

Experiment 4

Aim: To create GUI with python containing widgets such as labels, textbox, radio, checkboxes and custom dialog boxes.

THEORY:

- List the importance of GUI

Ans: The importance of GUI is -

1. GUIs enable interaction through clarity and control
2. Effective GUIs facilitate a seamless user journey
3. Good GUI design can be shaped to anticipate audience needs
4. It captures the user's attention, and keep it
5. It delivers consistency in the program.
6. GUI systems have made computers far more user-friendly than CLI systems.

- Write all functions to create various GUI widgets

Ans:

The different types of GUI widgets are -

1. Tkinter Label Widget -

For the label widget here, we will define it using the Label constructor itself. The label is going to go in the root main window and the text will say "Hey, welcome to this my GUI". Then we pack the label inside the window and we have provided an argument with pady to give us a little bit more space on the y-axis.

Example:

```
label=Label(root,text="Hey, welcome to this my GUI")  
label.pack(pady=10)
```

2. Tkinter Button Widget -

The button will be placed on the same main window and is created using the `Button()` constructor. The text for the button will say “press button”. Notice that the text color is green. For that, we have assigned green to the foreground. When the button is pressed we want to activate a function and we’re going to assign that function to the command argument. The name of the function is `button_trigger()`. When the button is pressed, it’s going to activate this function and print a message that says “button pressed”. We have packed the button into the main root window. So when we press this button it’s going to activate this function and it’s going to print the message in the console.

Example -

```
def button_trigger():  
    print("Button Pressed")  
  
button = Button(root,text="press button", foreground="green",  
command=button_trigger)  
button.pack(pady=10)
```

3. Tkinter Check Button Widget -

When we check this box or button it’s going to turn the background white, like turning on a light. Then if we uncheck it, it’s going to turn the background black like turning a light off. So first, create the check button using `Checkbutton()`. It’s going on the root main window. The text is “on/off”. We have associated a variable with this check button and it is an Integer. The function that will be activated by the check button and is named `button_action`. The check button has two default states which are 0 and 1 and those default states will be assigned to this variable here. This variable will keep track

of the state of the check button and to get the state of the check button. We just go ahead and reference the variable.get(). If the state of the check button is 1, that is equivalent to the box being checked and we're going to make the background of the window white. If it is 0, we're going to make the background of the root window black which gives us the effect of turning the light on or off. We have then packed this into the "main" frame with a pady of 10.

```
def check_button_action():  
    print(check_button_var.get())  
  
    if check_button_var.get() == 1:  
        root.configure(background='white')  
    else:  
        root.configure(background='black')  
  
check_button_var = IntVar()  
check_button = tk.Checkbutton(root, text="on/off", variable=check_button_var,  
command= button_action)  
check_button.pack(pady=10)
```

4. Tkinter Entry Widget -

The entry widget allows us to type in text and transfers the text from the text box or entry to the console and displays the message on the console. To create the entry widget, we've gone ahead and created a frame. To create the frame, we use Frame(). The frame is going to go in the main root window with a border width of 5 with a sunken effect. We reference the framed pack and that will pack the frame into the main window. We have then created our entry text box and the entry is going to go into the frame. We packed the entry into the frame. So the frame will go into the main window and the entry will go into the frame. Then we go ahead and create a button

that will transfer the text from the entry to the console. Now notice that our entry message is printed to the console, and also updated our label on the mainframe. To get the text we just use the `get()` method.

Example-

```
entry_frame = Frame(root, borderwidth=5, relief = SUNKEN)
entry_frame.pack()
```

```
text_box=Entry(entry_frame)
text_box.pack()
```

```
def get_entry_text():
    print(text_box.get())
    label.configure(text=text_box.get())
```

```
button = Button(entry_frame, text='get entry', command=get_entry_text)
button.pack(pady=10)
```

5. Tkinter ListBox Widget -

Next let's go over the list box widget. So here we have our list box with five items. In this example, we're just going to choose one of the items. Then we're going to press the button and we want to transfer the text from the item to a label. To create the list box we use `Listbox()`. We're going to put the list box inside of a frame. After we have created our list box, we can go ahead and insert items into the list box. If you'd like to insert several items, you can do it with a for Loop. Here we have created a button, when we press the button. It's going to activate the `list_item_selected()` created. To access the selection, we reference the `listbox.curselection()`. To make sure that we actually have something selected, we use `if selection:` and if we have an item selected, we reference the listbox item and then we get the actual item. The reason that we have

used the square brackets with the zero is that the item is typically a single digit Tuple and this will give us just the digit. Then we want to go ahead and update our label with the item that we selected.

Example-

```
listbox_frame=Frame(root,borderwidth=5, relief=SUNKEN)
listbox_frame.pack(pady=10)
```

```
listbox=Listbox(listbox_frame)
listbox.pack()
```

```
listbox.insert(END,"one")
```

```
for item in ["two","three","four","five"]:
    listbox.insert(END, item)
```

```
def list_item_selected():
    selection=listbox.curselection()
    if selection:
        print(listbox.get(selection[0]))
        list_box_label.configure(text=listbox.get(selection[0]))
```

```
list_box_button = Button(listbox_frame,text='list item button',
command=list_item_selected)
list_box_button.pack()
```

6. Tkinter RadioButton Widget -

So for the last example, let's go over radio buttons. Now depending on the radio button selected, we're going to get an image displayed here. So here we have mountains, boating and camping. All of the radio buttons will be placed inside the main root window. For the text, we have gone ahead and assigned "mountains, boating and camping". All of the radio buttons are going to have a value associated with one variable and we have created that variable. In this case, since you can only

click one radio button at a time, the value associated with either of the radio buttons, which we have assigned here 1, 2, or 3, will be assigned to the variable. When a radio button is clicked, it is going to activate or call the “radio_button_func()”. So if this first radio button is clicked for mountains, the value of 1 will be assigned to this variable and then we will get that value and test if it's equal to 1. And if it is equal to 1, we are going to use the Unicode text representation for mountains.

```
Label(root, text="choose icon")
```

```
def radio_button_func():  
    print(rb_icon_var.get())  
    if(rb_icon_var.get())==1:  
        radio_button_icon.configure(text='\u26F0')  
    elif rb_icon_var.get()==2:  
        radio_button_icon.configure(text='\u26F5')  
    elif rb_icon_var.get()==3:  
        radio_button_icon.configure(text='\u26FA')
```

```
rb_icon_var=IntVar()
```

```
Radiobutton(root,text="mountains",variable=rb_icon_var, value=1,  
command=radio_button_func).pack()  
Radiobutton(root,text="boating",variable=rb_icon_var, value=2,  
command=radio_button_func).pack()  
Radiobutton(root,text="camping",variable=rb_icon_var, value=3,  
command=radio_button_func).pack()
```

```
radio_button_icon = Label(root, text=' ', font=("Helvetica",150))  
radio_button_icon.pack()
```

IMPLEMENTATION:

1. Implement a registration form having widgets for

- **UserName (entry box)**
- **Password**
- **Contact No (entry box)**
- **Address (Text)**
- **Country (Combo box)**
- **Email (entry box)**
- **Gender(radiobutton)**
- **Languages(checkbutton)**
- **Course(List box, single selection)**
- **Button(Submit, Reset)**

2. Make use of Labels ,titles and frames in the above code

Code:

```
from tkinter import *
import tkinter as tk
from tkinter import ttk

def reset_values():
    entry_1.delete(0,'end')
    entry_2.delete(0,'end')
    entry_3.delete(0,'end')
    entry_4.delete(0,'end')
    number=""
    entry_6.delete(0,'end')
    var=""
    c.set('Interested Course')
```

```
var1=""
```

```
var2=""
```

```
var3=""
```

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

```
root.geometry('500x700')
```

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

```
label_0 = Label(root, text="Registration form",width=20,font=("bold", 20))
```

```
label_0.place(x=90,y=53)
```

```
label_1 = Label(root, text="Username",width=20,font=("bold", 10))
```

```
label_1.place(x=80,y=130)
```

```
entry_1 = Entry(root)
```

```
entry_1.place(x=240,y=130)
```

```
label_2 = Label(root, text="Password",width=20,font=("bold", 10))
```

```
label_2.place(x=80,y=180)
```

```
entry_2 = Entry(root)
```

```
entry_2.place(x=240,y=180)
```

```
label_3 = Label(root, text="Contact No",width=20,font=("bold", 10))
```



```
label_3.place(x=73,y=230)
```

```
entry_3 = Entry(root)
```

```
entry_3.place(x=240,y=230)
```

```
label_4 = Label(root, text="Address",width=20,font=("bold", 10))
```

```
label_4.place(x=73,y=280)
```

```
entry_4 = Entry(root)
```

```
entry_4.place(x=240,y=280)
```

```
label_5 = Label(root, text="Country",width=20,font=("bold", 10))
```

```
label_5.place(x=73,y=330)
```

```
number = tk.StringVar()
```

```
numberChosen = ttk.Combobox(root, width = 17)# Adding Values
```

```
numberChosen['values'] = ("India", "USA", "England")
```

```
numberChosen.grid(column = 0, row = 1)
```

```
numberChosen.current()# Calling Main()
```

```
numberChosen.place(x=240,y=330)
```

```
label_6 = Label(root, text="Email",width=20,font=("bold", 10))
```

```
label_6.place(x=73,y=380)
```

```
entry_6 = Entry(root)
```

```
entry_6.place(x=240,y=380)
```

```
label_7 = Label(root, text="Gender",width=20,font=("bold", 10))
label_7.place(x=73,y=430)
var = IntVar()
Radiobutton(root, text="Male",padx = 5, value=1).place(x=235,y=430)
Radiobutton(root, text="Female",padx = 20, value=2).place(x=290,y=430)
```

```
label_8 = Label(root, text="Languages",width=20,font=("bold", 10))
label_8.place(x=73,y=480)
var1 = IntVar()
Checkbutton(root, text="English").place(x=235,y=480)
var2 = IntVar()
Checkbutton(root, text="Hindi").place(x=290,y=480)
var3 = IntVar()
Checkbutton(root, text="German").place(x=345,y=480)
```


```
label_9 = Label(root, text="Courses",width=20,font=("bold", 10))
label_9.place(x=73,y=530)
```

```
listed = ['C','C++','Java','C#','Python','Ruby'];
c=StringVar()
droplist=OptionMenu(root,c, *listed)
droplist.config(width=15)
c.set('Interested Course')
droplist.place(x=240,y=530)
```

```
Button(root, text='Submit',width=20,bg='white',fg='black').place(x=100,y=580)
```

```
Button(root,command =reset_values,  
text='Reset',width=20,bg='white',fg='black').place(x=250,y=580)  
root.mainloop()
```

Output:

Registration Form

Registration form

Username

keeganvaz

Password

qwertyui

Contact No

1234567890

Address

abc, xyz 4000 069

Country

India

Email

abcxyz@gmail.com

Gender

☒ Male ☐ Female

Languages

☒ Englis ☒ Hindi ☐ German

Courses

Python

Submit

Reset

Academic Year: **2020-21**


Subject: **Skill Base Lab Course: Python Programming**

Class/ Branch/ Sem: **SE CMPN B**

Roll No. 42

PID: 192120

(After Reset)-

 Registration Form—□×

Registration form

Username

Password

Contact No

Address

Country

Email

Gender

☒ Male ☐ Female

Languages

☐ Englis ☐ Hindi ☐ German

Courses

Interested Course

Submit

Reset

ST. FRANCIS INSTITUTE OF TECHNOLOGY

MT. POINSUR, BORIVALI (W), MUMBAI

Academic Year: **2020-21**

Subject: **Skill Base Lab Course: Python Programming**

Class/ Branch/ Sem: **SE CMPN B**

Roll No. 42

PID: 192120

Conclusion:

Successfully learnt GUI widgets in Python and implementation of **Tkinter**.