|  |
| --- |
| 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: |

Top of Form

NAME :SUPRIYA SUBHASH VIRKAR DIV :SE-3 ROLL NO:58

**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 ttk

# Create main window

window = tk.Tk()

window.title("Simple Tkinter Form")

# Label widget

label = ttk.Label(window, text="Enter your information:")

label.pack()

# Entry widget (text field)

name\_label = ttk.Label(window, text="Name:")

name\_label.pack()

name\_entry = ttk.Entry(window)

name\_entry.pack()

# Radio button widget

gender\_label = ttk.Label(window, text="Gender:")

gender\_label.pack()

gender\_var = tk.StringVar()

male\_radio = ttk.Radiobutton(window, text="Male", variable=gender\_var, value="Male")

female\_radio = ttk.Radiobutton(window, text="Female", variable=gender\_var, value="Female")

male\_radio.pack()

female\_radio.pack()

# Checkbox widget

hobby\_label = ttk.Label(window, text="Hobbies:")

hobby\_label.pack()

hobby1\_var = tk.IntVar()

hobby2\_var = tk.IntVar()

hobby1\_check = ttk.Checkbutton(window, text="Reading", variable=hobby1\_var)

hobby2\_check = ttk.Checkbutton(window, text="Gaming", variable=hobby2\_var)

hobby1\_check.pack()

hobby2\_check.pack()

# Button widget

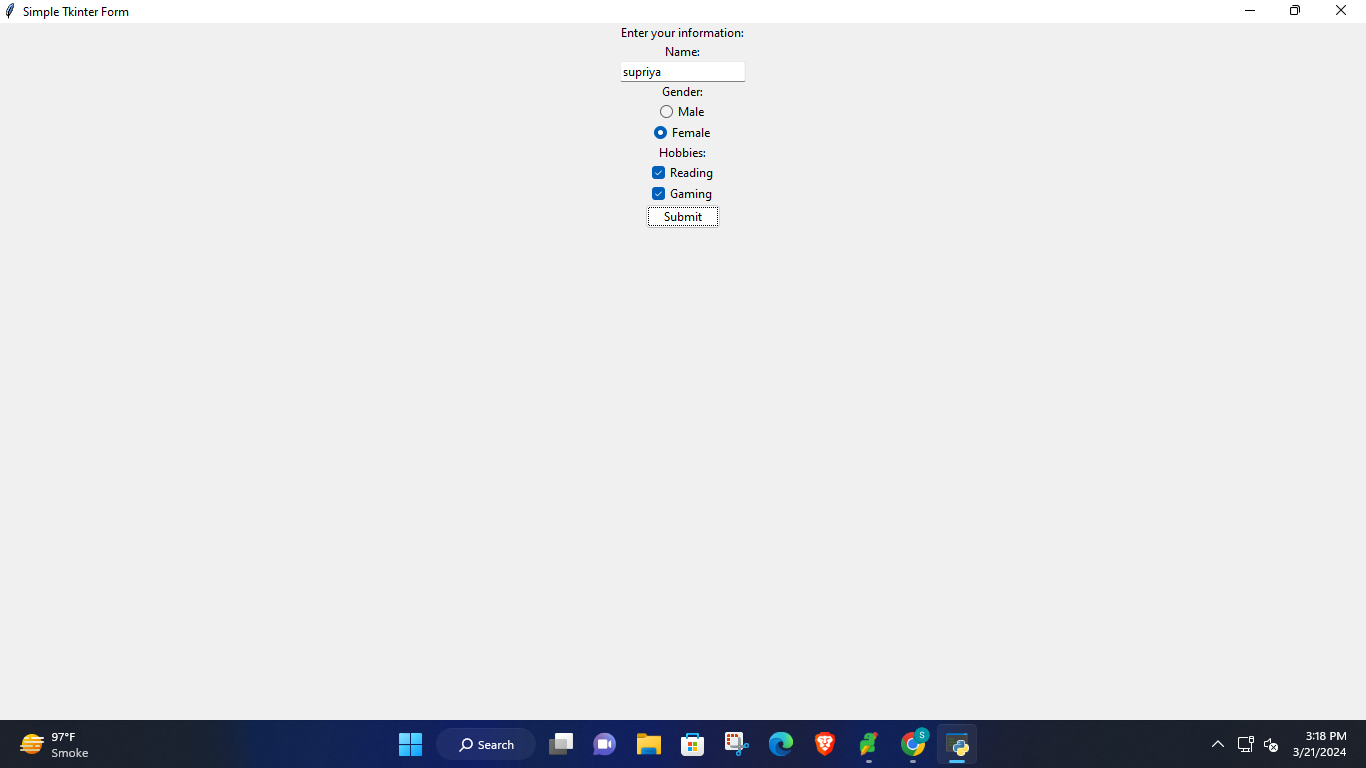
submit\_button = ttk.Button(window, text="Submit")

submit\_button.pack()

# Run the main event loop

window.mainloop()

Output:



**Conclusion:** In conclusion, the provided Python code demonstrates how to create a graphical user interface (GUI) using the Tkinter library. The GUI includes various widgets such as labels, text boxes, radio buttons, checkboxes, and buttons. Additionally, it showcases the integration of custom dialog boxes using Tkinter's **messagebox** module.

This example serves as a foundation for building more complex GUI applications in Python, allowing developers to create interactive interfaces for their programs with ease. Tkinter's simplicity and versatility make it a popular choice for GUI development in Python, suitable for both beginner and advanced users. By leveraging the features of Tkinter, developers can create intuitive and visually appealing applications tailored to their specific needs.

Top of Form