**PYTHON PROJECT**

**Online Recruitment System**



Submitted by: Aman Suryawanshi(roll-18)

Pir Asrar(roll-19)

Tarikul Islam(roll-21)

Section: K18JF

**Table of contents**

**1.** Introduction

**2.** Functional requirement

3. Non functional requirement

4. Source code

5. Screenshots of program.

**Introduction**

**Purpose**

This document describe the software requirement and specification for **Registering and applying for jobs**. The application is intended to provide complete solution for customers through it.

**Overview**

An application that will help users to find their desired jobs.

**Definition**

Recruitment system is the system by which any user can get their desired jobs in their desired field.

**Users**

This document is intended for all users and developers (designer, testers, maintainers). And also for admin of the application who look for details of the application.

**Scope**

This system allow the customer to get jobs as per their choice.

**Functional requirement**

This section provides requirement overview of the system. Various functional modules that can be implemented by the system will be –

1. Description

**Registration:**

If customer wants to use the application, then user must register in it.

Input: Enter username, password.

Output: Users can now log in.

Processing: A new account is created for the user to login.

2. Log in

Customer logins to system by entering valid user id and password for using it..

Input: Enter username and password.

Output: Logged in

Processing: check if the id and password is valid or not.

3. Find Jobs

In this module the user will set an alarm

Input: Press the Apply for jobs

Output: Find your desired field’s job

Processing: The system will help to register in their desired field as per input given by user.

**Non functional requirement**

**1. Security**

Sensitive data should be encrypted before sending over insecure path like internet.

**2. Reliability**

Backup of data which is continuously maintain and updated regularly for changes.

**3. Availability**

The system should be available at all times, meaning the user can access it using an

application.

4. **Portability**

An end-user can use this on any OS, either it is Windows or Linux or IOS.

The system should run on PC, Laptops, mobile, phones and tablets etc.

**Source code**

**Main page:**

**import os**

**from subprocess import call**

**import sys**

**try:**

**from Tkinter import \***

**except ImportError:**

**from tkinter import \***

**try:**

**import ttk**

**py3 = False**

**except ImportError:**

**import tkinter.ttk as ttk**

**py3 = True**

**def click\_login():**

**call(["python", "login.py"])**

**def click\_regis():**

**call(["python", "regis.py"])**

**def click\_find():**

**call(["python","projectpython.py"])**

**class Online\_Employee\_Recruitment\_System:**

**def \_\_init\_\_(self):**

**root = Tk()**

**'''This class configures and populates the toplevel window.**

**top is the toplevel containing window.'''**

**\_bgcolor = '#d9d9d9'**

**\_fgcolor = '#000000'**

**\_compcolor = '#ffffff'**

**\_ana1color = '#ffffff'**

**\_ana2color = '#ffffff'**

**font14 = "-family {Segoe UI} -size 15 -weight bold -slant " \**

**"roman -underline 0 -overstrike 0"**

**font16 = "-family {Swis721 BlkCn BT} -size 40 -weight bold " \**

**"-slant roman -underline 0 -overstrike 0"**

**font9 = "-family {Segoe UI} -size 9 -weight normal -slant " \**

**"roman -underline 0 -overstrike 0"**

**root.geometry("963x749+540+110")**

**root.title("Employee Recruitment System")**

**root.configure(background="#d9d9d9")**

**root.configure(highlightbackground="#d9d9d9")**

**root.configure(highlightcolor="black")**

**self.Frame1 = Frame(root)**

**self.Frame1.place(relx=0.02, rely=0.03, relheight=0.94, relwidth=0.96)**

**self.Frame1.configure(relief=GROOVE)**

**self.Frame1.configure(background="#d9d9d9")**

**self.Frame1.configure(highlightbackground="#d9d9d9")**

**self.Frame1.configure(highlightcolor="black")**

**self.Frame1.configure(width=925)**

**self.Message6 = Message(self.Frame1)**

**self.Message6.place(relx=0.09, rely=0.01, relheight=0.15, relwidth=0.86)**

**self.Message6.configure(background="#d9d9d9")**

**self.Message6.configure(font=font16)**

**self.Message6.configure(foreground="#000000")**

**self.Message6.configure(highlightbackground="#d9d9d9")**

**self.Message6.configure(highlightcolor="black")**

**self.Message6.configure(text='''WELCOME''')**

**self.Message6.configure(width=791)**

**self.Button2 = Button(self.Frame1)**

**self.Button2.place(relx=0.18, rely=0.17, height=103, width=566)**

**self.Button2.configure(activebackground="#d9d9d9")**

**self.Button2.configure(activeforeground="#000000")**

**self.Button2.configure(background="#d9d9d9")**

**self.Button2.configure(disabledforeground="#bfbfbf")**

**self.Button2.configure(font=font14)**

**self.Button2.configure(foreground="#000000")**

**self.Button2.configure(highlightbackground="#d9d9d9")**

**self.Button2.configure(highlightcolor="black")**

**self.Button2.configure(pady="0")**

**self.Button2.configure(text='''1.Login''')**

**self.Button2.configure(width=566)**

**self.Button2.configure(command=click\_login)**

**self.Button3 = Button(self.Frame1)**

**self.Button3.place(relx=0.18, rely=0.33, height=93, width=566)**

**self.Button3.configure(activebackground="#ff6600")**

**self.Button3.configure(background="#d9d9d9")**

**self.Button3.configure(disabledforeground="#bfbfbf")**

**self.Button3.configure(font=font14)**

**self.Button3.configure(text='''2.Register Yourself''')**

**self.Button3.configure(command=click\_regis)**

**self.Button4 = Button(self.Frame1)**

**self.Button4.place(relx=0.18, rely=0.48, height=90, width=566)**

**self.Button4.configure(activebackground="#ff6600")**

**self.Button4.configure(background="#d9d9d9")**

**self.Button4.configure(disabledforeground="#bfbfbf")**

**self.Button4.configure(font=font14)**

**self.Button4.configure(text='''3.Apply for Job''')**

**self.Button4.configure(command=click\_find)**

**if \_\_name\_\_ == '\_\_main\_\_':**

**GUUEST=Online\_Employee\_Recruitment\_System()**

**Log in:**

**from tkinter import \***

**import os**

**# Designing window for registration**

**def register():**

**global register\_screen**

**register\_screen = Toplevel(main\_screen)**

**register\_screen.title("Register")**

**register\_screen.geometry("500x500")**

**global username**

**global password**

**global username\_entry**

**global password\_entry**

**username = StringVar()**

**password = StringVar()**

**Label(register\_screen, text="Please enter details below", bg="blue").pack()**

**Label(register\_screen, text="").pack()**

**username\_lable = Label(register\_screen, text="Username \* ")**

**username\_lable.pack()**

**username\_entry = Entry(register\_screen, textvariable=username)**

**username\_entry.pack()**

**password\_lable = Label(register\_screen, text="Password \* ")**

**password\_lable.pack()**

**password\_entry = Entry(register\_screen, textvariable=password, show='\*')**

**password\_entry.pack()**

**Label(register\_screen, text="").pack()**

**Button(register\_screen, text="Register", width=10, height=1, bg="blue", command = register\_user).pack()**

**# Implementing event on register button**

**def register\_user():**

**username\_info = username.get()**

**password\_info = password.get()**

**file = open(username\_info, "w")**

**file.write(username\_info + "\n")**

**file.write(password\_info)**

**file.close()**

**username\_entry.delete(0, END)**

**password\_entry.delete(0, END)**

**Label(register\_screen, text="Registration Success", fg="green", font=("calibri", 11)).pack()**

**# Designing window for login**

**def login():**

**global login\_screen**

**login\_screen = Toplevel(main\_screen)**

**login\_screen.title("Login")**

**login\_screen.geometry("500x500")**

**Label(login\_screen, text="Please enter details below to login").pack()**

**Label(login\_screen, text="").pack()**

**global username\_verify**

**global password\_verify**

**username\_verify = StringVar()**

**password\_verify = StringVar()**

**global username\_login\_entry**

**global password\_login\_entry**

**Label(login\_screen, text="Username \* ").pack()**

**username\_login\_entry = Entry(login\_screen, textvariable=username\_verify)**

**username\_login\_entry.pack()**

**Label(login\_screen, text="").pack()**

**Label(login\_screen, text="Password \* ").pack()**

**password\_login\_entry = Entry(login\_screen, textvariable=password\_verify, show= '\*')**

**password\_login\_entry.pack()**

**Label(login\_screen, text="").pack()**

**Button(login\_screen, text="Login", width=10, height=1, command = login\_verify).pack()**

**# Implementing event on login button**

**def login\_verify():**

**username1 = username\_verify.get()**

**password1 = password\_verify.get()**

**username\_login\_entry.delete(0, END)**

**password\_login\_entry.delete(0, END)**

**list\_of\_files = os.listdir()**

**if username1 in list\_of\_files:**

**file1 = open(username1, "r")**

**verify = file1.read().splitlines()**

**if password1 in verify:**

**login\_sucess()**

**else:**

**password\_not\_recognised()**

**else:**

**user\_not\_found()**

**# Designing popup for login success**

**def login\_sucess():**

**global login\_success\_screen**

**login\_success\_screen = Toplevel(login\_screen)**

**login\_success\_screen.title("Success")**

**login\_success\_screen.geometry("150x100")**

**Label(login\_success\_screen, text="Login Success").pack()**

**Button(login\_success\_screen, text="OK", command=delete\_login\_success).pack()**

**# Designing popup for login invalid password**

**def password\_not\_recognised():**

**global password\_not\_recog\_screen**

**password\_not\_recog\_screen = Toplevel(login\_screen)**

**password\_not\_recog\_screen.title("Success")**

**password\_not\_recog\_screen.geometry("150x100")**

**Label(password\_not\_recog\_screen, text="Invalid Password ").pack()**

**Button(password\_not\_recog\_screen, text="OK", command=delete\_password\_not\_recognised).pack()**

**# Designing popup for user not found**

**def user\_not\_found():**

**global user\_not\_found\_screen**

**user\_not\_found\_screen = Toplevel(login\_screen)**

**user\_not\_found\_screen.title("Success")**

**user\_not\_found\_screen.geometry("150x100")**

**Label(user\_not\_found\_screen, text="User Not Found").pack()**

**Button(user\_not\_found\_screen, text="OK", command=delete\_user\_not\_found\_screen).pack()**

**# Deleting popups**

**def delete\_login\_success():**

**login\_success\_screen.destroy()**

**def delete\_password\_not\_recognised():**

**password\_not\_recog\_screen.destroy()**

**def delete\_user\_not\_found\_screen():**

**user\_not\_found\_screen.destroy()**

**#------------------------------------------------------------**

**# Designing Main(first) window**

**def main\_account\_screen():**

**global main\_screen**

**main\_screen = Tk()**

**main\_screen.geometry("500x550")**

**main\_screen.title("Account Login")**

**Label(text="Select Your Choice", bg="grey", width="300", height="2", font=("Calibri", 13)).pack()**

**Label(text="").pack()**

**Button(text="Login", height="2", width="30", command = login).pack()**

**Label(text="").pack()**

**Button(text="Register", height="2", width="30", command=register).pack()**

**main\_screen.mainloop()**

**main\_account\_screen()**

**Register Yourself:**

**from tkinter import \***

**import sqlite3**

**def sel():**

**selection =str(var.get())**

**label.config(text=selection)**

**root = Tk()**

**root.geometry('500x500')**

**root.title("Registration Form")**

**Fullname=StringVar()**

**Email=StringVar()**

**var = IntVar()**

**c=StringVar()**

**text=StringVar()**

**def database():**

**name1=Fullname.get()**

**email=Email.get()**

**gender=var.get()**

**country=c.get()**

**describe=text.get()**

**conn = sqlite3.connect('Form.db')**

**with conn:**

**cursor=conn.cursor()**

**cursor.execute('CREATE TABLE IF NOT EXISTS Student (Fullname TEXT,Email TEXT,Gender TEXT,country TEXT,describe TEXT)' )**

**cursor.execute('INSERT INTO Student (FullName,Email,Gender,country,Programming) VALUES(?,?,?,?,?)',(name1,email,gender,country,describe))**

**conn.commit()**

**call(["python", "regis.py"])**

**label\_0 = Label(root, text="Registration form",width=20,font=("bold", 20))**

**label\_0.place(x=90,y=53)**

**label\_1 = Label(root, text="FullName",width=20,font=("bold", 10))**

**label\_1.place(x=80,y=130)**

**entry\_1 = Entry(root,textvar=Fullname)**

**entry\_1.place(x=240,y=130)**

**label\_2 = Label(root, text="Email",width=20,font=("bold", 10))**

**label\_2.place(x=68,y=180)**

**entry\_2 = Entry(root,textvar=Email)**

**entry\_2.place(x=240,y=180)**

**label\_3 = Label(root, text="Gender",width=20,font=("bold", 10))**

**label\_3.place(x=70,y=230)**

**R1= Radiobutton(root, text="Male",padx = 5, variable=var, value=1,command=sel).place(x=235,y=230)**

**Radiobutton(root, text="Female",padx = 20, variable=var, value=2).place(x=290,y=230)**

**label\_4 = Label(root, text="country",width=20,font=("bold", 10))**

**label\_4.place(x=70,y=280)**

**list1 = ['INDIA','USA','UK','RUSSIA','CHINA','SOUTH AFRICA'];**

**droplist=OptionMenu(root,c, \*list1)**

**droplist.config(width=15)**

**c.set('select your country')**

**droplist.place(x=240,y=280)**

**label\_5 = Label(root, text="DescribeYourself",width=20,font=("bold", 10))**

**label\_5.place(x=70,y=330)**

**entry\_5 = Entry(root,textvar=text)**

**entry\_5.place(x=240,y=330)**

**Button(root, text='Submit',width=20,bg='grey',fg='white',command=database).place(x=180,y=400)**

**root.mainloop()**

**Apply For Jobs:**

**from tkinter import \***

**from tkinter import messagebox**

**a=Tk()**

**var=IntVar()**

**checkvar1=IntVar()**

**checkvar2=IntVar()**

**checkvar3=IntVar()**

**def sel():**

**selection='you selected the option'+ str(var.get())**

**label.config(text=selection)**

**def yoo():**

**e=Tk()**

**e.geometry("300x200")**

**l6=Label(e, text="You are done Thanks for applying")**

**l6.grid(row=0,column=7)**

**def done():**

**d=Tk()**

**d.geometry("800x500")**

**l4=Label(d, text="Select language")**

**l4.grid(row=0,column=0,padx=20,pady=30)**

**r2=Radiobutton(d, text='python',variable=var,value=1,command=sel)**

**r2.grid(row=0, column=1)**

**r3=Radiobutton(d, text='HTML',variable=var,value=2,command=sel)**

**r3.grid(row=0, column=2)**

**r4=Radiobutton(d, text='C',variable=var,value=3,command=sel)**

**r4.grid(row=0, column=3)**

**r5=Radiobutton(d, text='C++',variable=var,value=4,command=sel)**

**r5.grid(row=0, column=4)**

**b5=Button(d,text="Apply",command=yoo)**

**b5.grid(row=1,column=2,padx=20,pady=30)**

**b7=Button(d,text="Exit",command=processcancel)**

**b7.grid(row=1,column=4,padx=20,pady=30)**

**label=Label(d)**

**label.grid(row=7,column=4)**

**def apply3():**

**g=Tk()**

**g.geometry("800x800")**

**l4=Label(g, text="Select post")**

**l4.grid(row=0,column=0,padx=20,pady=30)**

**l5=Label(g, text="Assistant professor (salary:100000,Collage:Anna University,Qualification:Phd holder in computer science)")**

**l5.grid(row=1,column=0,padx=20,pady=30)**

**b3=Button(g,text="Next",command=processok)**

**b3.grid(row=2,column=0,padx=20,pady=30)**

**l6=Label(g, text="professor (salary:65000,Collage:Anna University,Qualification:Btech holder in computer science with atleat 80% marks)")**

**l6.grid(row=3,column=0,padx=20,pady=30)**

**b3=Button(g,text="Next",command=processok)**

**b3.grid(row=4,column=0,padx=20,pady=30)**

**l7=Label(g, text="Clerk (salary:35000,Collage:Anna University,Qualification:Passed 12th with atleat 80% marks)")**

**l7.grid(row=5,column=0,padx=20,pady=30)**

**b4=Button(g,text="Next",command=processok)**

**b4.grid(row=6,column=0,padx=20,pady=30)**

**b5=Button(g,text="Exit",command=processcancel)**

**b5.grid(row=7,column=4,padx=20,pady=30)**

**def apply2():**

**f=Tk()**

**f.geometry("800x800")**

**l4=Label(f, text="Select post")**

**l4.grid(row=0,column=0,padx=20,pady=30)**

**l5=Label(f, text="Assistant professor (salary:100000,Collage:Delhi Technological University,Qualification:Phd holder in computer science)")**

**l5.grid(row=1,column=0,padx=20,pady=30)**

**b3=Button(f,text="Next",command=processok)**

**b3.grid(row=2,column=0,padx=20,pady=30)**

**l6=Label(f, text="professor (salary:65000,Collage:Delhi Technological University,Qualification:Btech holder in computer science with atleat 80% marks)")**

**l6.grid(row=3,column=0,padx=20,pady=30)**

**b3=Button(f,text="Next",command=processok)**

**b3.grid(row=4,column=0,padx=20,pady=30)**

**l7=Label(f, text="Clerk (salary:35000,Collage:Delhi Technological University,Qualification:Passed 12th with atleat 80% marks)")**

**l7.grid(row=5,column=0,padx=20,pady=30)**

**b4=Button(f,text="Next",command=processok)**

**b4.grid(row=6,column=0,padx=20,pady=30)**

**b5=Button(f,text="Exit",command=processcancel)**

**b5.grid(row=7,column=4,padx=20,pady=30)**

**def apply():**

**c=Tk()**

**c.geometry("800x800")**

**l4=Label(c, text="Select post")**

**l4.grid(row=0,column=0,padx=20,pady=30)**

**l5=Label(c, text="Assistant professor (salary:100000,Collage:LPU,Qualification:Phd holder in computer science)")**

**l5.grid(row=1,column=0,padx=20,pady=30)**

**b3=Button(c,text="Next",command=processok)**

**b3.grid(row=2,column=0,padx=20,pady=30)**

**l6=Label(c, text="professor (salary:65000,Collage:LPU,Qualification:Btech holder in computer science with atleat 80% marks)")**

**l6.grid(row=3,column=0,padx=20,pady=30)**

**b3=Button(c,text="Next",command=processok)**

**b3.grid(row=4,column=0,padx=20,pady=30)**

**l7=Label(c, text="Clerk (salary:35000,Collage:LPU,Qualification:Passed 12th with atleat 80% marks)")**

**l7.grid(row=5,column=0,padx=20,pady=30)**

**b4=Button(c,text="Next",command=processok)**

**b4.grid(row=6,column=0,padx=20,pady=30)**

**b5=Button(c,text="Exit",command=processcancel)**

**b5.grid(row=7,column=4,padx=20,pady=30)**

**def processok():**

**top=Tk()**

**top.geometry("1000x900")**

**l8=Label(top, text="Select field")**

**l8.grid(row=0,column=0,padx=20,pady=30)**

**l9=Label(top, text="Computer Science")**

**l9.grid(row=1,column=0,padx=20,pady=30)**

**b1=Button(top,text="Next",command=done)**

**b1.grid(row=2,column=3,padx=20,pady=30)**

**l10=Label(top, text="Mechanical")**

**l10.grid(row=3,column=0,padx=20,pady=30)**

**b1=Button(top,text="Apply",command=yoo)**

**b1.grid(row=4,column=3,padx=20,pady=30)**

**l11=Label(top, text='Civil')**

**l11.grid(row=5,column=0,padx=20,pady=30)**

**b1=Button(top,text="Apply",command=yoo)**

**b1.grid(row=6,column=3,padx=20,pady=30)**

**b2=Button(top,text="Exit",command=processcancel)**

**b2.grid(row=7,column=5,padx=20,pady=30)**

**label=Label(top)**

**label.grid(row=7,column=4)**

**def processcancel():**

**b=Tk()**

**b.geometry("300x200")**

**l3=Label(b, text="Thank you for using our window")**

**l3.grid(row=0,column=3)**

**a.title("Find job")**

**a.geometry("900x800")**

**l2=Label(a, text="Select location")**

**l2.grid(row=0,column=0,padx=20,pady=30)**

**l3=Label(a, text='Jalandhar')**

**l3.grid(row=1,column=0,padx=20,pady=30)**

**b1=Button(a,text="Next",command=apply)**

**b1.grid(row=2,column=3,padx=20,pady=30)**

**l4=Label(a, text='Delhi')**

**l4.grid(row=3,column=0,padx=20,pady=30)**

**b1=Button(a,text="Next",command=apply2)**

**b1.grid(row=4,column=3,padx=20,pady=30)**

**l5=Label(a, text='Chennai')**

**l5.grid(row=5,column=0,padx=20,pady=30)**

**b1=Button(a,text="Next",command=apply3)**

**b1.grid(row=6,column=3,padx=20,pady=30)**

**b2=Button(a,text="Exit",command=processcancel)**

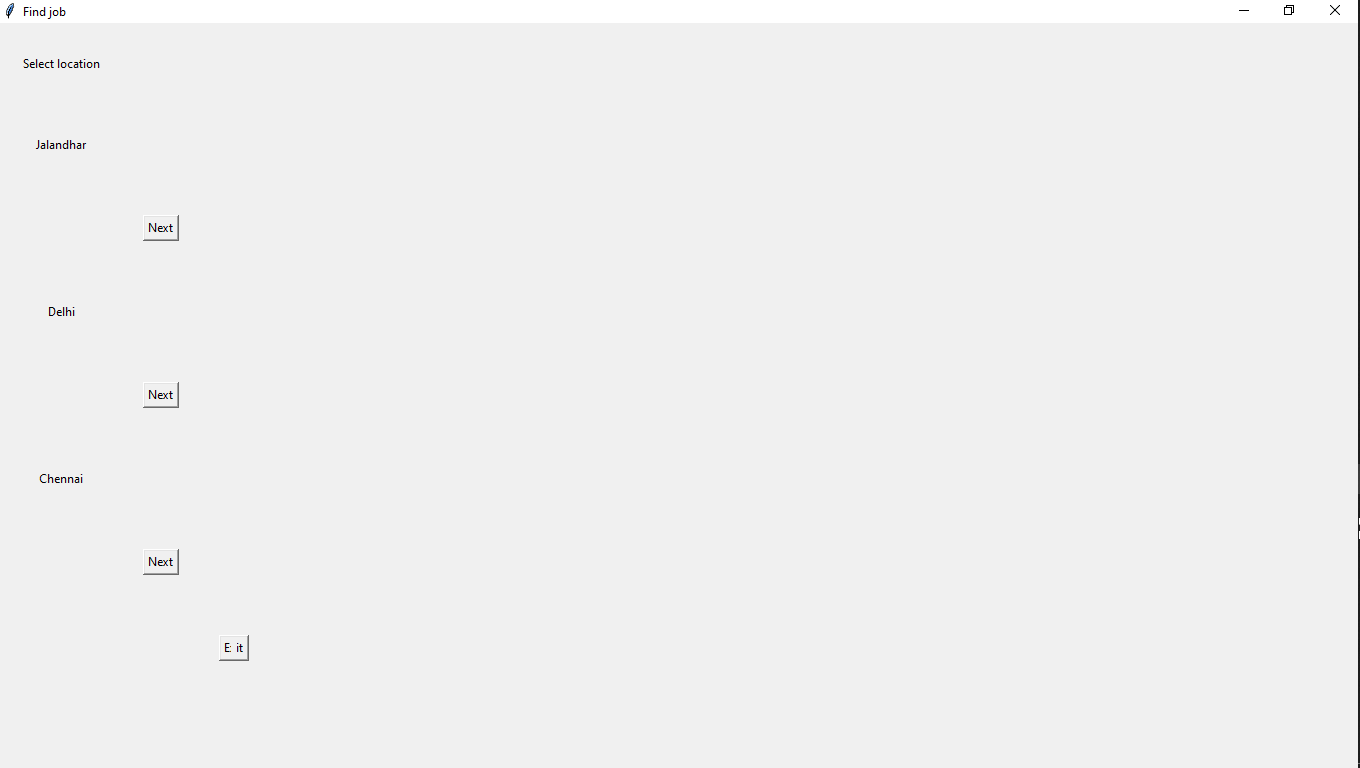
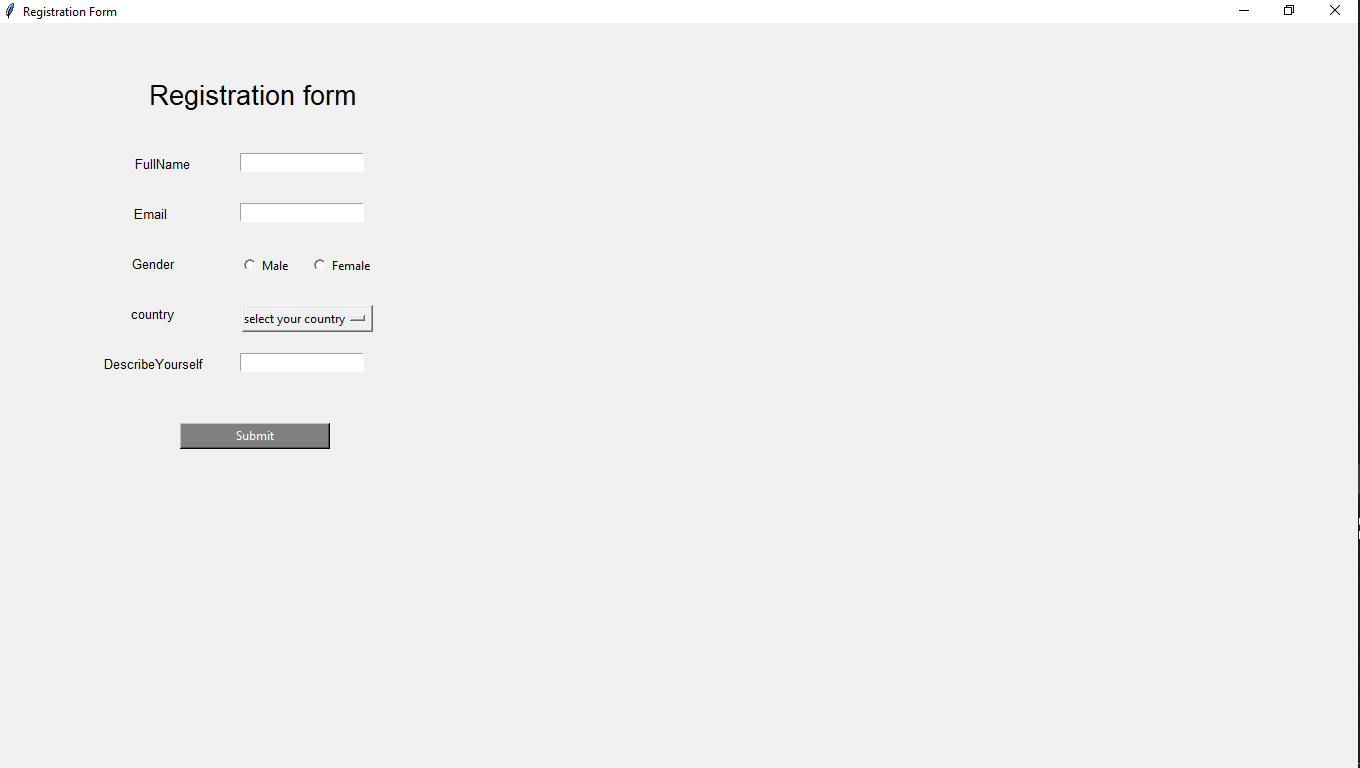
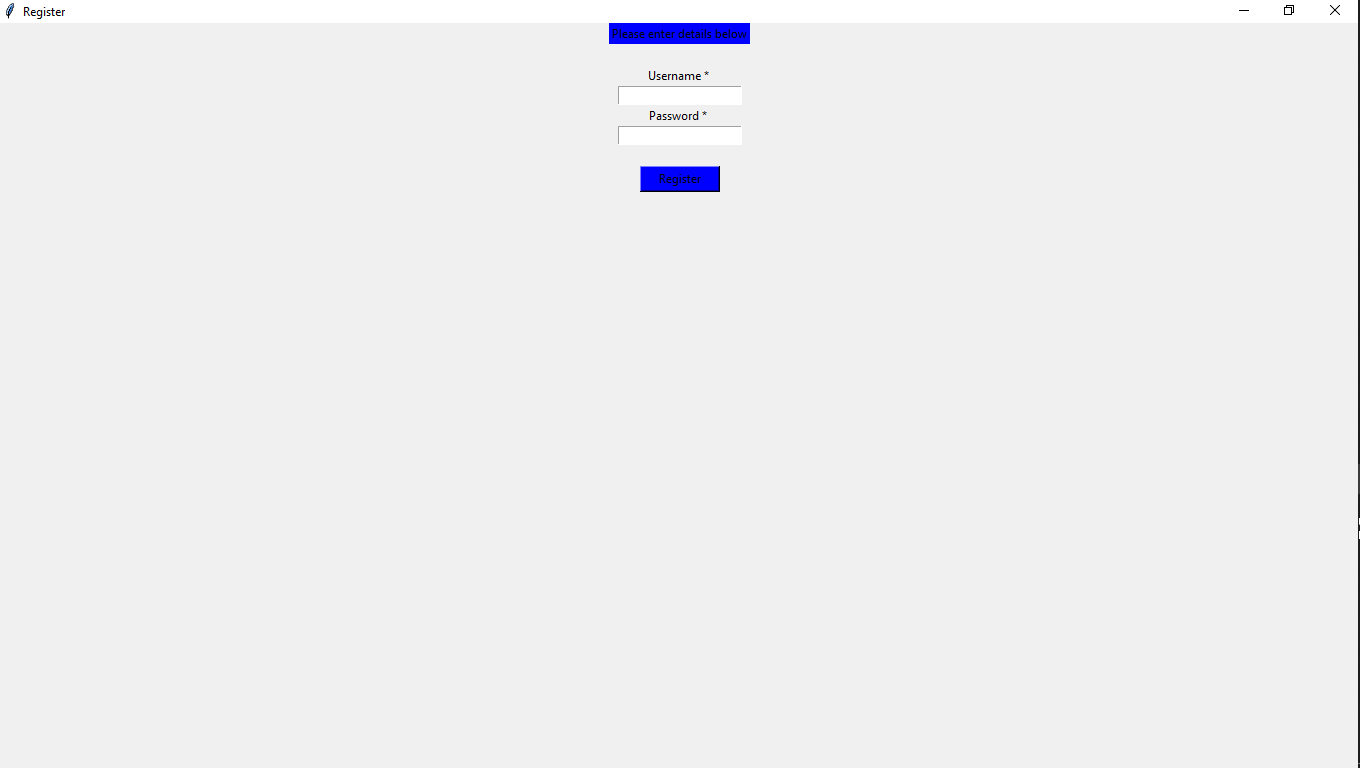
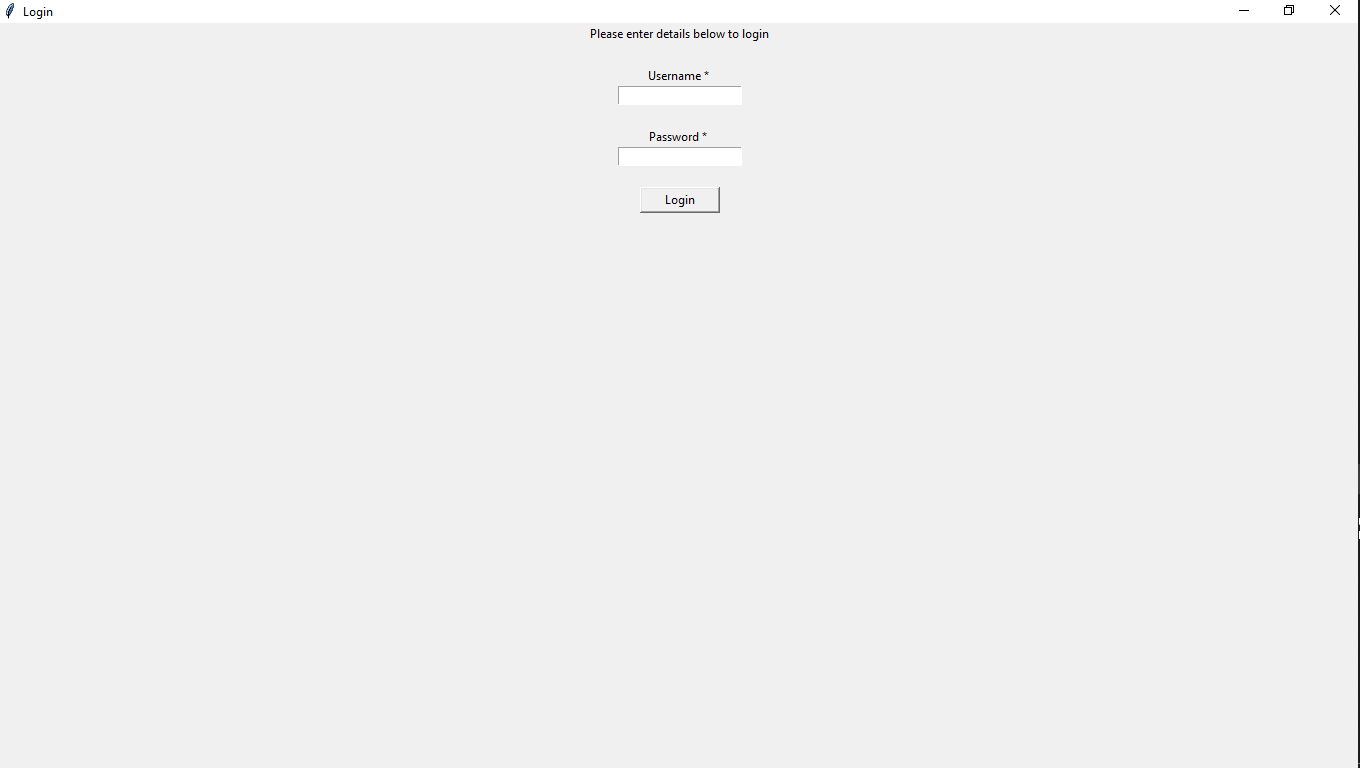
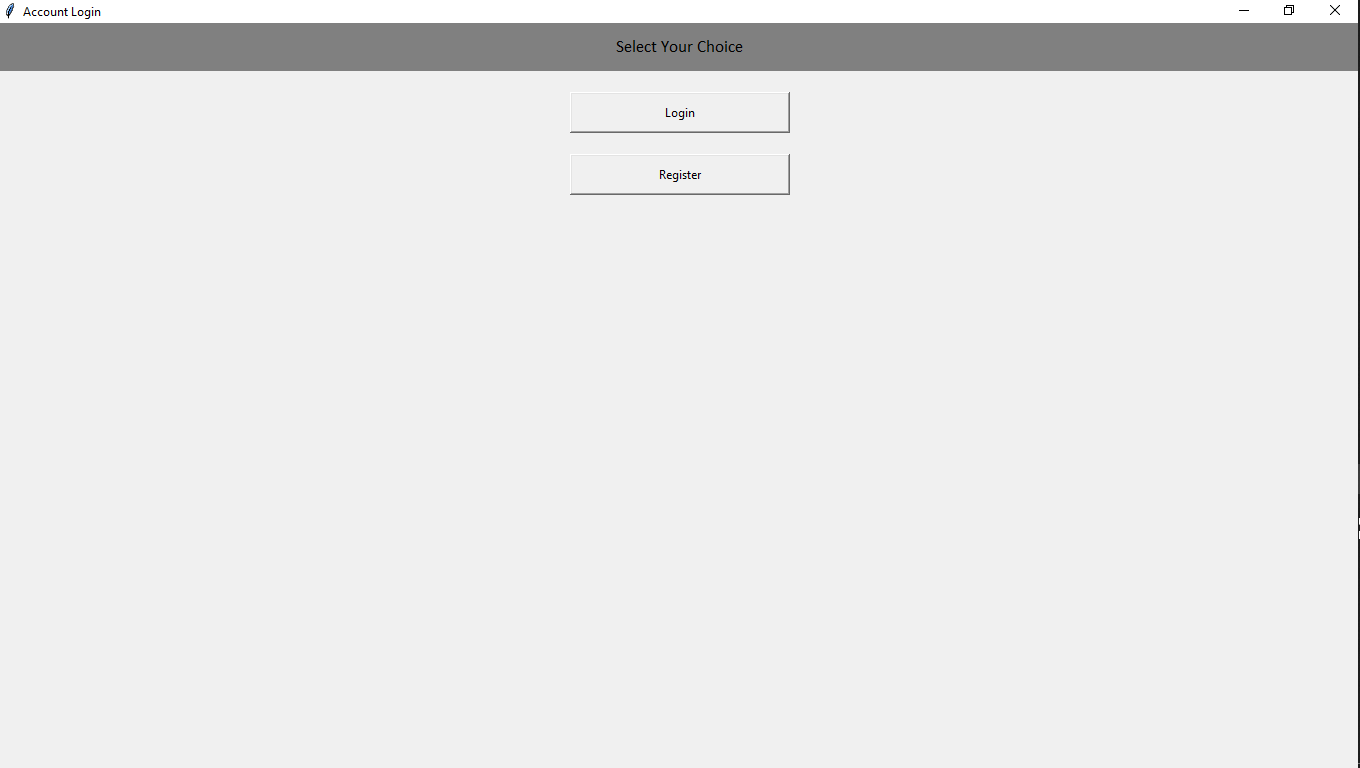
**b2.grid(row=7,column=4,padx=20,pady=30)**

**label=Label(a)**

**label.grid(row=7,column=4)**

**a.mainloop()**

**Screenshots**

****