



**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: 21<sup>st</sup> February, 2020**

**ASSIGNMENT SUBMISSION DATE: 21<sup>st</sup> February, 2020**

# Table of Content

1. Introduction.....	4
2.Screenshots of User Interface and Code .....	5
User Login Form Interface.....	5
Login form interface code .....	6
User Registration Form Interface .....	12
User registration interface code .....	13
Selection form interface .....	18
Selection form interface code.....	19
Employee registration interface.....	20
Employment form interface code .....	21
Interface for Department form .....	26
Code for department interface .....	27
View interface code .....	31
3. Conclusion .....	33
4. References .....	34

## List of Figures:

Figure 1: User Login Interface .....	5
Figure 2: Code for User Login interface .....	7
Figure 3: Login button clicked filling wrong username and password .....	8
Figure 4: Login button clicked without entries .....	9
Figure 5: Forgot password button clicked .....	10
Figure 6: Register button clicked with registered username and password .....	11
Figure 7: User registration interface .....	12
Figure 8: Code for user signup interface .....	15
Figure 9: Clicked register button without entries .....	16
Figure 10: Login interface opened after clicking register .....	17
Figure 11: Selection form interface window .....	18
Figure 12: Code for selection form interface .....	19
Figure 13: Employee registration form interface .....	20
Figure 14: Employment form interface code .....	22
Figure 15: Employee registration interface with no entries .....	23
Figure 16: Employee registration form with filled entries .....	24
Figure 17: Details of registered employee .....	25
Figure 18: Department form interface .....	26
Figure 19: Code for department interface .....	28
Figure 20: Department interface form with filled entries. ....	29
Figure 21: Department interface form with no entries .....	30
Figure 22: Code for view button interface .....	32

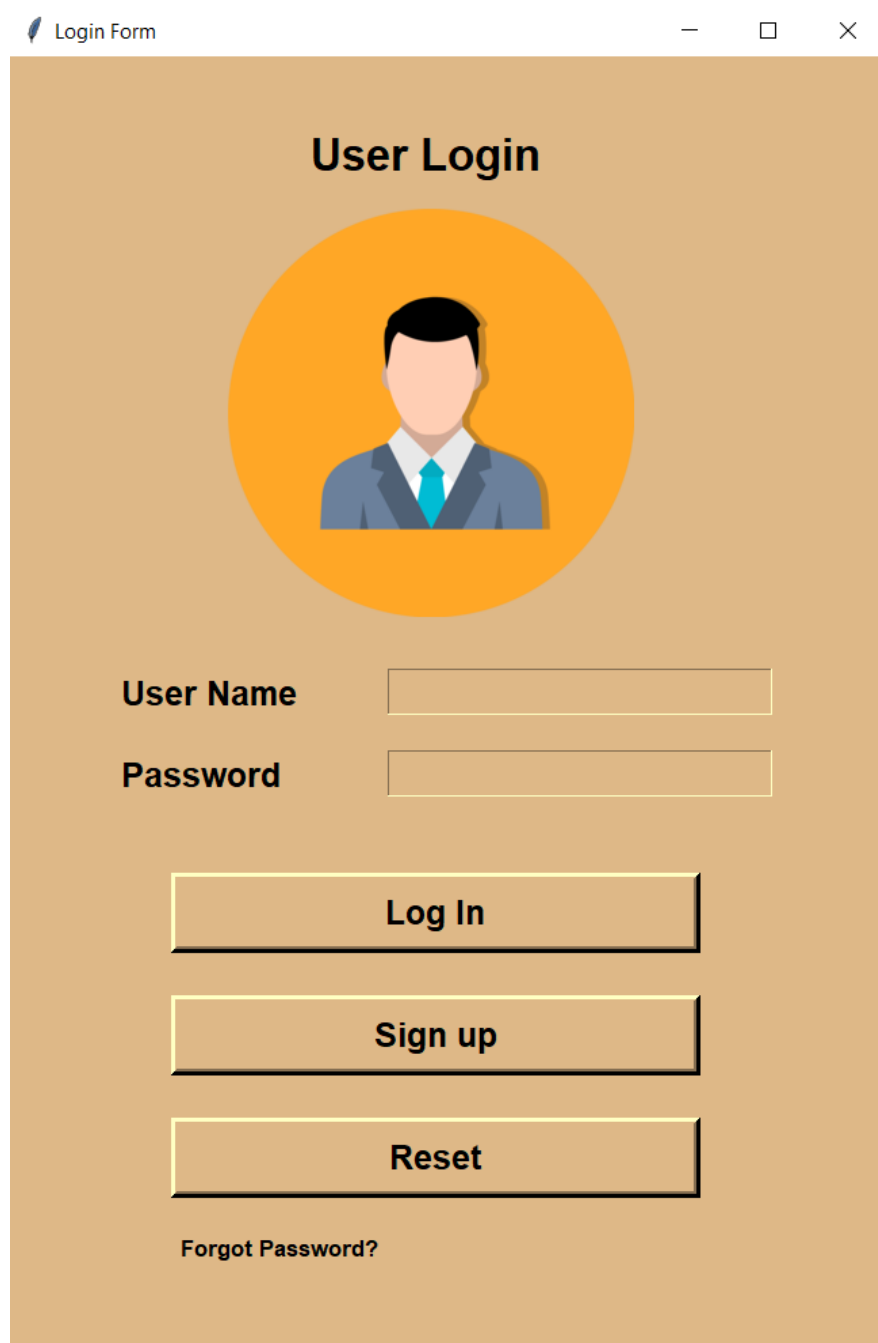
# 1. Introduction

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



The image shows a web browser window titled "Login Form". The page has a light brown background. At the top center, the text "User Login" is displayed in a bold, black font. Below this text is a large orange circle containing a stylized illustration of a person with black hair, wearing a blue suit jacket, a white shirt, and a blue tie. Underneath the circle, there are two input fields. The first is labeled "User Name" and the second is labeled "Password". Both labels are in a bold, black font. Below the input fields, there are three buttons stacked vertically. The first button is labeled "Log In", the second is labeled "Sign up", and the third is labeled "Reset". All buttons have a light brown background and a thin black border. At the bottom of the form, there is a link labeled "Forgot Password?" in a smaller, black font.

Figure 1: User Login Interface

## Login form interface code

```
1 from tkinter import *
2 import os
3 import pickle
4 import user_registration_form
5 from tkinter import messagebox
6 import select_form
7 class Login_Form:
8     def __init__(self,window):
9         self.wn=window
10        self.wn.title('Login Form')
11        self.wn.geometry('650x950+40+50')
12
13        #Top image
14        self.wn.config(bg='burlywood')
15        self.pic = PhotoImage(file='login.png')
16        self.labelpic= Label(self.wn,image=self.pic,bg='burlywood')
17        self.labelpic.image=self.pic
18        self.labelpic.place(x=160,y=110)
19
20        self.uname=StringVar()
21        self.pw=StringVar()
22
23        #Label for interface heading
24        self.lb_heading=Label(self.wn,text='User Login',font=('arial',20,'bold'),fg='black',bg='burlywood')
25        self.lb_heading.place(x=220,y=50)
26
27        #Label and Entry for username
28
29        self.lb_username=Label(self.wn,text='User Name',font=('arial',15,'bold'),fg='black',bg='burlywood')
30        self.lb_username.place(x=80,y=450)
31
32        self.ent_username=Entry(self.wn,font=('arial',15,'bold'),textvariable=self.uname,bg='burlywood')
33        self.ent_username.place(x=280,y=450)
34
35        # Label and Entry for password
36        self.lb_pass = Label(self.wn, text='Password', font=('arial', 15, 'bold'), fg='black',bg='burlywood')
37        self.lb_pass.place(x=80,y=510)
38
39        self.ent_pass = Entry(self.wn, font=('arial', 15, 'bold'),show='*',bg='burlywood',textvariable=self.pw)
40        self.ent_pass.place(x=280,y=510)
41
42        #Login Button
43
44        self.btnlog = Button(self.wn, text='Log In', bd=5, font=('arial', 15, 'bold'), width=25,command=self.btn_login_click, bg='burlywood')
45        self.btnlog.place(x=120, y=600)
46
47        #Signup Button
48        self.btn_signup = Button(self.wn, text='Sign up', bd=5, font=('arial', 15, 'bold'), width=25,command=self.btn_sign_up, bg='burlywood')
49        self.btn_signup.place(x=120, y=690)
50
51        #Reset Button
52        self.btnres = Button(self.wn, text='Reset',bd=5, font=('arial', 15, 'bold'),width=25,bg='burlywood',command=self.res)
53        self.btnres.place(x=120,y=780)
54
55        #Forgot Password Button
56        self.btnfp = Button(self.wn, text='Forgot Password?', font=('arial',10,'bold'),fg='black',relief=FLAT,command=self.fp,bg='burlywood')
57        self.btnfp.place(x=120,y=860)
58
59        #
60        def btn_sign_up(self):
61            self.usr_window=Toplevel(self.wn)
62            user_registration_form.User_Form(self.usr_window)
63
64        def fp(self):
65            messagebox.showinfo('Please check your e-mail','Dear user a mail has been sent to your e-mail.')
66
67        #Function for Login button
68        def btn_login_click(self):
69            self.load()
```

```

71     #Function for shifting in new interface
72     def clk_ok(self):
73         self.user_window = Toplevel(self.wn)
74         select_form.Selection_Form(self.user_window)
75
76     #Function for reset button
77     def res(self):
78         self.uname.set("")
79         self.pw.set("")
80
81     #Function for Login using users saved from file
82     def load(self):
83
84         le = os.path.getsize("D:\\Assignment_files\\file.txt")
85         if le > 0:
86             f = open("file.txt", "rb")
87             lod = pickle.load(f)
88             try:
89                 if len(self.ent_username.get()) == 0 or len(self.ent_pass.get()) == 0:
90                     raise ValueError("Empty Feild")
91             except ValueError as hh:
92                 messagebox.showerror(hh, "All entries must be filled.")
93                 f.close()
94             else:
95
96                 for i, j in lod.items():
97                     pos_value = 0
98                     neg_value = 0
99                     if i == self.ent_username.get() and j == self.ent_pass.get():
100                         pos_value = pos_value + 1
101                         break
102                     else:
103                         neg_value = neg_value + 1
104                 if pos_value == 1:
105                     messagebox.showinfo("Successful login", f"Successfully registered Welcome {i}")
106
107                 else:
108                     messagebox.showerror('Error', "The entered data are incorrect.")
109
110                 f.close()
111             else:
112                 messagebox.showerror("Empty", "File is empty")
113
114
115
116
117 wn=Tk()
118 Login_Form(wn)
119 wn.mainloop()

```

Figure 2: Code for User Login interface

After Login button is clicked by wrong Username and Password

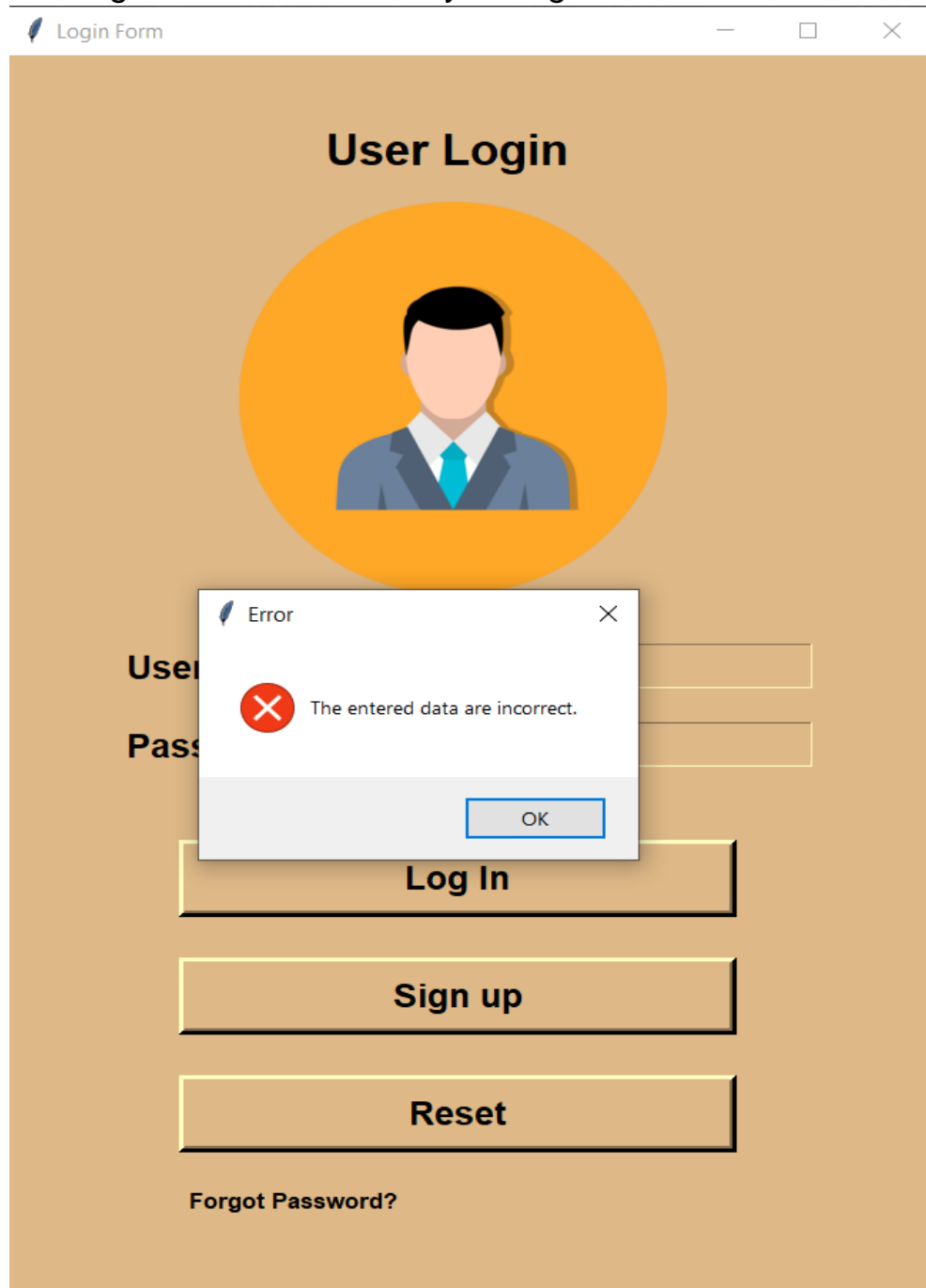


Figure 3: Login button clicked filling wrong username and password



After Login button clicked filling no any entries

The image shows a web application window titled "Login Form". The main heading is "User Login". Below the heading is a large orange circle containing a stylized user icon. Underneath the icon are two input fields: "User Name" and "Password". Below these fields are three buttons: "Log In", "Sign up", and "Reset". At the bottom, there is a link "Forgot Password?". An error message dialog box is displayed over the form. The dialog box is titled "Empty Feild" and contains a red "X" icon and the text "All entries must be filled.". There is an "OK" button at the bottom of the dialog box.

Figure 4: Login button clicked without entries

After Forgot password button is clicked

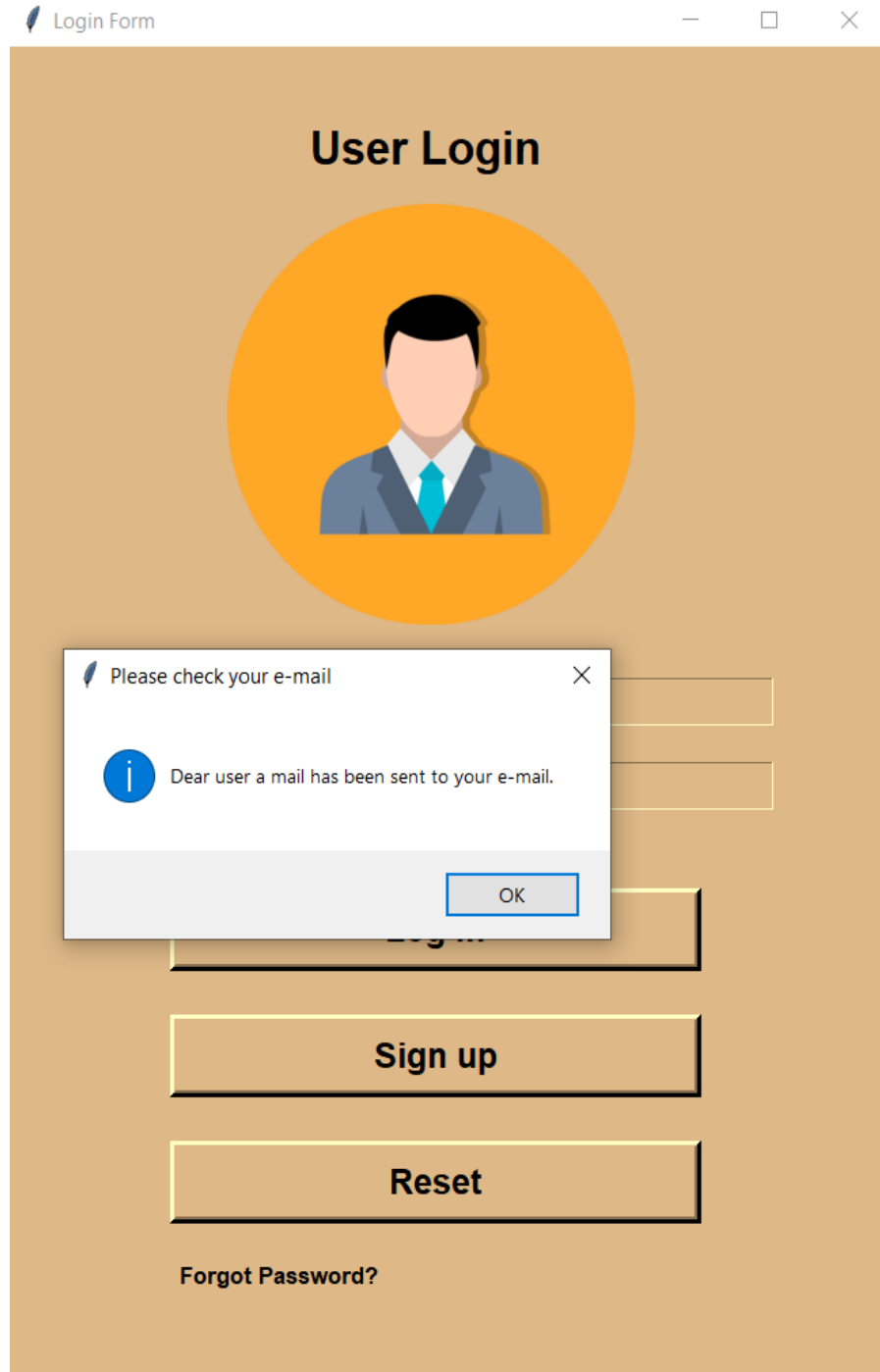


Figure 5: Forgot password button clicked

After clicking register button with registered username and password

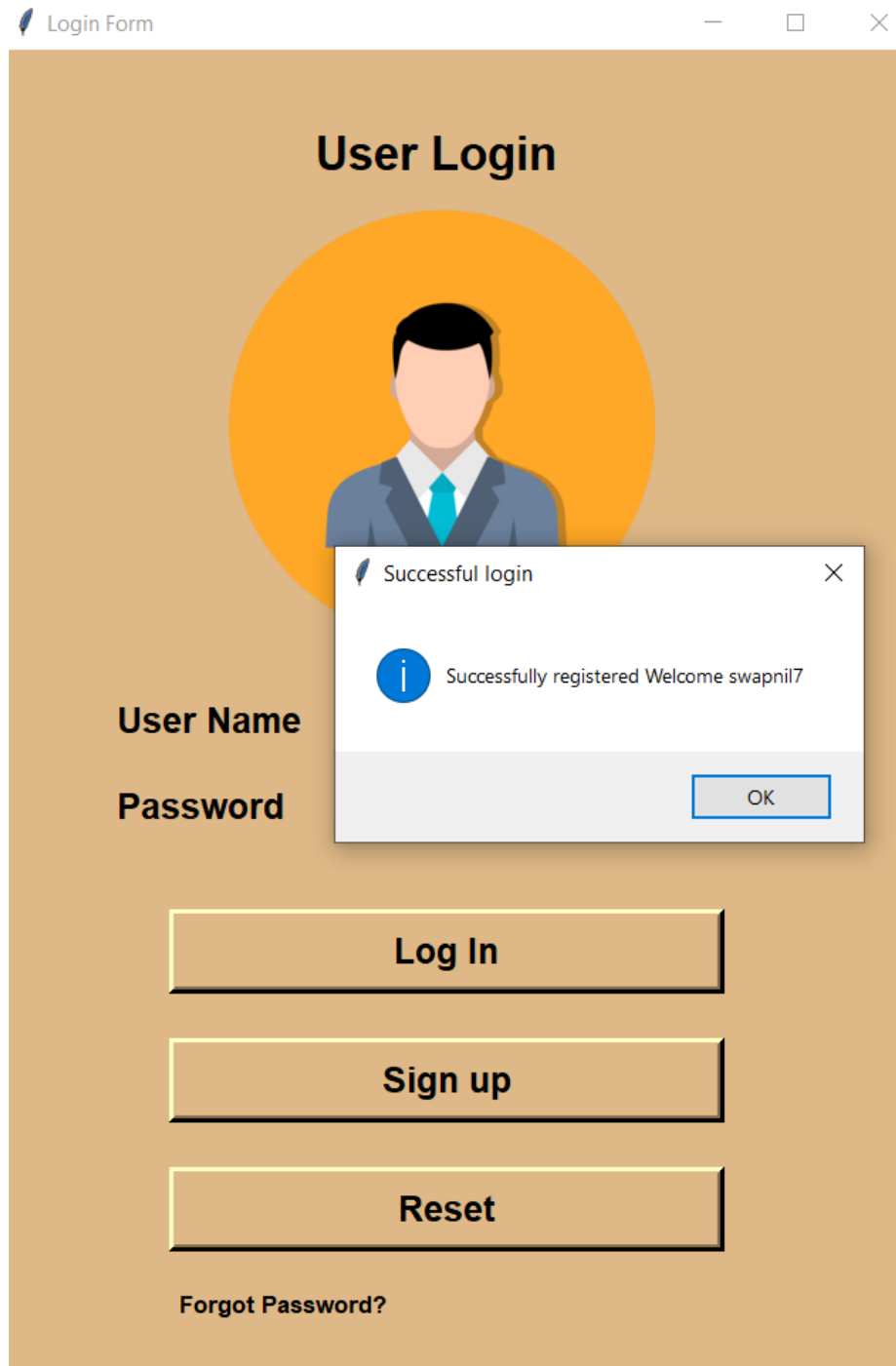


Figure 6: Register button clicked with registered username and password

## User Registration Form Interface

Login Form

### User Registration Form

**Full Name**

**User Name**

**Password**

**Address**

**Contact Number**

**Gender** ☒ Male ☐ Female

**Age**

**Date of Birth**

**Register** **Reset**

Figure 7: User registration interface

## User registration interface code

```
user_registration_form.py X
1 import tkinter as tk
2 from tkinter import*
3 from tkinter import messagebox
4 import pickle
5 import os
6 d = {}
7 class User_Form:
8     def __init__(self,window):
9         self.wn=window
10        self.wn.geometry('650x850+20+50')
11        self.wn.config(bg='burlywood')
12
13
14        self.first=StringVar()
15        self.user_value=StringVar()
16        self.pass_value=StringVar()
17        self.add_value=StringVar()
18        self.contact_value=StringVar()
19        self.gen=StringVar()
20        self.date1=StringVar()
21        self.date2=StringVar()
22        self.date3=StringVar()
23        self.age=StringVar()
24
25        self.gen_value=self.gen.get()
26        self.date1_value=self.date1.get()
27        self.date2_value =self.date2.get()
28        self.date3_value =self.date3.get()
29
30        #Label for
31        self.lb_heading=Label(self.wn,text='User Registration Form',bg='black',fg='white',font=('arial',15,'bold'))
32        self.lb_heading.place(relx=0,relx=0,relwidth=1)
33
34        # Labels and entry for data fields
35        self.lb_fullname = Label(self.wn, text='Full Name', font=('arial', 14, 'bold'), fg='black',bg='burlywood')
36        self.lb_fullname.place(relx=0.1, relx=0.1)
37
38        self.ent_fullname = Entry(self.wn, font=('arial', 14, 'bold'), fg='black', textvariable=self.first,bg='burlywood')
39        self.ent_fullname.place(relx=0.45, relx=0.1)
40
41
42        self.lb_username = Label(self.wn, text='User Name', font=('arial', 14, 'bold'), fg='black',bg='burlywood')
43        self.lb_username.place(relx=0.1,relx=0.2)
44
45        self.ent_username = Entry(self.wn,font=('arial', 14, 'bold'), fg='black',textvariable=self.user_value,bg='burlywood')
46        self.ent_username.place(relx=0.45,relx=0.2)
47
48        self.lb_pass= Label(self.wn, text='Password', font=('arial', 14, 'bold'), fg='black',bg='burlywood')
49        self.lb_pass.place(relx=0.1,relx=0.3)
50
51        self.ent_pass = Entry(self.wn, font=('arial', 14, 'bold'), fg='black', textvariable=self.pass_value,bg='burlywood')
52        self.ent_pass.place(relx=0.45, relx=0.3)
53
54        self.lb_add= Label(self.wn, text='Address', font=('arial', 14, 'bold'), fg='black',bg='burlywood')
55        self.lb_add.place(relx=0.1,relx=0.4)
56
57        self.ent_add = Entry(self.wn,font=('arial', 14, 'bold'), fg='black',textvariable=self.add_value,bg='burlywood')
58        self.ent_add.place(relx=0.45,relx=0.4)
59
60        self.lb_contact = Label(self.wn, text='Contact Number', font=('arial', 14, 'bold'), fg='black',bg='burlywood')
61        self.lb_contact.place(relx=0.1, relx=0.5)
62
63        self.ent_contact = Entry(self.wn, font=('arial', 14, 'bold'), fg='black', textvariable=self.contact_value,bg='burlywood')
64        self.ent_contact.place(relx=0.45, relx=0.5)
65
66        self.lb_gen =Label(self.wn, text='Gender',font=('arial',14,'bold'), fg='black',bg='burlywood')
67        self.lb_gen.place(relx=0.1,relx=0.6)
68        Radiobutton(self.wn,text='Male',value='Male',variable=self.gen_value,bg='burlywood').place(relx=0.45,relx=0.6)
```

```

69 Radiobutton(self.wn, text='Female',value='Female',variable=self.gen_value,bg='burlywood').place(relx=0.75, rely=0.6)
70
71 self.lb_age =Label(self.wn,text='Age',font=('arial', 14, 'bold'), fg='black',bg='burlywood')
72 self.lb_age.place(relx=0.1, rely=0.67)
73
74 self.age = [x for x in range(18,66)]
75 self.droplist_age=OptionMenu(self.wn,self.ag,*self.age)
76 self.droplist_age.config(width=15)
77 self.droplist_age.config(bg='burlywood',foreground='white')
78 self.ag.set('Age')
79 self.droplist_age.place(relx=0.45,rely=0.67)
80
81
82 self.lb_dob =Label(self.wn,text='Date of Birth',font=('arial', 14, 'bold'), fg='black',bg='burlywood')
83 self.lb_dob.place(relx=0.1, rely=0.74)
84
85
86 self.month = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']
87 self.droplist_month=OptionMenu(self.wn,self.date1,*self.month)
88 self.droplist_month.config(width=15)
89 self.droplist_month.config(bg='burlywood',foreground='white')
90 self.date1.set('Month')
91 self.droplist_month.place(relx=0.1,rely=0.8)
92
93 self.day=[x for x in range(1,31)]
94 self.droplist_day=OptionMenu(self.wn,self.date2,*self.day)
95 self.droplist_day.config(bg='burlywood',foreground='white')
96 self.date2.set('Day')
97 self.droplist_day.place(relx=0.4,rely=0.8)
98 self.droplist_day.config(width=15)
99
100 self.year = [x for x in range(1980,2021)]
101 self.droplist_year = OptionMenu(self.wn, self.date3, *self.year)
102 self.droplist_year.config(bg='burlywood', foreground='white')
103 self.date3.set('Year')
104 self.droplist_year.place(relx=0.7,rely=0.8)
105 self.droplist_year.config(width=15)
106
107 self.btn_register=Button(self.wn,text='Register', font=('arial', 14, 'bold'), fg='black',command=self.btn_register_click,width=15,bg='burlywood')
108 self.btn_register.place(relx=0.1,rely=0.9)
109
110 self.btn_reset = Button(self.wn, text='Reset', font=('arial', 14, 'bold'), fg='black',width=15,command=self.reset,bg='burlywood')
111 self.btn_reset.place(relx=0.6,rely=0.9)
112
113 def btn_register_click(self):
114     self.insert()
115
116 #Function for storing data in text file
117 def insert(self):
118     global d
119     username = self.user_value.get()
120     password = self.pass_value.get()
121     address = self.add_value.get()
122     name = self.first.get()
123     contact = self.contact_value.get()
124     gender = self.gen_value
125     month = self.date1_value
126     year = self.date2_value
127     day = self.date3_value
128
129     self.ls = [name,username, password, address,contact,gender,month,year,day]
130     try:
131         for i in self.ls:
132             if len(i) == 0:
133                 raise ValueError("Empty string")

```

```

134     except ValueError as e:
135         messagebox.showerror(e, "All entries must be filled")
136     else:
137         try:
138             if len(password) < 6:
139                 raise ValueError("Inadequate alphabets")
140         except ValueError as h:
141             messagebox.showerror(h, "A password must be at least of 6 characters.")
142     else:
143         di = {username: password}
144         le = os.path.getsize("D:\\Assignment_files\\file.txt")
145         if le > 0:
146             f = open("file.txt", "rb+")
147             d = pickle.load(f)
148             d.update(di)
149             f.seek(0)
150             pickle.dump(d, f)
151             messagebox.showinfo('Successful', 'Submission Successful')
152             f.close()
153         else:
154             f = open("file.txt", "wb")
155             d.update(di)
156             pickle.dump(d, f)
157             messagebox.showinfo('Successful', 'Submission Successful')
158             f.close()
159
160     #Function for reset button
161     def reset(self):
162         self.first.set("")
163         self.user_value.set("")
164         self.pass_value.set("")
165         self.add_value.set("")
166         self.contact_value.set("")
167         self.date1.set("Month")
168         self.date2.set("Day")
169         self.date3.set("Year")

```

Figure 8: Code for user signup interface

After pressing register button without any entries

The image shows a web application window titled "Login Form" with a "User Registration" section. The registration form contains several input fields: "Full Name", "User Name", "Password", "Address", "Contact Number", "Gender", "Age", and "Date of Birth" (which is split into "Month", "Day", and "Year" dropdowns). At the bottom are "Register" and "Reset" buttons. An error message dialog box is displayed over the "Contact Number" field, titled "Empty string" with a red 'X' icon and the text "All entries must be filled". The dialog has an "OK" button.

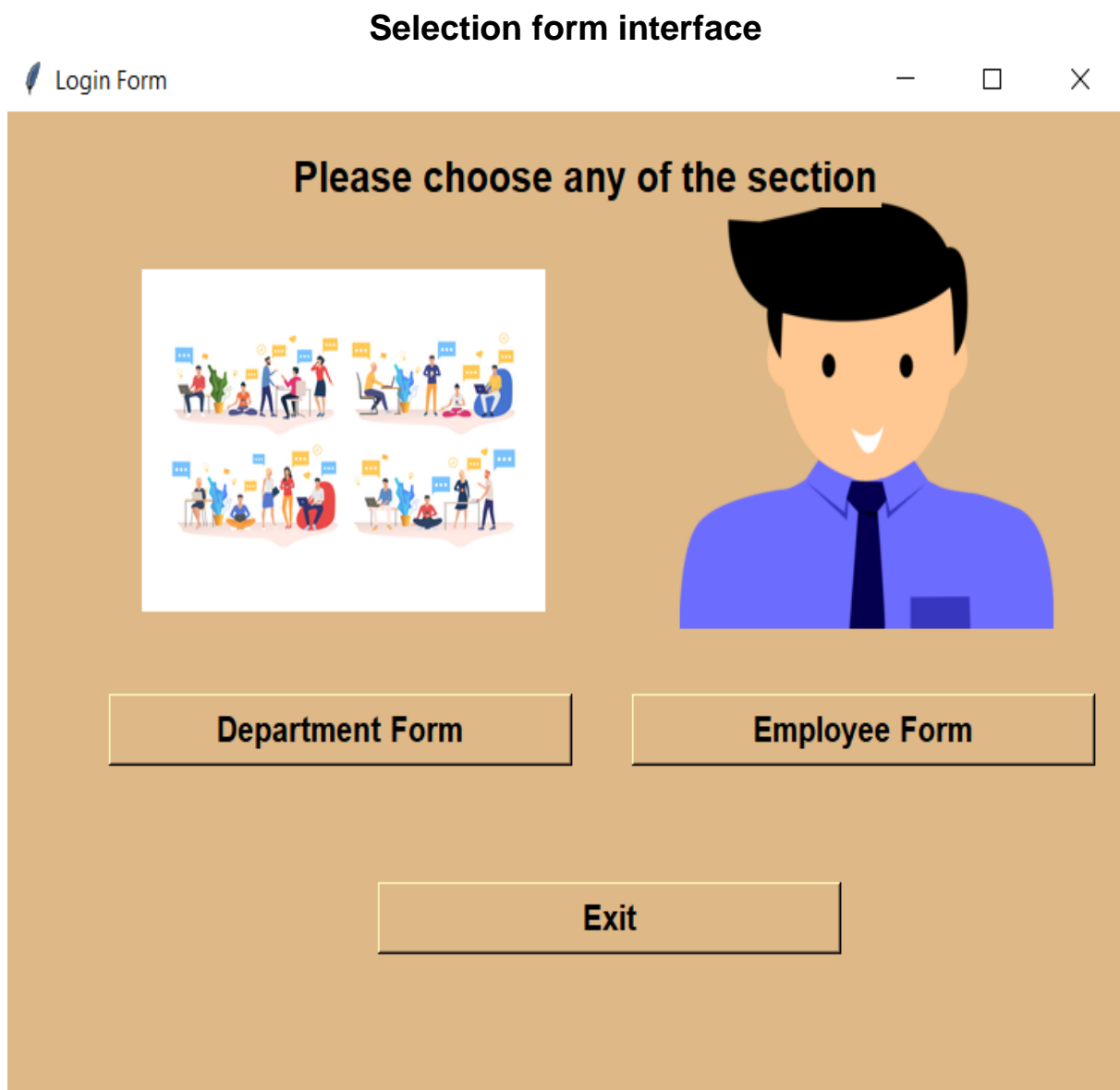
Figure 9: Clicked register button without entries



After register button clicked filling all entries

The image shows a web application window titled "Login Form". The main content area has a light brown background. At the top, it says "User Login" in bold black text. Below this is a large orange circle containing a stylized illustration of a person in a suit. Underneath the circle are two input fields labeled "User Name" and "Password". Below these fields are three large, light brown buttons with black borders, labeled "Log In", "Sign up", and "Reset". At the bottom, there is a link that says "Forgot Password?". A small white dialog box is open in the center-right, titled "Registered" with a close button (X). It contains an information icon (i) and the text "Submission Successful". At the bottom of the dialog is an "OK" button.

Figure 10: Login interface opened after clicking register



*Figure 11: Selection form interface window*

## Selection form interface code


```
select_form.py
1 from tkinter import *
2 from tkinter import messagebox
3 import employ
4 import department
5 import pickle
6
7 class Selection_Form:
8     def __init__(self,window):
9         self.wn=window
10        self.wn.geometry('750x550+20+50')
11
12        #Inserting image
13        self.wn.config(bg='burlywood')
14        self.pic = PhotoImage(file='Department.png')
15        self.labelpic = Label(self.wn, image=self.pic, bg='burlywood')
16        self.labelpic.image = self.pic
17        self.labelpic.place(x=90, y=90)
18
19        self.wn.config(bg='burlywood')
20        self.pic = PhotoImage(file='Employee.png')
21        self.labelpic = Label(self.wn, image=self.pic, bg='burlywood')
22        self.labelpic.image = self.pic
23        self.labelpic.place(x=450, y=50)
24
25        #Label for interface
26        lb=Label(self.wn,text='Please choose any of the section',font=('arial',15,'bold'),bg='burlywood',fg='black')
27        lb.place(x=190,y=20)
28
29        #Buttons
30        self.btn_emp = Button(self.wn, text='Employee Form', font=('arial', 12, 'bold'), fg='black', width=25, bg='burlywood',command=self.employee)
31        self.btn_emp.place(x=420, y=340)
32
33        self.btn_dep = Button(self.wn, text='Department Form', font=('arial', 12, 'bold'), fg='black', width=25, bg='burlywood',command=self.depart)
34        self.btn_dep.place(x=70, y=340)
35
36        self.btn_ex = Button(self.wn, text='Exit', font=('arial', 12, 'bold'), fg='black', width=25, bg='burlywood',command=self.exit)
37        self.btn_ex.place(x=250,y=450)
38
39        #Function for opening the Employee registration form inetrface
40        def employee(self):
41            self.empscr = Toplevel(self.wn)
42            employ.Employee_Form(self.empscr)
43
44        #Function for opening the interface for Department registration form
45        def depart(self):
46            self.depscr = Toplevel(self.wn)
47            department.Department_Registration_Form(self.depscr)
48
49        #Function for exit button
50        def exit(self):
51            self.wn.destroy()
```

Figure 12: Code for selection form interface

## Employee registration interface

Login Form

### Employee Registration Form



**Name**

**Age**

**Address**

**Contact no.**

**Department**

**Id**

**Register**

**Reset**

**View**

Figure 13: Employee registration form interface

## Employment form interface code

```
employ.py x
1 from tkinter import*
2 from tkinter import messagebox
3 import pickle
4 import os
5 import viiew
6
7 class Employee_Form:
8     def __init__(self,window):
9         self.wn=window
10        self.wn.geometry('650x850+20+50')
11
12        self.name_value=StringVar()
13        self.age_value=StringVar()
14        self.add_value=StringVar()
15        self.id_value=StringVar()
16        self.contact_value=StringVar()
17        self.department_value=StringVar()
18
19        self.wn.config(bg='burlywood')
20        self.pic = PhotoImage(file='Employee.png')
21        self.labelpic = Label(self.wn, image=self.pic, bg='burlywood')
22        self.labelpic.image = self.pic
23        self.labelpic.place(x=180, y=60)
24
25
26        self.lb_heading=Label(self.wn,text='Employee Registration Form',bg='black',fg='white',font=('arial',15,'bold'))
27        self.lb_heading.place(relx=0,relx=0,relwidth=1)
28
29        #Label and entries for fields
30        self.lb_name = Label(self.wn, text='Name', font=('arial', 14, 'bold'), fg='black', bg='burlywood')
31        self.lb_name.place(relx=0.1,relx=0.4)
32
33        self.ent_name = Entry(self.wn,font=('arial', 14, 'bold'), fg='black',textvariable=self.name_value, bg='burlywood')
34        self.ent_name.place(relx=0.45,relx=0.4)
35
36
37        self.lb_age= Label(self.wn, text='Age', font=('arial', 14, 'bold'), fg='black', bg='burlywood')
38        self.lb_age.place(relx=0.1,relx=0.45)
39
40        self.ent_age = Entry(self.wn, font=('arial', 14, 'bold'), fg='black', textvariable=self.age_value, bg='burlywood')
41        self.ent_age.place(relx=0.45, relx=0.45)
42
43        self.lb_add= Label(self.wn, text='Address', font=('arial', 14, 'bold'), fg='black', bg='burlywood')
44        self.lb_add.place(relx=0.1, relx=0.5)
45
46        self.ent_add = Entry(self.wn,font=('arial', 14, 'bold'), fg='black',textvariable=self.add_value, bg='burlywood')
47        self.ent_add.place(relx=0.45,relx=0.5)
48
49        self.lb_contact = Label(self.wn, text='Contact no.', font=('arial', 14, 'bold'), fg='black', bg='burlywood')
50        self.lb_contact.place(relx=0.1, relx=0.55)
51
52        self.ent_contact = Entry(self.wn, font=('arial', 14, 'bold'), fg='black', textvariable=self.contact_value, bg='burlywood')
53        self.ent_contact.place(relx=0.45, relx=0.55)
54
55        self.lb_dep =Label(self.wn, text='Department',font=('arial',14,'bold'), fg='black', bg='burlywood')
56        self.lb_dep.place(relx=0.1,relx=0.6)
57
58        self.ent_dep = Entry(self.wn, font=('arial', 14, 'bold'), fg='black', textvariable=self.department_value, bg='burlywood')
59        self.ent_dep.place(relx=0.45, relx=0.6)
60
61        self.lb_id = Label(self.wn, text='Id', font=('arial', 14, 'bold'), fg='black', bg='burlywood')
62        self.lb_id.place(relx=0.1, relx=0.65)
63
64        self.ent_id = Entry(self.wn, font=('arial', 14, 'bold'), fg='black', textvariable=self.id_value, bg='burlywood')
65        self.ent_id.place(relx=0.45, relx=0.65)
```

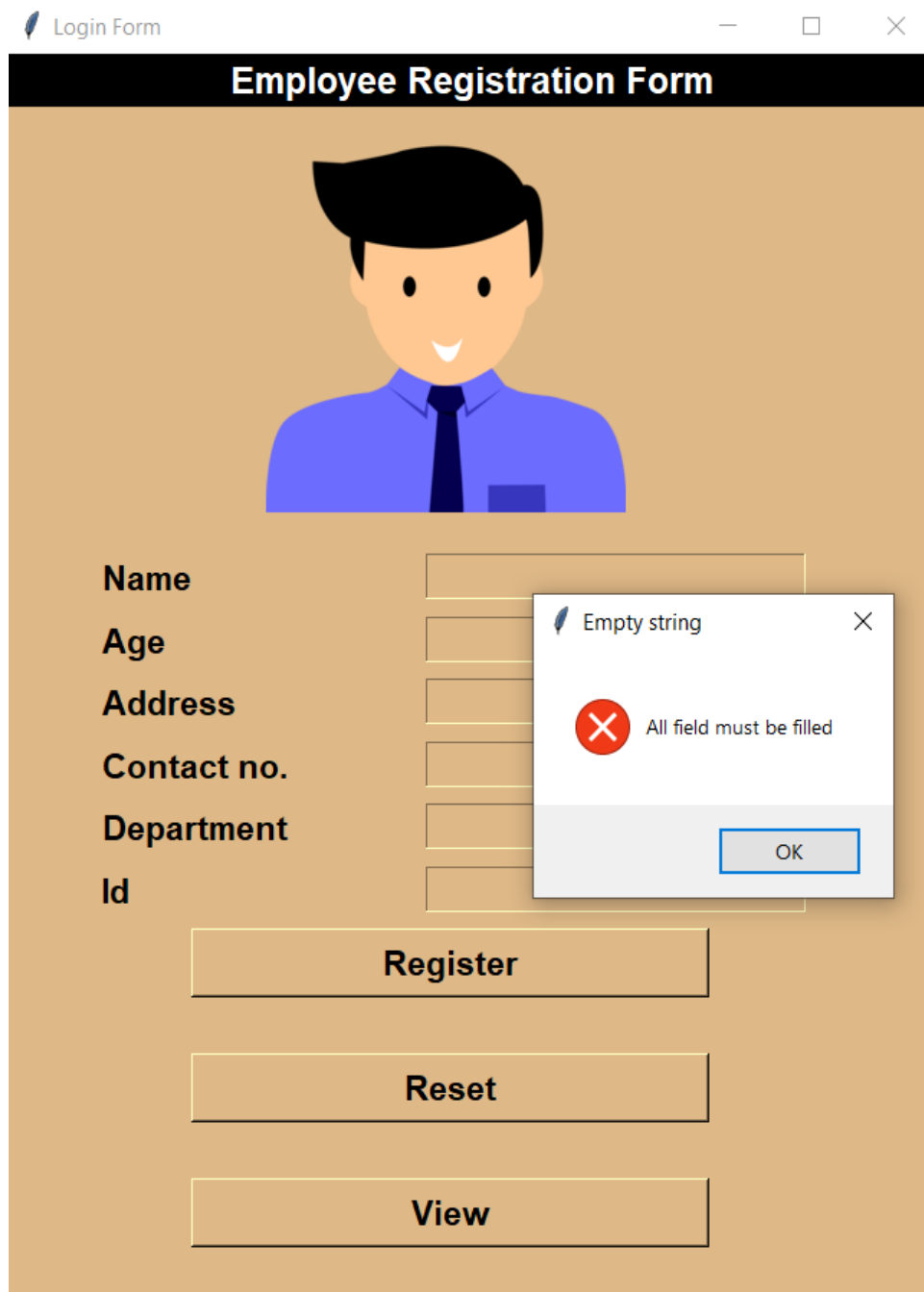
```

66         #Buttons
67         self.btn_register=Button(self.wn,text='Register', font=('arial', 14, 'bold'), fg='black',width=25,command=self.btn_register_click, bg='burlywood')
68         self.btn_register.place(relx=0.2,relx=0.70)
69
70         self.btn_reset = Button(self.wn, text='Reset', font=('arial', 14, 'bold'), fg='black',width=25,command=self.reset, bg='burlywood')
71         self.btn_reset.place(relx=0.2,relx=0.8)
72
73         self.btn_view = Button(self.wn, text='View', font=('arial', 14, 'bold'), fg='black',width=25, bg='burlywood',command=self.view)
74         self.btn_view.place(relx=0.2,relx=0.9)
75
76     def btn_register_click(self):
77         self.insert()
78     def insert(self):
79         global e
80         name=self.name_value.get()
81         age=self.age_value.get()
82         add=self.add_value.get()
83         id=self.id_value.get()
84         cont=self.contact_value.get()
85         dep=self.department_value.get()
86
87         allitems = [name, id, age, add, cont, dep]
88         mylist = [name, id, age, add, cont, dep]
89         ei = {name: mylist}
90         try:
91             for i in allitems:
92                 if len(i) == 0:
93                     raise ValueError("Empty string")
94         except ValueError as e:
95             messagebox.showerror(e, "All field must be filled")
96         else:
97             le = os.path.getsize("D:\\Assignment_files\\file.txt")
98             if le > 0:
99                 f = open("file.txt", "rb+")
100
101                 e = pickle.load(f)
102                 e.update(ei)
103                 print(e)
104                 f.seek(0)
105                 pickle.dump(e, f)
106                 messagebox.showinfo('Successful', 'Submission Successful')
107                 f.close()
108             else:
109                 f = open("file.txt", "wb")
110                 e.update(ei)
111                 print(e)
112                 pickle.dump(e, f)
113                 messagebox.showinfo('Successful', 'Submission Successful')
114                 f.close()
115
116         #Function enabling view button in another interface
117     def view(self):
118         self.viewscr = Toplevel(self.wn)
119         viiew.Employee007_Form(self.viewscr)
120
121         #Function for reset button
122     def reset(self):
123         self.nam.set("")
124         self.id.set("")
125         self.age.set("")
126         self.address.set("")
127         self.contact.set("")
128         self.department.set("Choose your Department.")

```

Figure 14:Employment form interface code

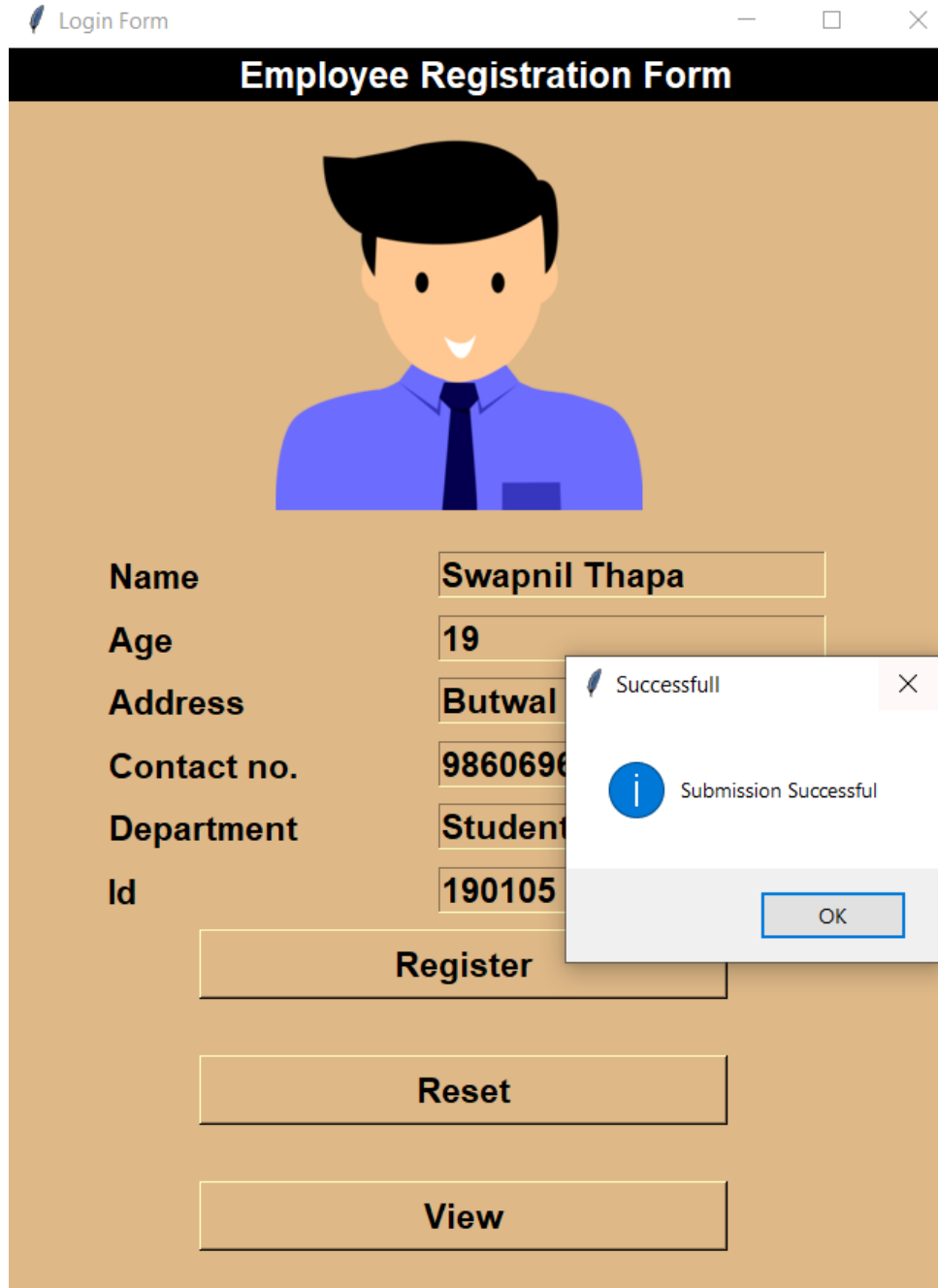
After clicking register button with no entries



The image shows a web application window titled "Login Form" with a standard window control bar (minimize, maximize, close). The main content area has a black header bar with the text "Employee Registration Form" in white. Below the header is a large orange background area. At the top of this area is a cartoon illustration of a man with black hair, wearing a blue shirt and a dark tie. Below the illustration are six input fields, each with a label to its left: "Name", "Age", "Address", "Contact no.", "Department", and "Id". All input fields are empty. Below the input fields are three large, light brown buttons with black text: "Register", "Reset", and "View". A modal dialog box is open over the "Register" button. The dialog has a title bar with a feather icon and the text "Empty string", and a close button (X). The main area of the dialog contains a red circle with a white "X" icon and the text "All field must be filled". At the bottom of the dialog is an "OK" button.

Figure 15: Employee registration interface with no entries

After clicking register button with filled entries



The image shows a web application window titled "Login Form" with standard window controls. The main content area is titled "Employee Registration Form" and features a cartoon illustration of a man in a blue shirt and tie. Below the illustration is a registration form with the following fields and values:

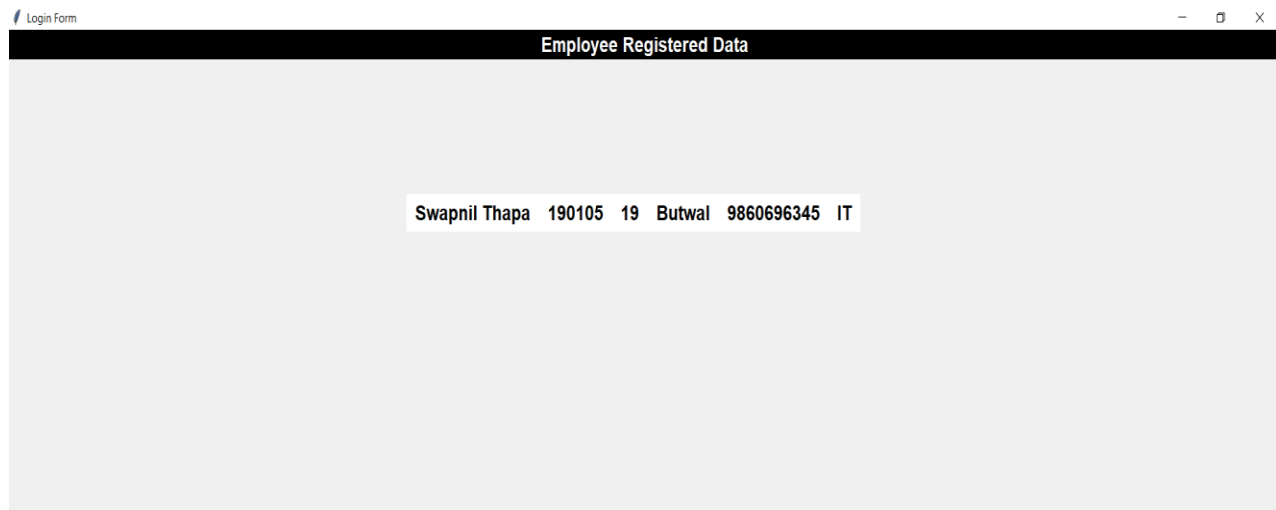
Field	Value
Name	Swapnil Thapa
Age	19
Address	Butwal
Contact no.	9860696
Department	Student
Id	190105

Below the form are three buttons: "Register", "Reset", and "View". A modal dialog box is open over the "Register" button, titled "Successfull" (with a typo), displaying an information icon and the text "Submission Successful". The dialog has an "OK" button at the bottom right.

Figure 16: Employee registration form with filled entries



After clicking view button in Employee registration form



The screenshot shows a web application window with a title bar containing 'Login Form' and standard window controls. The main content area has a black header bar labeled 'Employee Registered Data'. Below this, a table displays the details of a registered employee.

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

```
department.py x
1 from tkinter import *
2 from tkinter import messagebox
3 import os
4 import pickle
5 class Department_Registration_Form:
6     def __init__(self,window):
7         self.wn=window
8         self.wn.title('Department Form')
9         self.wn.geometry('560x750+40+50')
10
11         self.wn.config(bg='burlywood')
12         self.pic = PhotoImage(file='dep.png')
13         self.labelpic = Label(self.wn, image=self.pic, bg='burlywood')
14         self.labelpic.image = self.pic
15         self.labelpic.place(x=140, y=50)
16
17         self.dname_value = StringVar()
18         self.dcode_value = StringVar()
19         self.dloc_value = StringVar()
20         self.did_value=StringVar()
21
22         self.lb_heading=Label(self.wn,text='Department Registration Form',font=('arial',15,'bold'),fg='Black', bg='burlywood')
23         self.lb_heading.place(x=120,y=20)
24
25         self.lb_dname = Label(self.wn, text='Department Name', font=('arial', 12, 'bold'), fg='black', bg='burlywood')
26         self.lb_dname.place(x=50, y=300)
27
28         self.ent_dname = Entry(self.wn, font=('arial', 12, 'bold'), bg='burlywood',textvariable=self.dname_value)
29         self.ent_dname.place(x=280, y=300)
30
31         self.lb_dcode = Label(self.wn, text='Department Code', font=('arial', 12, 'bold'), fg='black', bg='burlywood')
32         self.lb_dcode.place(x=50, y=370)
33
34         self.ent_dcode = Entry(self.wn, font=('arial', 12, 'bold'), bg='burlywood',textvariable=self.dcode_value)
35         self.ent_dcode.place(x=280, y=370)
36
37         self.lb_id = Label(self.wn, text='Department ID', font=('arial', 12, 'bold'), fg='black', bg='burlywood')
38         self.lb_id.place(x=50, y=440)
39
40         self.ent_id= Entry(self.wn,font=('arial',12,'bold'),bg='burlywood', textvariable=self.did_value)
41         self.ent_id.place(x=280, y=440)
42
43         self.lb_loc = Label(self.wn, text='Department Location', font=('arial', 12, 'bold'), fg='black', bg='burlywood')
44         self.lb_loc.place(x=50, y=510)
45
46         self.ent_loc = Entry(self.wn, font=('arial', 12, 'bold'), bg='burlywood', textvariable=self.dloc_value)
47         self.ent_loc.place(x=280, y=510)
48
49         self.btn_register = Button(self.wn, text='Register', font=('arial', 12, 'bold'), fg='black', width=25, command=self.btn_register_click, bg='burlywood')
50         self.btn_register.place(x=130, y=580)
51
52         self.btn_reset = Button(self.wn, text='Reset', font=('arial', 12, 'bold'), fg='black', width=25,bg='burlywood')
53         self.btn_reset.place(x=130,y=650)
54
55         self.btn_exit = Button(self.wn, text='Exit', font=('arial', 12, 'bold'), fg='black', width=25, bg='burlywood')
56         self.btn_exit.place(x=130, y=650)
57
58     def btn_register_click(self):
59         self.insert()
```

```

61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91

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

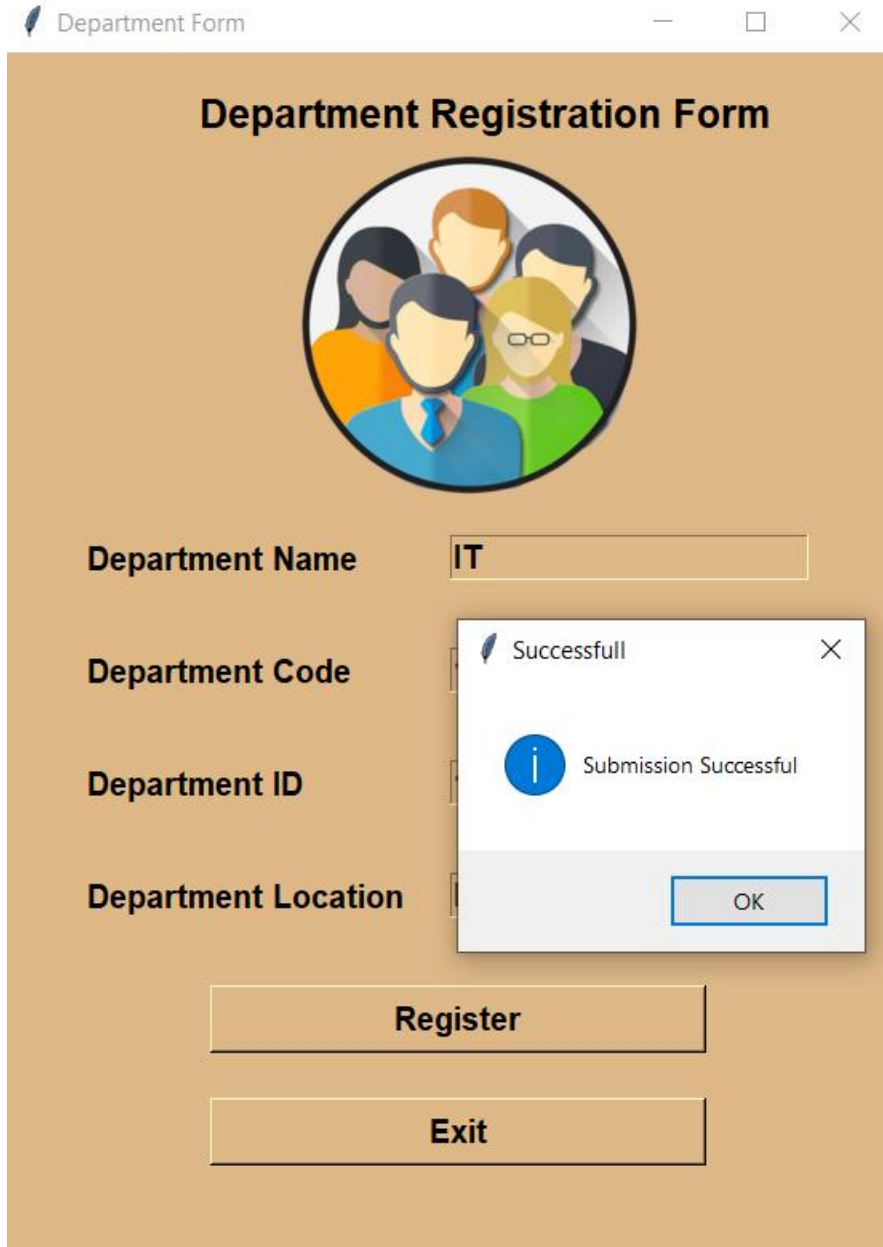
93
94
95
96
97

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



The image shows a software window titled "Department Form" with a standard Windows title bar (minimize, maximize, close buttons). The main content area has a tan background and is titled "Department Registration Form" in bold black text. Below the title is a circular icon depicting a group of five stylized people. The form contains four labeled input fields: "Department Name" (containing "IT"), "Department Code", "Department ID", and "Department Location". At the bottom of the form are two large, light-brown buttons labeled "Register" and "Exit". A small, white modal dialog box is overlaid on the form, titled "Successfull" (with a typo). It features a blue information icon and the text "Submission Successful". An "OK" button is located at the bottom right of the dialog box.

Figure 20: Department interface form with filled entries.

After clicking register with no entries

The image shows a software window titled "Department Form". Inside, there is a "Department Registration Form" with a circular icon of four people. Below the icon are four labels: "Department Name", "Department Code", "Department ID", and "Department Location", each followed by an empty text input field. At the bottom are two buttons: "Register" and "Exit". A modal dialog box is open in the center, titled "Empty string" with a close button. It contains a red "X" icon and the text "All field must be filled". An "OK" button is at the bottom of the dialog.

Figure 21: Department interface form with no entries

## View interface code

```
1 from tkinter import*
2 import pickle
3 import os
4
5 class Employee007_Form:
6     def __init__(self,window):
7         self.wn=window
8         self.wn.geometry('650x850+20+50')
9
10        self.name_value=StringVar()
11        self.age_value=StringVar()
12        self.add_value=StringVar()
13        self.id_value=StringVar()
14        self.contact_value=StringVar()
15        self.department_value=StringVar()
16        # creating frame to accomodate objects/frame1
17        self.frame2 = Frame(self.wn, bg="white", height=2, width=40)
18        self.frame2.place(x=600, y=200)
19
20
21        self.lb_heading=Label(self.wn,text='Employee Registered Data',bg='black',fg='white',font=('arial',15,'bold'))
22        self.lb_heading.place(relx=0,relx=0,relwidth=1)
23
24        self.load()
25        self.bb=10
26        self.length_x = len(x)
27        print(self.length_x)
28        self.null= -1
29        self.va=-1
30
31    for ff in range(round(len(x) / 6)):
32        if ff == 0:
33            self.dat_prt()
34        elif ff >= 1:
35            self.va = self.null +(4*ff)+(6*ff)
36            print(self.va)
37            self.bb = self.bb + 5
38            self.dat_prt()
39    def dat_prt(self):
40        self.lb_name_val = Label(self.frame2, text=x[self.va+1], fg="black", bg="white", font=("Arial", 15, "bold"))
41        self.lb_name_val.grid(row=self.bb, column=0, padx=10, pady=5)
42
43        self.lb_id_val = Label(self.frame2, text=x[self.va + 2], fg="black", bg="white", font=("Arial", 15, "bold"))
44        self.lb_id_val.grid(row=self.bb, column=1, padx=10, pady=5)
45
46        self.lb_Address_val = Label(self.frame2, text=x[self.va + 3], fg="black", bg="white", font=("Arial", 15, "bold"))
47        self.lb_Address_val.grid(row=self.bb, column=2, padx=10, pady=5)
48
49        self.lb_Department_val = Label(self.frame2, text=x[self.va + 4], fg="black",bg="white", font=("Arial", 15, "bold"))
50        self.lb_Department_val.grid(row=self.bb, column=3, padx=10, pady=5)
51
52        self.lb_AGE_val = Label(self.frame2, text=x[self.va + 5], fg="black",bg="white", font=("Arial", 15, "bold"))
53        self.lb_AGE_val.grid(row=self.bb, column=4, padx=10, pady=5)
54
55        self.lb_CONTACT_val = Label(self.frame2, text=x[self.va + 6], fg="black",bg="white", font=("Arial", 15, "bold"))
56        self.lb_CONTACT_val.grid(row=self.bb, column=5, padx=10, pady=5)
```

```

58     def load(self):
59         global x
60         le = os.path.getsize("D:\\Assignment_files\\file.txt")
61         if le > 0:
62             f = open("file.txt", "rb")
63             lod = pickle.load(f)
64             print(lod)
65             d = ""
66             for i in lod.values():
67                 if type(i) == list:
68                     for k in i:
69                         d = d + k + "\t"
70             x = d.split("\t")
71             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](https://www.tutorialspoint.com/python/python_gui_programming.htm)

[Accessed 2020].