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:

```python
import tkinter as tk

from tkinter import messagebox

from tkinter import ttk


def show_message():

    messagebox.showinfo("Message", "Hello, you clicked the button!")


def submit():

    user_input = entry.get()

    label.config(text="You entered: " + user_input)


# Create the main window

root = tk.Tk()

root.title("Tkinter GUI")


# Create and add widgets

label = tk.Label(root, text="Welcome to Tkinter GUI", font=("Helvetica", 16))

label.pack(pady=10)


entry = tk.Entry(root, width=30, font=("Helvetica", 12))

entry.pack(pady=5)
```

```python
button = tk.Button(root, text="Click Me", command=show_message, font=("Helvetica", 12))

button.pack(pady=5)


submit_button = tk.Button(root, text="Submit", command=submit, font=("Helvetica", 12))

submit_button.pack(pady=5)


check_var = tk.BooleanVar()

check_var.set(True)

check_button = tk.Checkbutton(root, text="Check me", variable=check_var, font=("Helvetica", 12))

check_button.pack(pady=5)


radio_var = tk.StringVar()

radio_var.set("Option 1")

radio_button1 = tk.Radiobutton(root, text="Option 1", variable=radio_var, value="Option 1", font=("Helvetica", 12))

radio_button1.pack()

radio_button2 = tk.Radiobutton(root, text="Option 2", variable=radio_var, value="Option 2", font=("Helvetica", 12))

radio_button2.pack()


scale = tk.Scale(root, from_=0, to=100, orient=tk.HORIZONTAL, font=("Helvetica", 12))

scale.pack(pady=5)
```

```python
# Add Treeview widget to display Indian cities with states

tree = ttk.Treeview(root, columns=("City", "State"), selectmode="browse", height=5)

tree.heading("#0", text="City")

tree.heading("#1", text="State")

tree.pack(pady=5)


# Add Indian city names and states

cities = [

    ("Delhi", "Delhi"),

    ("Mumbai", "Maharashtra"),

    ("Bangalore", "Karnataka"),

    ("Kolkata", "West Bengal"),

    ("Chennai", "Tamil Nadu")

]

for city, state in cities:

    tree.insert("", tk.END, text=city, values=(city, state))


# Add Spinbox widget

spinbox = tk.Spinbox(root, from_=0, to=100, font=("Helvetica", 12))

spinbox.pack(pady=5)


# Add Text widget
```
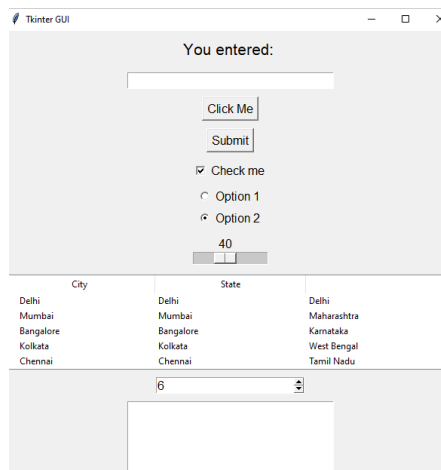
text_widget = tk.Text(root, height=5, width=30, font=("Helvetica", 12))

text_widget.pack(pady=5)

Run the application

root.mainloop()

Output:

Conclusion:

GUI package TKinter has been studied and implemented.