| Experiment No. 8 |
| --- |
| Creating GUI with python containing widgets such as labels, textbox, radio, checkboxes and custom dialog boxes |
| Date of Performance: |
| Date of Submission: |

# Experiment No. 8

**Title:** Creating GUI with python containing widgets such as labels, textbox, radio, checkboxes and custom dialog boxes

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

**Objective:** To introduce GUI, TKinter in python

**Theory:**

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

To create a tkinter app:

Importing the module – tkinter

Create the main window (container)

Add any number of widgets to the main window

Apply the event Trigger on the widgets.

Importing tkinter is same as importing any other module in the Python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x it is 'tkinter'.

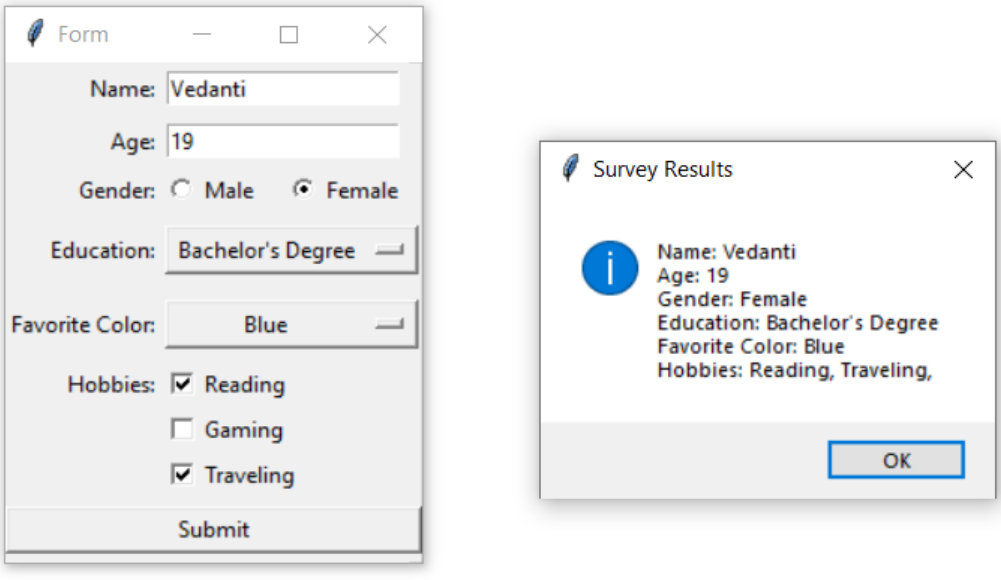
**CODE:**

```
import tkinter as tk
from tkinter import messagebox

def submit_survey():
    name = name_entry.get()
    age = age_entry.get()
    gender = gender_var.get()
    education = education_var.get()
    favorite_color = color_var.get()
    hobbies = ""
    for hobby, var in hobbies_vars.items():
        if var.get():
            hobbies += hobby + ", "

    messagebox.showinfo("Survey Results",
                        f"Name: {name}\n"
                        f"Age: {age}\n"
                        f"Gender: {gender}\n"
                        f"Education: {education}\n"
                        f"Favorite Color: {favorite_color}\n"
                        f"Hobbies: {hobbies}")

root = tk.Tk()
root.title("Form")

name_label = tk.Label(root, text="Name:")
name_label.grid(row=0, column=0, sticky="E")
name_entry = tk.Entry(root)
name_entry.grid(row=0, column=1, pady=5)

age_label = tk.Label(root, text="Age:")
age_label.grid(row=1, column=0, sticky="E")
age_entry = tk.Entry(root)
age_entry.grid(row=1, column=1, pady=5)

gender_label = tk.Label(root, text="Gender:")
gender_label.grid(row=2, column=0, sticky="E")
gender_var = tk.StringVar()
male_radio = tk.Radiobutton(root, text="Male", variable=gender_var, value="Male")
male_radio.grid(row=2, column=1, sticky="W")
female_radio = tk.Radiobutton(root, text="Female", variable=gender_var, value="Female")
female_radio.grid(row=2, column=1, sticky="E")

education_label = tk.Label(root, text="Education:")
education_label.grid(row=3, column=0, sticky="E")
```

```
education_var = tk.StringVar()
education_options = ["High School", "Bachelor's Degree", "Master's Degree", "PhD"]
education_dropdown = tk.OptionMenu(root, education_var, *education_options)
education_dropdown.grid(row=3, column=1, columnspan=2, pady=5, sticky="WE")

color_label = tk.Label(root, text="Favorite Color:")
color_label.grid(row=4, column=0, sticky="E")
color_var = tk.StringVar()
color_options = ["Red", "Green", "Blue", "Yellow", "Other"]
color_dropdown = tk.OptionMenu(root, color_var, *color_options)
color_dropdown.grid(row=4, column=1, columnspan=2, pady=5, sticky="WE")

hobbies_label = tk.Label(root, text="Hobbies:")
hobbies_label.grid(row=5, column=0, sticky="E")
hobbies_vars = {}
hobbies_options = ["Reading", "Gaming", "Traveling", "Cooking"]
for i, hobby in enumerate(hobbies_options):
    hobbies_vars[hobby] = tk.IntVar()
    hobby_check = tk.Checkbutton(root, text=hobby, variable=hobbies_vars[hobby])
    hobby_check.grid(row=5+i, column=1, sticky="W")

submit_button = tk.Button(root, text="Submit", command=submit_survey)
submit_button.grid(row=6+len(hobbies_options)//2, column=0, columnspan=3, pady=5,
sticky="WE")

root.mainloop()
```

**OUTPUT:**

```
============== RESTART: C:\Vedanti_Degree\SEM_4\SBL_PYTHON\pr8.py ==============
```



**CONCLUSION:**

In conclusion, creating graphical user interfaces (GUIs) with Python using libraries such as Tkinter provides a powerful and flexible way to develop interactive applications. By leveraging various widgets such as labels, textboxes, radio buttons, checkboxes, and custom dialog boxes, we can design intuitive and user-friendly interfaces for our applications. By combining these widgets and arranging them using layout managers such as grid or pack, we can create visually appealing and functional GUIs tailored to our application's requirements.