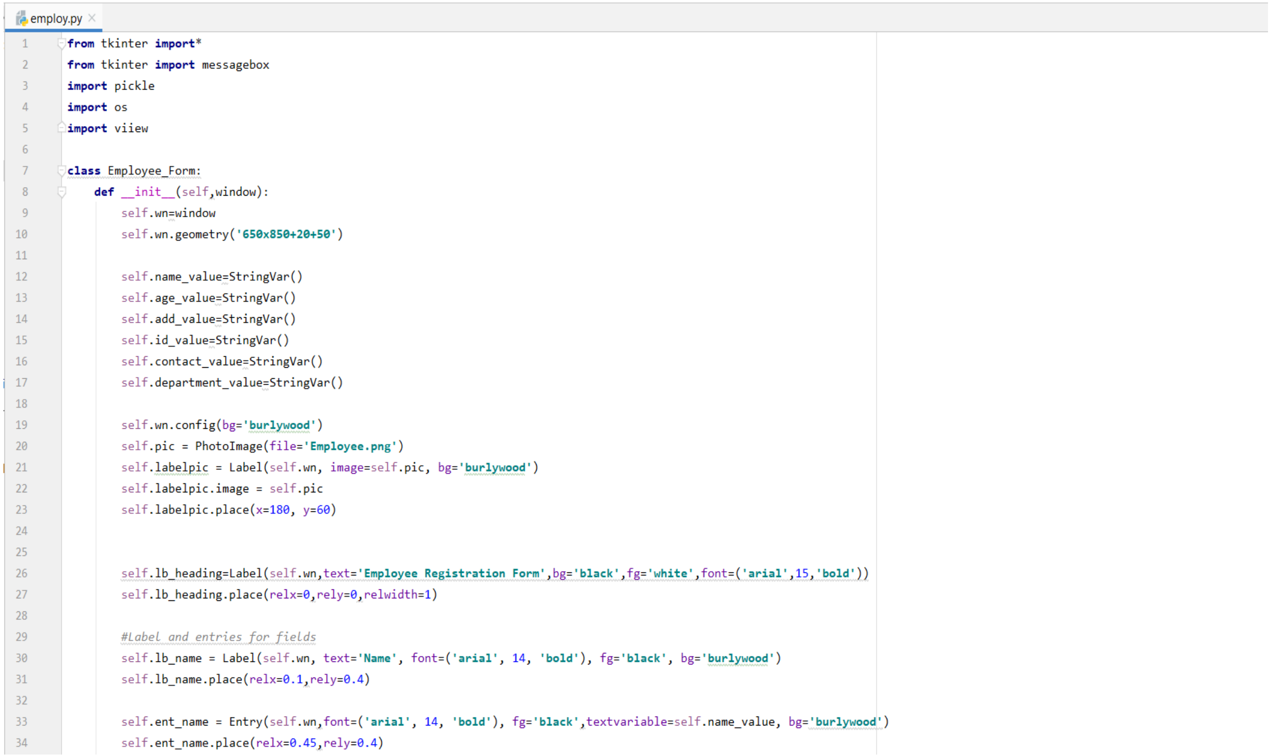


Employee Registration Form

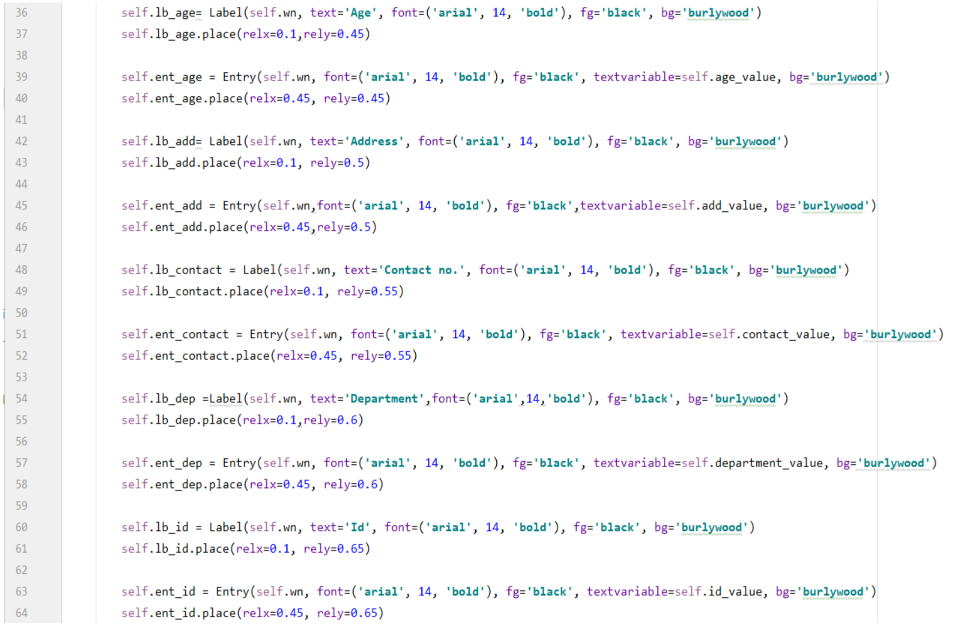
Name	
Age	
Address	
Contact no.	
Department	
Id	

Figure 13: Employee registration form interface

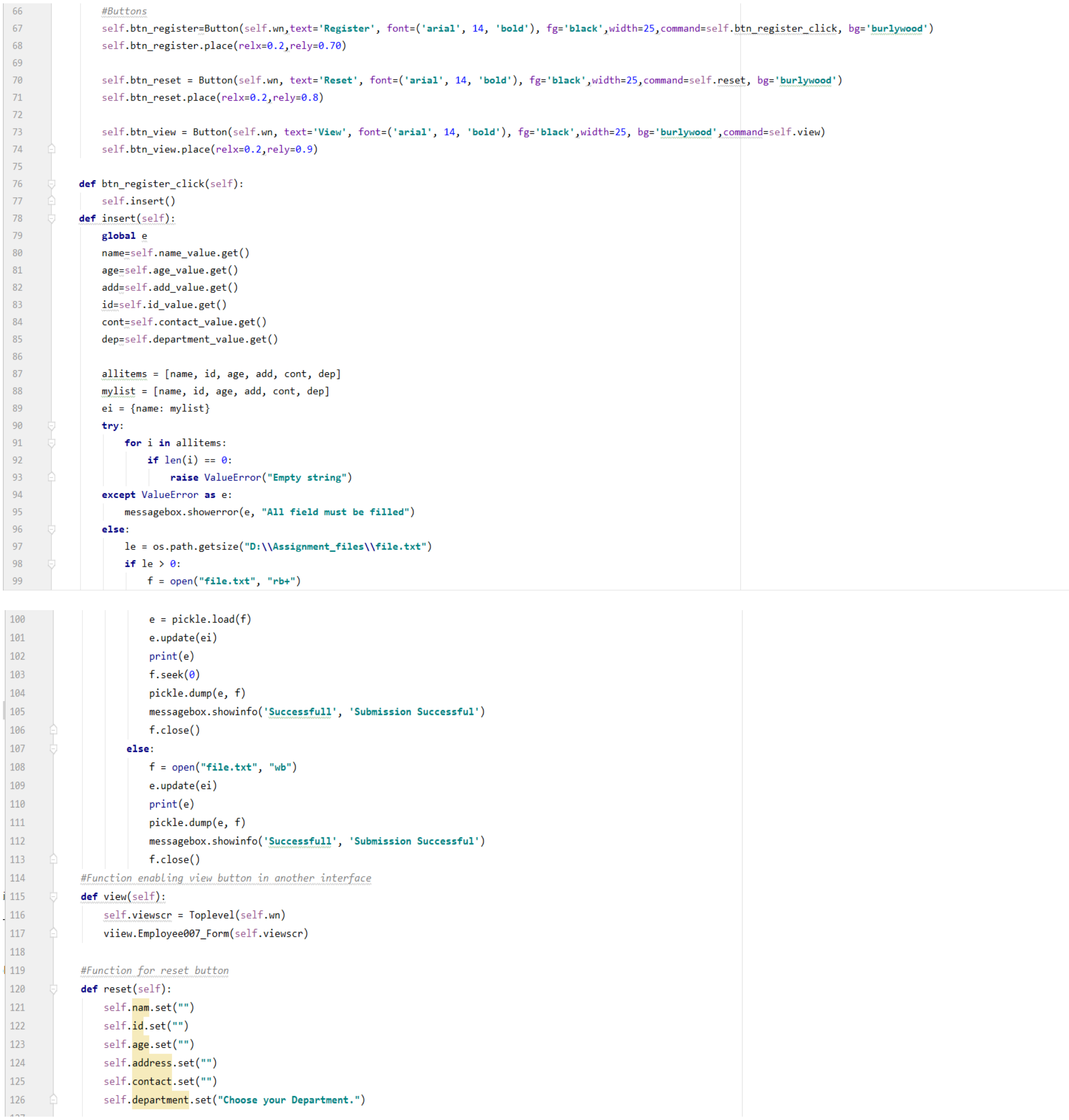
Employment form interface code



```python
from tkinter import*
from tkinter import messagebox
import pickle
import os
import viiew

class Employee_Form:
    def __init__(self,window):
        self.wn=window
        self.wn.geometry('650x850+20+50')

        self.name_value=StringVar()
        self.age_value=StringVar()
        self.add_value=StringVar()
        self.id_value=StringVar()
        self.contact_value=StringVar()
        self.department_value=StringVar()

        self.wn.config(bg='burlywood')
        self.pic = PhotoImage(file='Employee.png')
        self.labelpic = Label(self.wn, image=self.pic, bg='burlywood')
        self.labelpic.image = self.pic
        self.labelpic.place(x=180, y=60)


        self.lb_heading=Label(self.wn,text='Employee Registration Form',bg='black',fg='white',font=('arial',15,'bold'))
        self.lb_heading.place(relx=0,rely=0,relwidth=1)

        #Label and entries for fields
        self.lb_name = Label(self.wn, text='Name', font=('arial', 14, 'bold'), fg='black', bg='burlywood')
        self.lb_name.place(relx=0.1,rely=0.4)

        self.ent_name = Entry(self.wn,font=('arial', 14, 'bold'), fg='black',textvariable=self.name_value, bg='burlywood')
        self.ent_name.place(relx=0.45,rely=0.4)
```



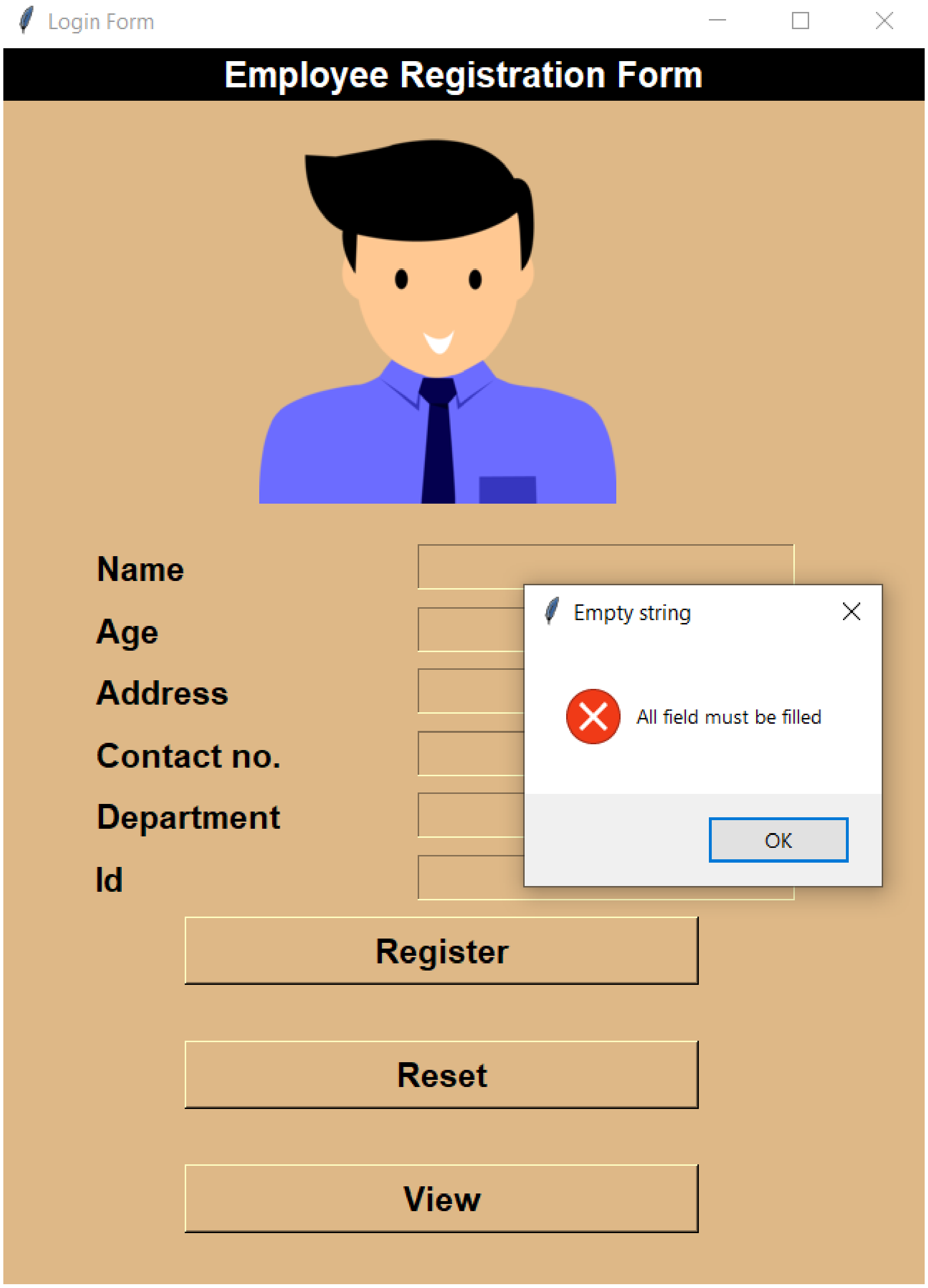
```python
        self.lb_age= Label(self.wn, text='Age', font=('arial', 14, 'bold'), fg='black', bg='burlywood')
        self.lb_age.place(relx=0.1,rely=0.45)

        self.ent_age = Entry(self.wn, font=('arial', 14, 'bold'), fg='black', textvariable=self.age_value, bg='burlywood')
        self.ent_age.place(relx=0.45, rely=0.45)

        self.lb_add= Label(self.wn, text='Address', font=('arial', 14, 'bold'), fg='black', bg='burlywood')
        self.lb_add.place(relx=0.1, rely=0.5)

        self.ent_add = Entry(self.wn,font=('arial', 14, 'bold'), fg='black',textvariable=self.add_value, bg='burlywood')
        self.ent_add.place(relx=0.45,rely=0.5)

        self.lb_contact = Label(self.wn, text='Contact no.', font=('arial', 14, 'bold'), fg='black', bg='burlywood')
        self.lb_contact.place(relx=0.1, rely=0.55)

        self.ent_contact = Entry(self.wn, font=('arial', 14, 'bold'), fg='black', textvariable=self.contact_value, bg='burlywood')
        self.ent_contact.place(relx=0.45, rely=0.55)

        self.lb_dep =Label(self.wn, text='Department',font=('arial',14,'bold'), fg='black', bg='burlywood')
        self.lb_dep.place(relx=0.1,rely=0.6)

        self.ent_dep = Entry(self.wn, font=('arial', 14, 'bold'), fg='black', textvariable=self.department_value, bg='burlywood')
        self.ent_dep.place(relx=0.45, rely=0.6)

        self.lb_id = Label(self.wn, text='Id', font=('arial', 14, 'bold'), fg='black', bg='burlywood')
        self.lb_id.place(relx=0.1, rely=0.65)

        self.ent_id = Entry(self.wn, font=('arial', 14, 'bold'), fg='black', textvariable=self.id_value, bg='burlywood')
        self.ent_id.place(relx=0.45, rely=0.65)
```



```
        #Buttons
        self.btn_register=Button(self.wn,text='Register', font=('arial', 14, 'bold'), fg='black',width=25,command=self.btn_register_click, bg='burlywood')
        self.btn_register.place(relx=0.2,rely=0.70)

        self.btn_reset = Button(self.wn, text='Reset', font=('arial', 14, 'bold'), fg='black',width=25,command=self.reset, bg='burlywood')
        self.btn_reset.place(relx=0.2,rely=0.8)

        self.btn_view = Button(self.wn, text='View', font=('arial', 14, 'bold'), fg='black',width=25, bg='burlywood',command=self.view)
        self.btn_view.place(relx=0.2,rely=0.9)

    def btn_register_click(self):
        self.insert()
    def insert(self):
        global e
        name=self.name_value.get()
        age=self.age_value.get()
        add=self.add_value.get()
        id=self.id_value.get()
        cont=self.contact_value.get()
        dep=self.department_value.get()

        allitems = [name, id, age, add, cont, dep]
        mylist = [name, id, age, add, cont, dep]
        ei = {name: mylist}
        try:
            for i in allitems:
                if len(i) == 0:
                    raise ValueError("Empty string")
        except ValueError as e:
            messagebox.showerror(e, "All field must be filled")
        else:
            le = os.path.getsize("D:\\Assignment_files\\file.txt")
            if le > 0:
                f = open("file.txt", "rb+")
                e = pickle.load(f)
                e.update(ei)
                print(e)
                f.seek(0)
                pickle.dump(e, f)
                messagebox.showinfo('Successfull', 'Submission Successful')
                f.close()
            else:
                f = open("file.txt", "wb")
                e.update(ei)
                print(e)
                pickle.dump(e, f)
                messagebox.showinfo('Successfull', 'Submission Successful')
                f.close()
    #Function enabling view button in another interface
    def view(self):
        self.viewscr = Toplevel(self.wn)
        viiew.Employee007_Form(self.viewscr)

    #Function for reset button
    def reset(self):
        self.nam.set("")
        self.id.set("")
        self.age.set("")
        self.address.set("")
        self.contact.set("")
        self.department.set("Choose your Department.")
```

Figure 14:Employment form interface code

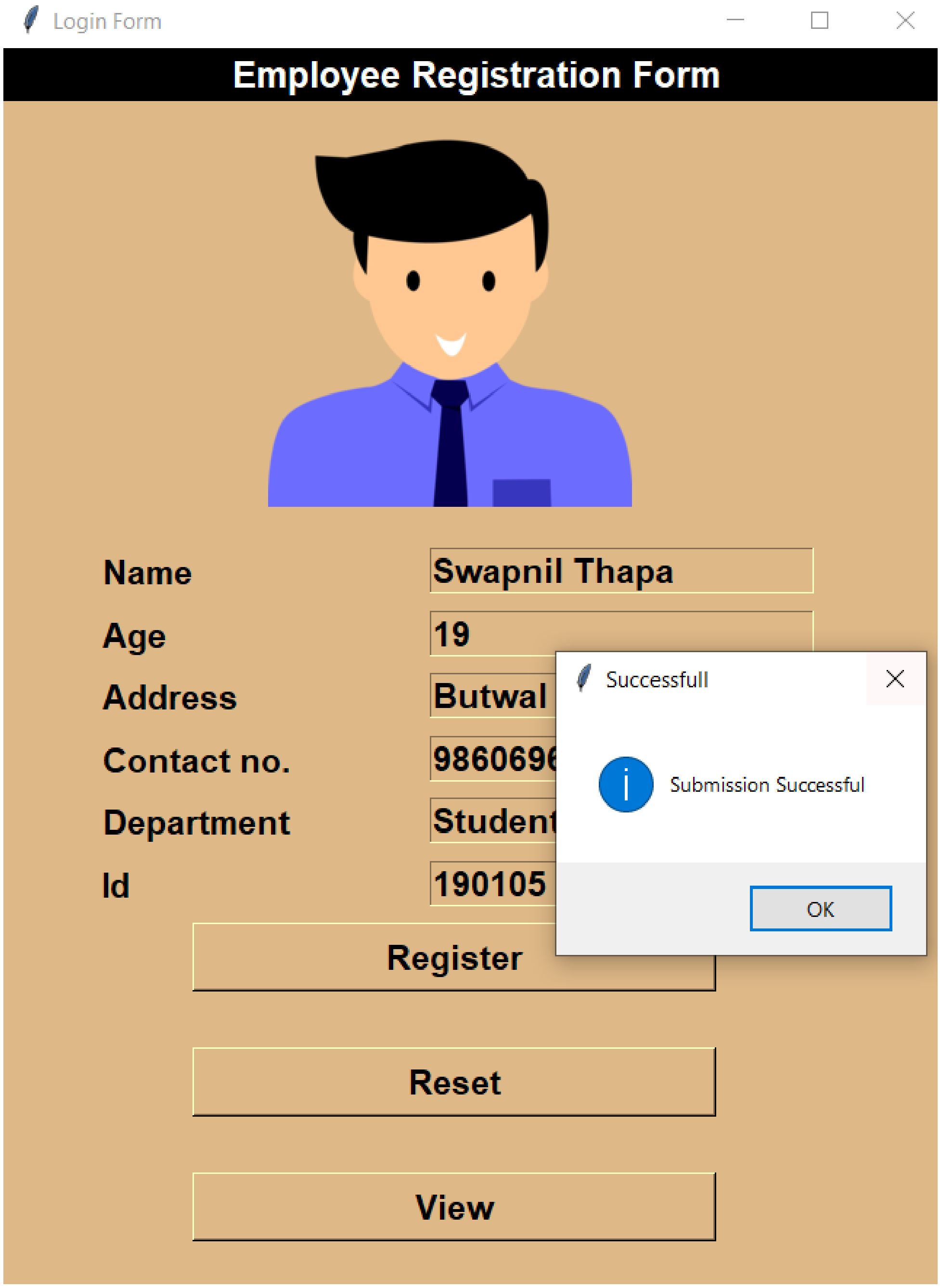
After clicking register button with no entries



All field must be filled

Figure 15: Employee registration interface with no entries

After clicking register button with filled entries



Employee Registration Form

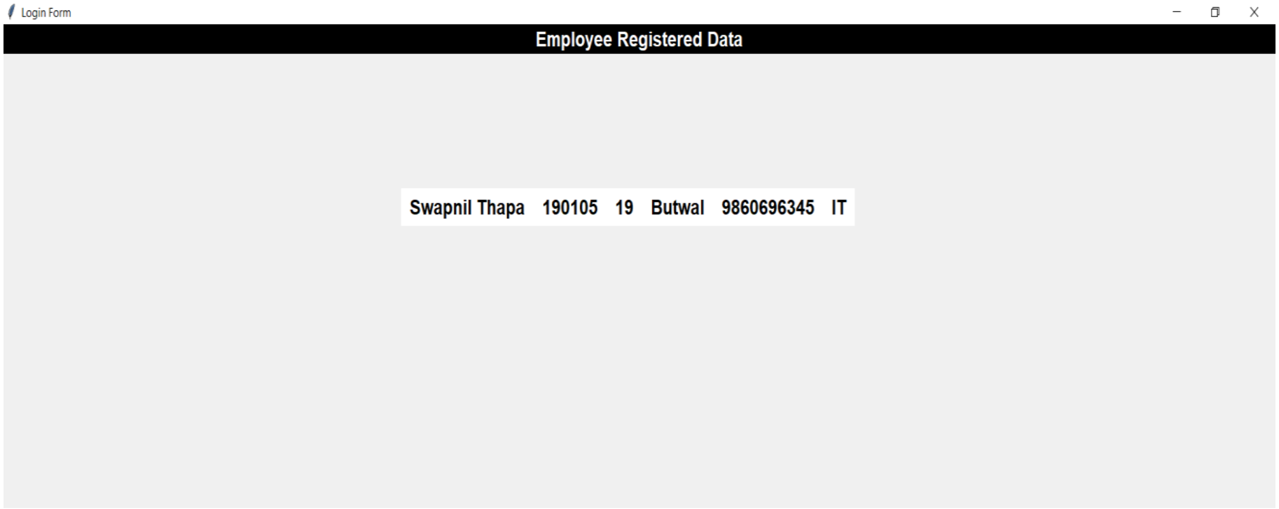
Name	Swapnil Thapa
Age	19
Address	Butwal
Contact no.	9860696
Department	Student
Id	190105

Successfull

Submission Successful

Figure 16: Employee registration form with filled entries

After clicking view button in Employee registration form

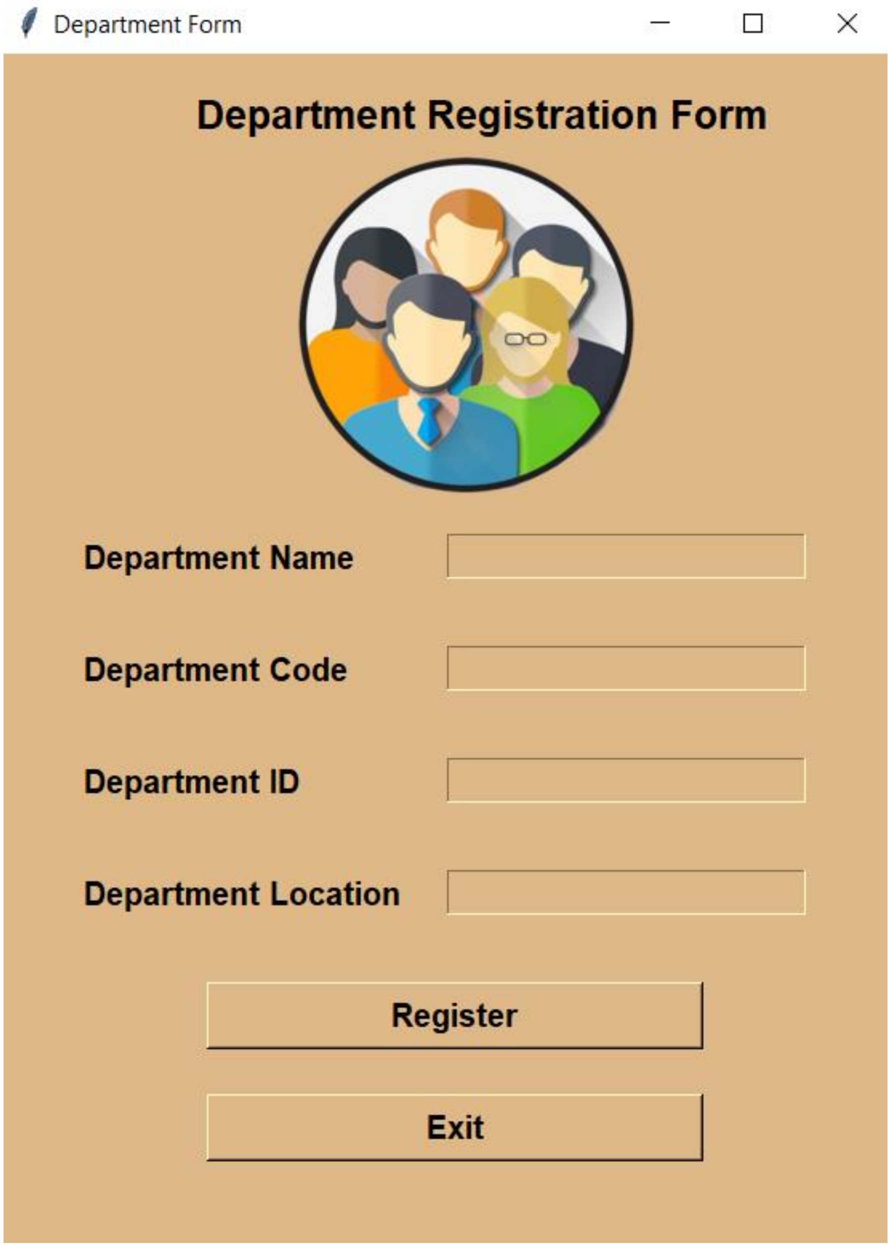


Employee Registered Data

Swapnil Thapa 190105 19 Butwal 9860696345 IT

Figure 17: Details of registered employee

Interface for Department form



Department Form

Department Registration Form

Department Name

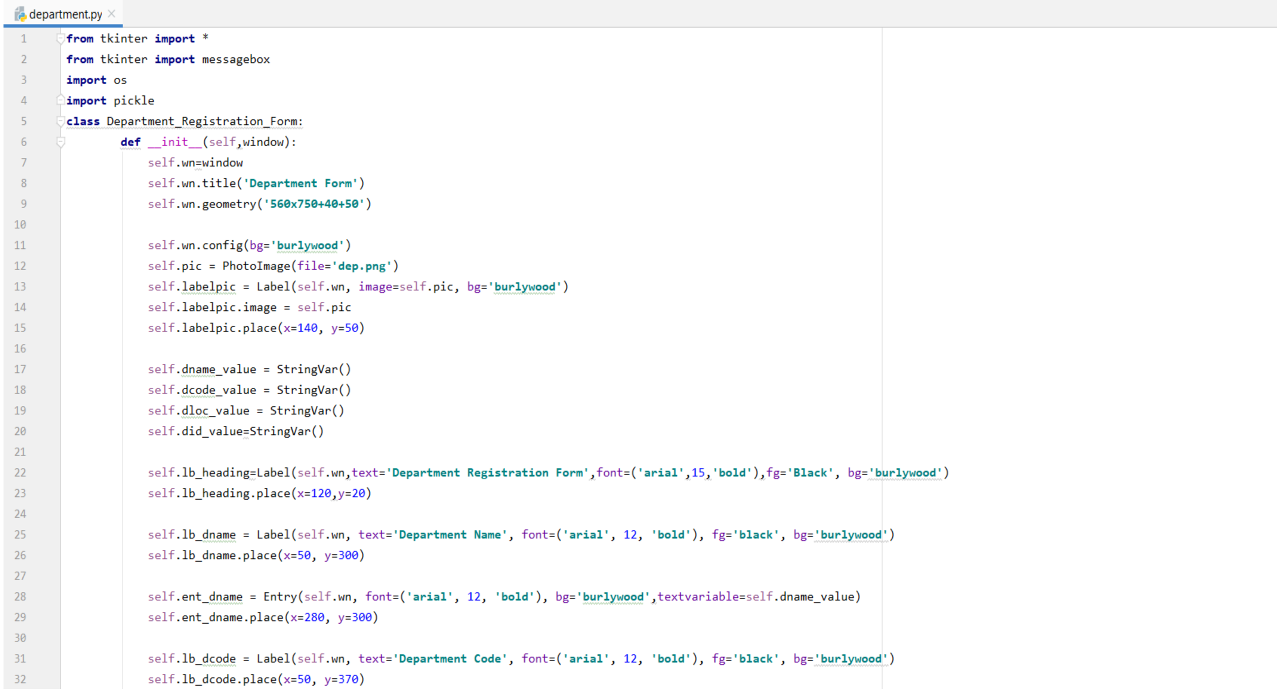
Department Code

Department ID

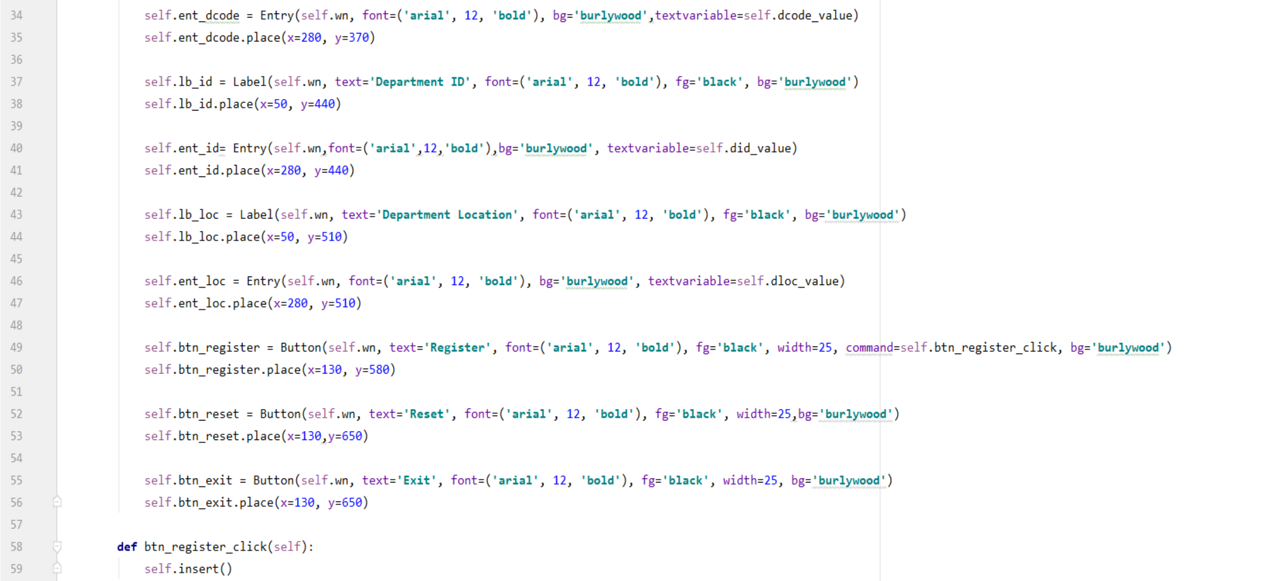
Department Location

Figure 18: Department form interface

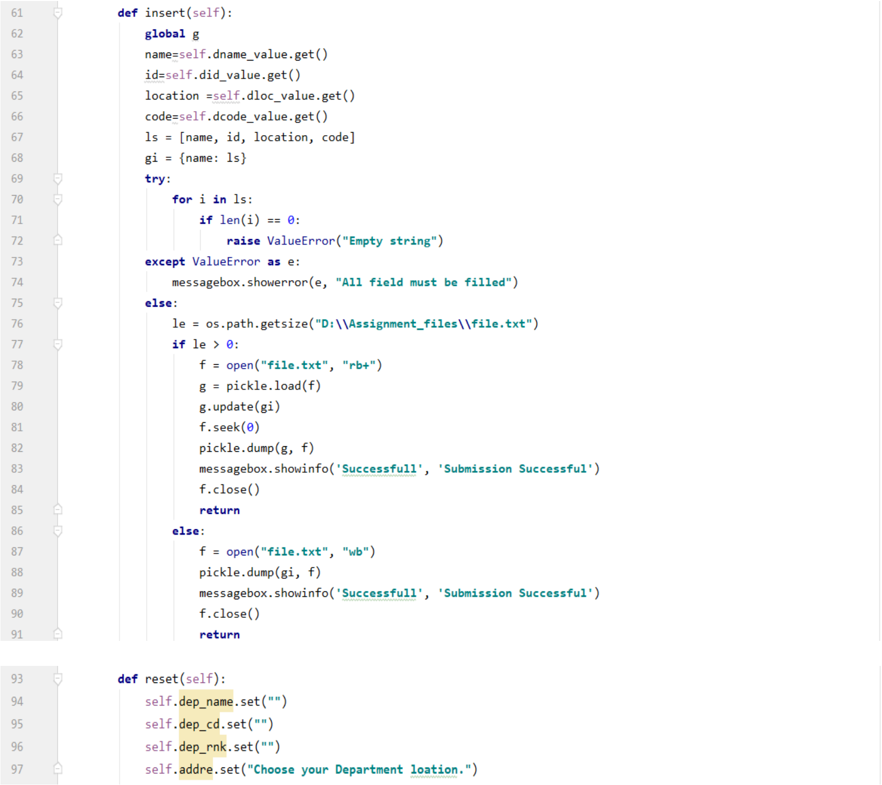
Code for department interface



```
from tkinter import *
from tkinter import messagebox
import os
import pickle
class Department_Registration_Form:
        def __init__(self,window):
            self.wn=window
            self.wn.title('Department Form')
            self.wn.geometry('560x750+40+50')

            self.wn.config(bg='burlywood')
            self.pic = PhotoImage(file='dep.png')
            self.labelpic = Label(self.wn, image=self.pic, bg='burlywood')
            self.labelpic.image = self.pic
            self.labelpic.place(x=140, y=50)

            self.dname_value = StringVar()
            self.dcode_value = StringVar()
            self.dloc_value = StringVar()
            self.did_value=StringVar()

            self.lb_heading=Label(self.wn,text='Department Registration Form',font=('arial',15,'bold'),fg='Black', bg='burlywood')
            self.lb_heading.place(x=120,y=20)

            self.lb_dname = Label(self.wn, text='Department Name', font=('arial', 12, 'bold'), fg='black', bg='burlywood')
            self.lb_dname.place(x=50, y=300)

            self.ent_dname = Entry(self.wn, font=('arial', 12, 'bold'), bg='burlywood',textvariable=self.dname_value)
            self.ent_dname.place(x=280, y=300)

            self.lb_dcode = Label(self.wn, text='Department Code', font=('arial', 12, 'bold'), fg='black', bg='burlywood')
            self.lb_dcode.place(x=50, y=370)
```



```
            self.ent_dcode = Entry(self.wn, font=('arial', 12, 'bold'), bg='burlywood',textvariable=self.dcode_value)
            self.ent_dcode.place(x=280, y=370)

            self.lb_id = Label(self.wn, text='Department ID', font=('arial', 12, 'bold'), fg='black', bg='burlywood')
            self.lb_id.place(x=50, y=440)

            self.ent_id= Entry(self.wn,font=('arial',12,'bold'),bg='burlywood', textvariable=self.did_value)
            self.ent_id.place(x=280, y=440)

            self.lb_loc = Label(self.wn, text='Department Location', font=('arial', 12, 'bold'), fg='black', bg='burlywood')
            self.lb_loc.place(x=50, y=510)

            self.ent_loc = Entry(self.wn, font=('arial', 12, 'bold'), bg='burlywood', textvariable=self.dloc_value)
            self.ent_loc.place(x=280, y=510)

            self.btn_register = Button(self.wn, text='Register', font=('arial', 12, 'bold'), fg='black', width=25, command=self.btn_register_click, bg='burlywood')
            self.btn_register.place(x=130, y=580)

            self.btn_reset = Button(self.wn, text='Reset', font=('arial', 12, 'bold'), fg='black', width=25,bg='burlywood')
            self.btn_reset.place(x=130,y=650)

            self.btn_exit = Button(self.wn, text='Exit', font=('arial', 12, 'bold'), fg='black', width=25, bg='burlywood')
            self.btn_exit.place(x=130, y=650)

        def btn_register_click(self):
            self.insert()
```



```
    def insert(self):
        global g
        name=self.dname_value.get()
        id=self.did_value.get()
        location =self.dloc_value.get()
        code=self.dcode_value.get()
        ls = [name, id, location, code]
        gi = {name: ls}
        try:
            for i in ls:
                if len(i) == 0:
                    raise ValueError("Empty string")
        except ValueError as e:
            messagebox.showerror(e, "All field must be filled")
        else:
            le = os.path.getsize("D:\\Assignment_files\\file.txt")
            if le > 0:
                f = open("file.txt", "rb+")
                g = pickle.load(f)
                g.update(gi)
                f.seek(0)
                pickle.dump(g, f)
                messagebox.showinfo('Successfull', 'Submission Successful')
                f.close()
                return
            else:
                f = open("file.txt", "wb")
                pickle.dump(gi, f)
                messagebox.showinfo('Successfull', 'Submission Successful')
                f.close()
                return

    def reset(self):
        self.dep_name.set("")
        self.dep_cd.set("")
        self.dep_rnk.set("")
        self.addre.set("Choose your Department loation.")
```

Figure 19: Code for department interface

After clicking register button with filled entries

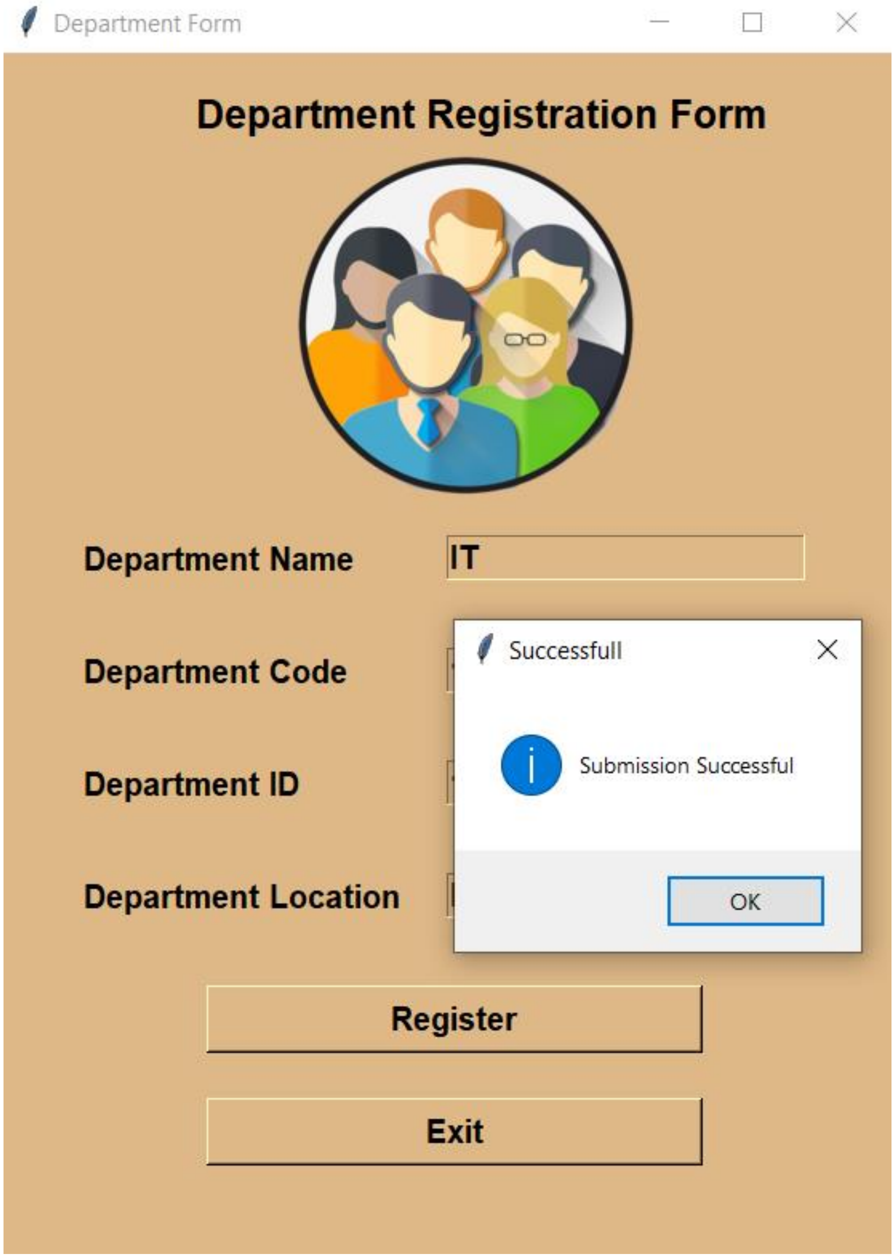
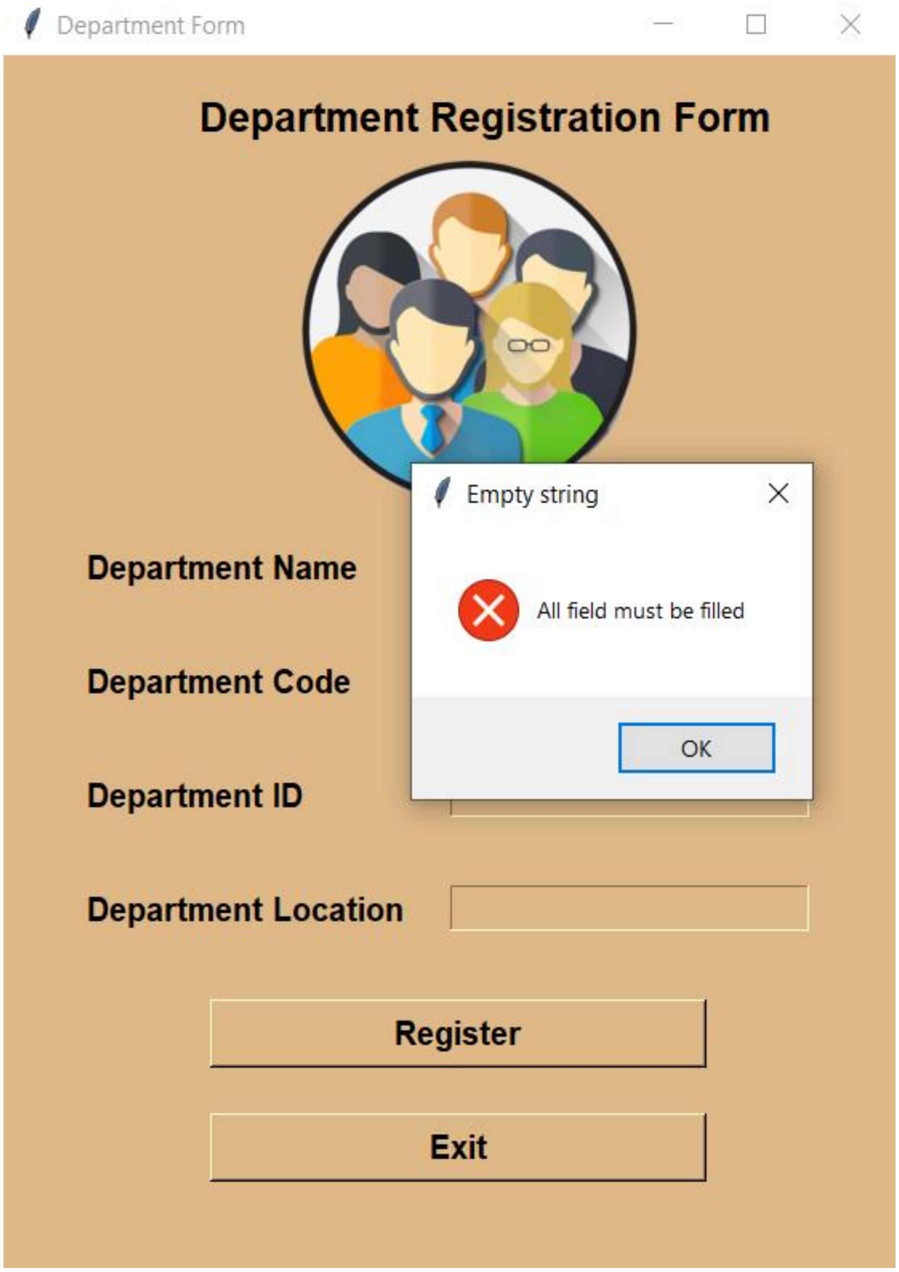


Figure 20: Department interface form with filled entries.

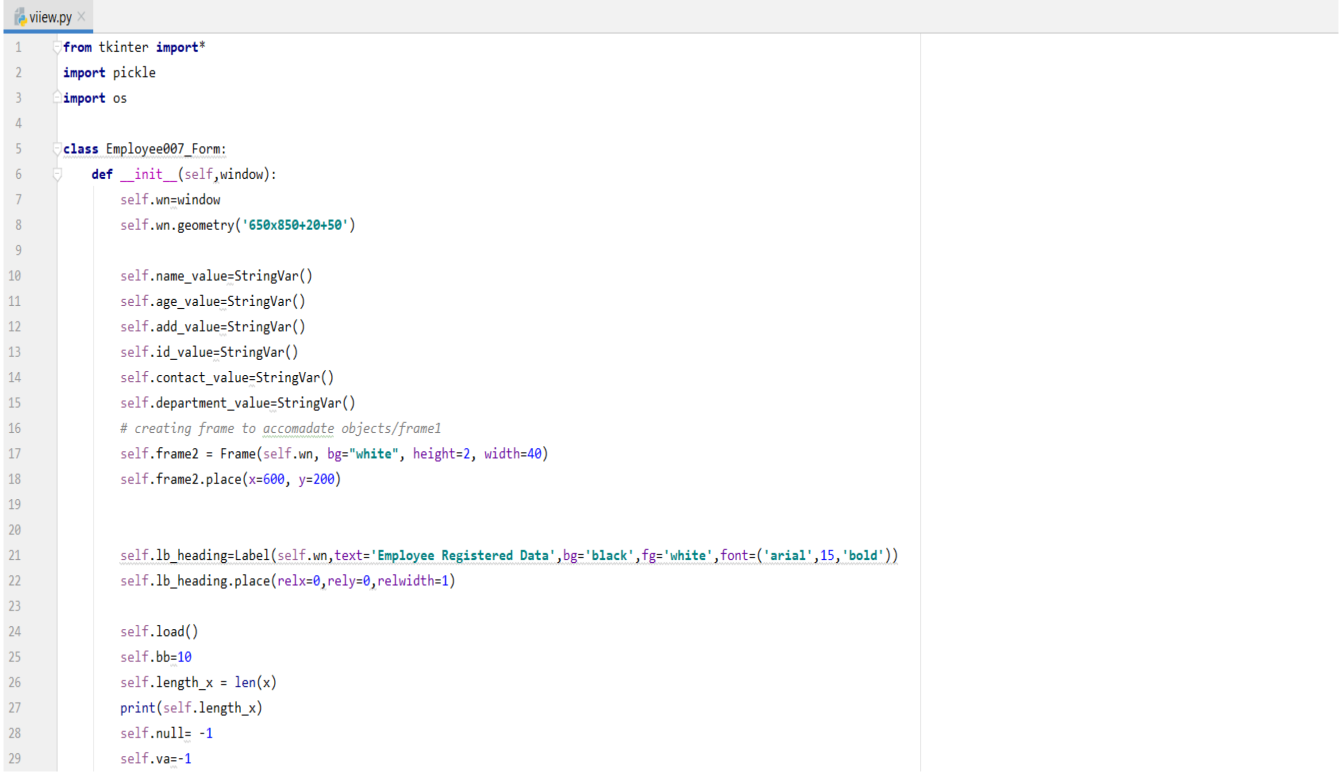
After clicking register with no entries



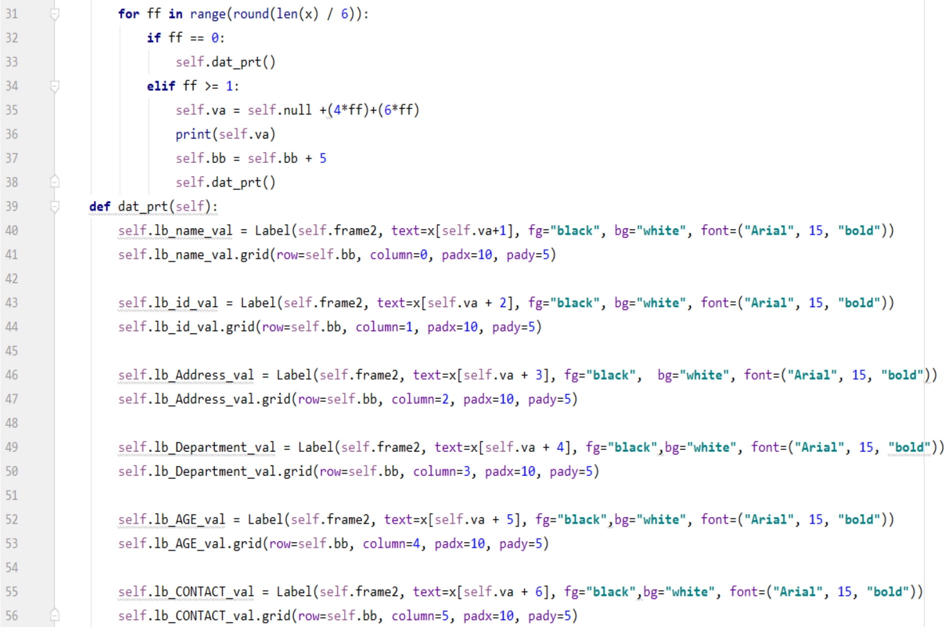
All field must be filled

Figure 21: Department interface form with no entries

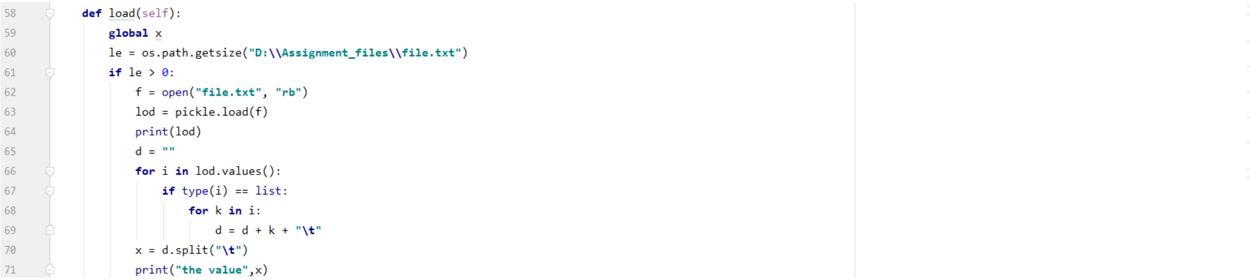
View interface code



```python
from tkinter import*
import pickle
import os

class Employee007_Form:
    def __init__(self,window):
        self.wn=window
        self.wn.geometry('650x850+20+50')

        self.name_value=StringVar()
        self.age_value=StringVar()
        self.add_value=StringVar()
        self.id_value=StringVar()
        self.contact_value=StringVar()
        self.department_value=StringVar()
        # creating frame to accomadate objects/frame1
        self.frame2 = Frame(self.wn, bg="white", height=2, width=40)
        self.frame2.place(x=600, y=200)


        self.lb_heading=Label(self.wn,text='Employee Registered Data',bg='black',fg='white',font=('arial',15,'bold'))
        self.lb_heading.place(relx=0,rely=0,relwidth=1)

        self.load()
        self.bb=10
        self.length_x = len(x)
        print(self.length_x)
        self.null= -1
        self.va=-1
```



```python
        for ff in range(round(len(x) / 6)):
            if ff == 0:
                self.dat_prt()
            elif ff >= 1:
                self.va = self.null +(4*ff)+(6*ff)
                print(self.va)
                self.bb = self.bb + 5
                self.dat_prt()
    def dat_prt(self):
        self.lb_name_val = Label(self.frame2, text=x[self.va+1], fg="black", bg="white", font=("Arial", 15, "bold"))
        self.lb_name_val.grid(row=self.bb, column=0, padx=10, pady=5)

        self.lb_id_val = Label(self.frame2, text=x[self.va + 2], fg="black", bg="white", font=("Arial", 15, "bold"))
        self.lb_id_val.grid(row=self.bb, column=1, padx=10, pady=5)

        self.lb_Address_val = Label(self.frame2, text=x[self.va + 3], fg="black",  bg="white", font=("Arial", 15, "bold"))
        self.lb_Address_val.grid(row=self.bb, column=2, padx=10, pady=5)

        self.lb_Department_val = Label(self.frame2, text=x[self.va + 4], fg="black",bg="white", font=("Arial", 15, "bold"))
        self.lb_Department_val.grid(row=self.bb, column=3, padx=10, pady=5)

        self.lb_AGE_val = Label(self.frame2, text=x[self.va + 5], fg="black",bg="white", font=("Arial", 15, "bold"))
        self.lb_AGE_val.grid(row=self.bb, column=4, padx=10, pady=5)

        self.lb_CONTACT_val = Label(self.frame2, text=x[self.va + 6], fg="black",bg="white", font=("Arial", 15, "bold"))
        self.lb_CONTACT_val.grid(row=self.bb, column=5, padx=10, pady=5)
```



```
def load(self):
    global x
    le = os.path.getsize("D:\\Assignment_files\\file.txt")
    if le > 0:
        f = open("file.txt", "rb")
        lod = pickle.load(f)
        print(lod)
        d = ""
        for i in lod.values():
            if type(i) == list:
                for k in i:
                    d = d + k + "\t"
        x = d.split("\t")
        print("the value",x)
```

Figure 22: Code for view button interface

3. Conclusion

This program was implemented by importing Tkinter module for GUI interface. The program code was brought into the action as the individual registers him/herself into the system and is provided with range of facilities for storing the data, concerned with employee's basic data along with the department. The program was facilitated by various functions which were either user defined or a part of python built-in library. The program is truly efficient for storing employee data and later view as per needed. The project indeed act as platform to enhance the knowledge and ensure high level of understanding regarding python programming language, python project and its implementation in real life scenario.

4. References

tutorialspoint, 2020. *tutorialspoint.* [Online]
Available at: https://www.tutorialspoint.com/python/python_gui_programming.htm
[Accessed 2020].