TASK ASSIGNMENT -1

TASK:

Simple Registration form using Tkinter in Python:
You will be responsible for creating a basic registration form using Tkinter library in python. The form should include fields such as name, email, age, etc.

SIMPLE REGISTRATION FORM USING TKINTER IN PYTHON

This tutorial will show you how to use Python's tkinter to make a simple registration form.
We need to know a little bit about tkinter before we start. Therefore, we will first examine this tkinter module before making a basic registration form.

INTRODUCING TKINTER :

Although there are other GUI development alternatives in Python, Tkinter is the most popular. An ordinary Python library is Tkinter. The quickest and simplest approach to create an object-oriented GUI application is with Python and the tkinter package.
It offers us a number of standard GUI (graphical user interface) building blocks, such as buttons, menus, and other types of entry fields and display regions, that we may utilise to create our user interfaces.
Some of the common widgets used in Tkinter are :
Frame : serves as a holding area for other widgets and serves as a container.
Text : It enables us to display and alter text in a variety of styles and offers a prepared text display.
Label: Used to display text and images, but we are unable to interact with it.
Button: Often used add buttons and we may add functions and methods to it.
Entry: One-line string text can be entered into this widget.
Labelframe: For intricate window layouts, this widget serves as a separator or container.
Listbox : It just has text elements, all of which are the same colour and font.
Scrollbar: This gives a sliding controller.
Canvas : Custom widgets can be implemented using the canvas widget.
Scale : This widget offers graphical slider items that let us choose different scale values.
Radiobutton : Use a radio button to carry out one of several choices.
Checkbox : Use a checkbox to implement on-off choices.

Listbox : It just has text elements, all of which are the same colour and font.

STEPS FOR PREPARING THE REGISTRATION FORM:

Step 1: The first step is to import the tkinter module (using either tkinter import * or just import tkinter).

Step 2: The primary window of the GUI programme was created.

Step 3: Include one or more widgets in the GUI programme (controls such as buttons, labels, and text boxes, etc.).

Step 4: Enter the primary events to react to each event that the user has triggered.

MAKING A REGISTRATION FORM USING THE TKINTER IN PYTHON:

This section will discuss the tkinter-based registration form written in Python. At that time, we will talk about widgets and how to use them. The registration page's interface is the only part of the code that is published here. The entire application code is published at the bottom.

So, let's begin by building a simple registration form now, one that requires no backend effort. It can be utilized for database connectivity in the future.

Firstly, we will import the Tkinter module into the Python program.

Now, we will draw a window using the geometry method to specify the dimensions of the window and the title method to give title to the window drawn

Next, we will use various methods to create the variable entry boxes. First, we will create ‘starting’ label for registration form and using the place method it will be placed at certain a dimension. Next, we will create ‘name’ label for Full name and using the place method it will be placed at certain a dimension along with an entry box using Entry method with another place method. Next, we will create ‘age’ label for age and using the place method it will be placed at certain a dimension along with an entry box using Entry method with another place method. Next, we will create ‘dob’ label for date of birth and using the place method it will be placed at certain a dimension along with an entry box using Entry method with another place method. And later, we will create ‘place’ label for place and using the place method it will be placed at certain a dimension along with an entry box using Entry method with another place method. We will create ‘email’ label for mail details and using the place method it will be placed at certain a dimension along with an entry box using Entry method with another place method. We will create ‘password’ label for password and using the place method it will be placed at certain a dimension along with an entry box

using Entry method with another place method. We will create ‘confirm_password’ label for again confirming the password and using the place method it will be placed at certain a dimension along with an entry box using Entry method with another place method. We will also use a button method to add a register button and using the place method it will be placed at certain a dimension along with an entry box using Entry method with another place method.

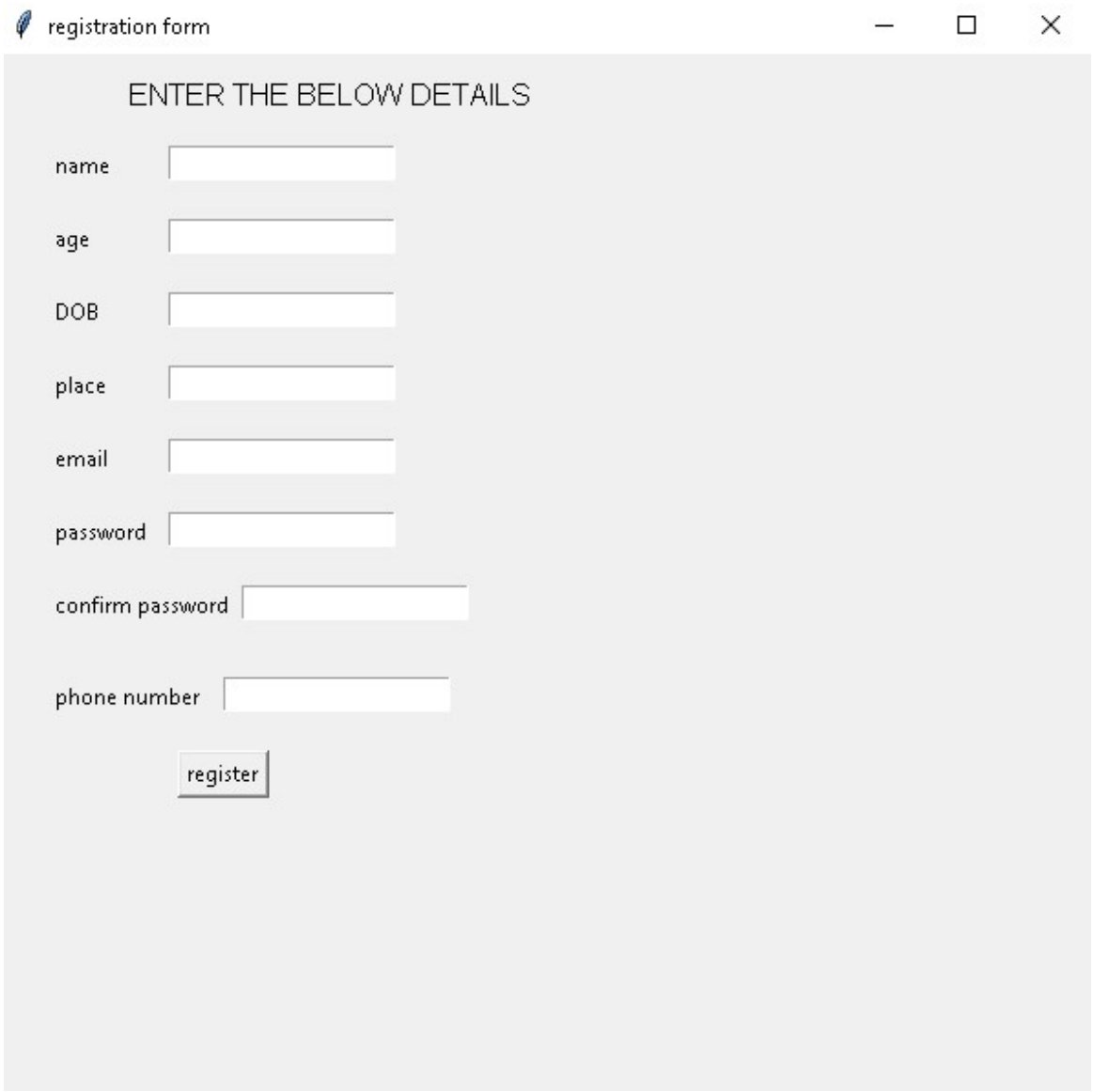
COMPLETE CODE:

```
from tkinter import*
from tkinter import messagebox
win = Tk()
win.title("registration form")
win.geometry('600x600')
starting = Label(win,text='ENTER THE BELOW
DETAILS',font=25,foreground='black').place(x=70,y=10)
def register_button():
    name=name_info.get()
    age=age_info.get()
    dob=dob_info.get()
    place=place_info.get()
    email=email_info.get()
    password=password_info.get()
    confirm_password=confirm_password_info.get()
    phno=phno_info.get()

    if name=="":
        messagebox.showwarning("Warning","please enter
your name")
    elif age=="":
        messagebox.showwarning("Warning","please enter
ur age")
    elif dob=="":
```

```
        messagebox.showwarning("Warning","please enter
your dob")
    elif place=="":
        messagebox.showwarning("warning","please enter
ur place")
    elif email=="":
        messagebox.showwarning("warning","please enter
ur email")
    elif password=="":
        messagebox.showwarning("warning","please enter
ur password")
    elif confirm_password=="":
        messagebox.showerror("Error","this is not
matching the password")
    elif phno=="":
        messagebox.showwarning("warning","please enter
ur phno")
    else:
        Label(win,text="you have registered
successfully",font='30',fg="green").place(x=145,y=500)

name = Label(win, text = 'name').place(x=30,y=50)
name_info=StringVar()
entry1 =
Entry(win,textvariable=name_info).place(x=95,y=50)
age = Label(win, text = 'age').place(x=30,y=90)
age_info=StringVar()
entry2 =
Entry(win,textvariable=age_info).place(x=95,y=90)
dob = Label(win, text = 'DOB').place(x=30,y=130)
dob_info=StringVar()
```

```
entry3 =
Entry(win,textvariable=dob_info).place(x=95,y=130)
place = Label(win, text = 'place').place(x=30,y=170)
place_info=StringVar()
entry4 =
Entry(win,textvariable=place_info).place(x=95,y=170)
email = Label(win, text = 'email').place(x=30,y=210)
email_info=StringVar()
entry5 =
Entry(win,textvariable=email_info).place(x=95,y=210)
password = Label(win, text =
'password').place(x=30,y=250)
password_info=StringVar()
entry6 =
Entry(win,textvariable=password_info).place(x=95,y=250)
confirm_password = Label(win, text = 'confirm
password').place(x=30,y=290)
confirm_password_info=StringVar()
entry7 =
Entry(win,textvariable=confirm_password_info).place(x=13
5,y=290)
phno = Label(win, text = 'phone
number').place(x=30,y=340)
phno_info=StringVar()
entry8 =
Entry(win,textvariable=phno_info).place(x=125,y=340)
register_button = Button(win,text = 'register',
activebackground='black',
activeforeground='blue',command=register_button).place(x
=100,y=380)
win.mainloop()
```

OUTPUT:



CONCLUSION:

The above window is opened when the code is said to be run. As we have already written in code that the warning has to be appeared whenever any entry is not filled. If the register button is

pressed then it will automatically shows the warning box that to fill the left over boxes. After the registration it shows a successful registration.