Python Assignment-8

Problem Statement: GUI using Tkinter

Theory:

- Tkinter is a standard GUI (Graphical User Interface) library for Python. It is a built-in module in Python, so you don't need to install any additional software to use it.

To create a basic Tkinter app in Python:

1. Import the Tkinter module: First, you'll need to import the Tkinter module by including the following line at the beginning of your script:

```
import tkinter as tk
```

2. Create the main window: Next, create the main window for your application by calling the Tk() function, like this:

```
root = tk.Tk()
```

3. Add widgets: Now, you can add widgets to your main window using various widget classes provided by Tkinter. For example, to add a label to your window, you can use the Label class like this:

```
my_label = tk.Label(root, text="Hello, Tkinter!")
my_label.pack()
```

   Label is the widget class, root is the parent window for the label, and text is the text to display on the label.

4. Pack the widgets: Once you've added your widgets to the main window, you need to pack them using the pack() method. This method tells Tkinter to arrange the widgets in the window.

```
my_label.pack()
```

Geometry Managers in Tkinter:

In Tkinter, a geometry manager is a layout manager that arranges the widgets in the main window. Tkinter provides three geometry managers: pack(), grid(), and place(), each with its own set of options.

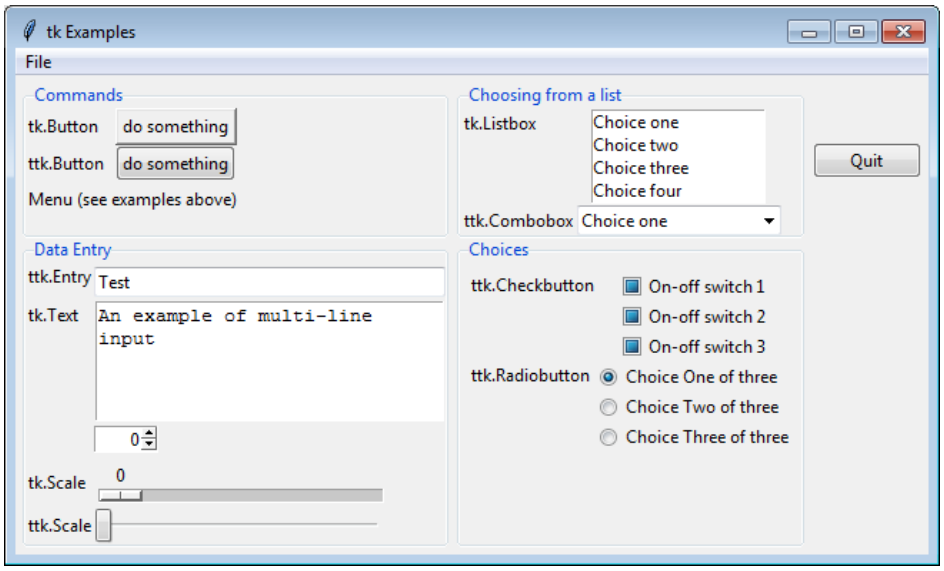
1. pack(): The pack() method arranges the widgets vertically or horizontally in the parent window. By default, widgets are added to the top of the parent window and arranged vertically. You can change this behavior by specifying the side option.

2. grid(): The grid() method arranges widgets in a grid pattern, where each cell can contain one widget. You can specify the number of rows and columns in the grid, as well as the row and column where each widget should be placed. You can also specify the sticky option to make the widget stick to the sides of the cell.

3. place(): The place() method allows you to specify the exact position of a widget using the x and y coordinates. You can also specify the anchor option to align the widget to a particular position, such as top, bottom, left, or right.

When using any of the geometry managers, you can also specify the options for each widget. Some common options include width, height, padx, and pady. These options allow you to set the size and padding of a widget.

Widgets in Tkinter:



In Tkinter, a widget is an object that represents a GUI component such as a button, label, entry, text box, or canvas. Tkinter provides a wide range of widgets that can be used to create a variety of GUI applications. Here are some commonly used widgets in Tkinter:

1. Button: The Button widget is used to create a clickable button that performs an action when clicked.
2. Label: The Label widget is used to display text or an image on the screen.
3. Entry: The Entry widget is used to accept input from the user, such as a name or email address.
4. Text: The Text widget is used to display and edit multiline text, such as a document or email.
5. Canvas: The Canvas widget is used to draw graphics, such as shapes or images.
6. Checkbutton: The Checkbutton widget is used to create a checkbox that can be checked or unchecked by the user.
7. Radiobutton: The Radiobutton widget is used to create a set of radio buttons that allow the user to choose only one option from a group.
8. Menu: The Menu widget is used to create a dropdown menu that allows the user to select an option from a list.
9. Scrollbar: The Scrollbar widget is used to add scrolling functionality to a widget, such as a Text or Listbox widget.
10. Listbox: The Listbox widget is used to display a list of items that the user can select from.
11. Frame: The Frame widget is used to group other widgets together and apply a common layout to them.
12. Toplevel: The Toplevel widget is used to create a separate window that can be used for dialogs, pop-ups, or multi-window applications.

In conclusion, Tkinter is a popular GUI framework for Python that provides a wide range of tools and widgets for creating interactive applications. Tkinter is easy to learn and use, making it an excellent choice for beginners who want to develop simple GUI applications quickly.

Code:

```
from tkinter import *
from tkinter import ttk

# Initialising the window
form_window = Tk()
form_window.geometry("600x900")
form_window.title("Industrial Visit Form")
form_window.configure(bg="light blue")

heading = Label(text = "Python Form", bg = "yellow")
heading.pack()

# Saving details in files
def register():
    fname_info = fname.get()
    lname_info = lname.get()
    age_info = age.get()
    gender_info = gender.get()
    depart_info = department.get()
    location_info = location.get()
    with open("TripDetails.txt","a") as f:
        f.write(f"{fname_info}\t {lname_info}\t {age_info}\t {gender_info}\t
{depart_info}\t  {location_info}\n ")
        print(f"User {fname_info} has been registered successfully")
       (f"{fname_info}\t {lname_info}\t {age_info}\t {gender_info}\t
{depart_info}\t  {location_info}\n")
# Defining Labels
fname_text = Label(text = "First Name: ",)
```

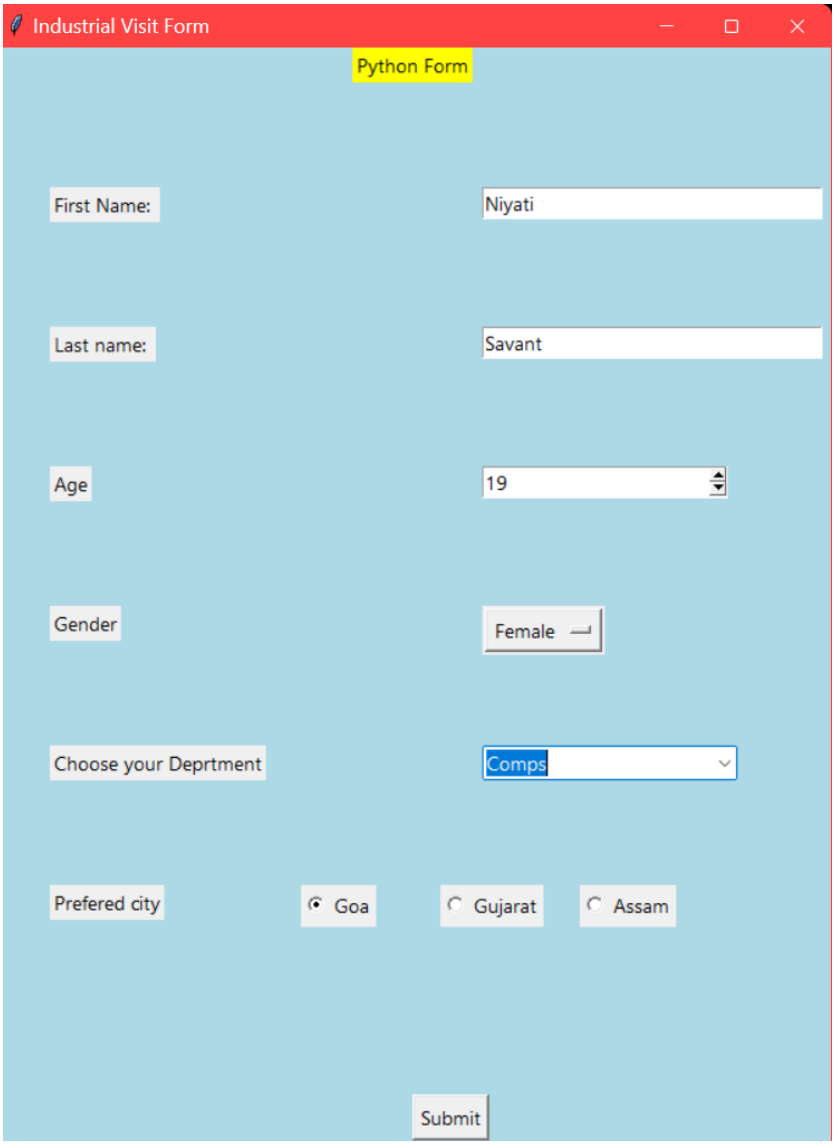
```
lname_text = Label(text = "Last name: ",)
age_text = Label(text = "Age",)
gender_text = Label(text = "Gender",)
department_text = Label(text = "Choose your Deprtment",)
location_text = Label(text = "Prefered city",)

# Defining Variables
fname = StringVar()
lname = StringVar()
age = StringVar()
gender = StringVar()
options = ["Male","Female","Other"]
department = StringVar()
location=StringVar()
places=("Goa","Gujarat","Assam")
#Text Areas
fname_entry = Entry(textvariable = fname, width="30")
lname_entry = Entry(textvariable = lname, width ="30")
#Spinbox
age_entry =Spinbox(form_window, from_=0, to = 30)
# Optionmenu
gender_entry = OptionMenu(form_window, gender, *options)
#Combobox
department_entry = ttk.Combobox(form_window, width = 20)
department_entry['values'] = ('Comps', 'AIDS','Chemical','IT')
#Radiobutton
i=0
for place in places:
    location_entry = Radiobutton(form_window, text =place,value=place,
variable=location)
```

```
    location_entry.place(x = 220 +i , y = 600)
    i =i+100
#Submit Button
submit = Button(form_window, text = "Submit" , command = register)
#Placing the Components
fname_text.place(x = 40, y = 100)
lname_text.place(x = 40, y = 200)
age_text.place(x = 40, y = 300)
gender_text.place(x = 40, y = 400)
department_text.place(x = 40, y = 500)
location_text.place(x = 40, y = 600)
fname_entry.place(x = 350, y = 100)
lname_entry.place(x = 350, y = 200)
age_entry.place(x = 350, y = 300)
gender_entry.place(x = 350, y = 400)
department_entry.place(x = 350, y = 500)
submit.place(x = 300, y = 750)
form_window.mainloop()
```

Output:

```
User Niyati has been registered successfully
Niyati Savant      19    Female             Comps      Goa
User Tanishqa has been registered successfully
Tanishqa Sawant 18       Female        AIDS     Gujarat
User Sai has been registered successfully
Sai    Sadu  17    Male         Chemical      Assam
User Jagjeet has been registered successfully
Jagjeet      Sappal      20    Male         IT         Gujarat
User Kaveri has been registered successfully
Kaveri       Solat       21    Female            Comps      Goa
```


Industrial Visit Form
Python Form
First Name:
Niyati
Last name:
Savant
Age
19
Gender
Female
Choose your Deprtment
Comps
Prefered city
Goa
Gujarat
Assam
Submit