Experiment No. 8
Creating GUI with python containing widgets such as labels, textbox, radio, checkboxes and custom dialog boxes
Date of Performance:12-03-2024
Date of Submission:26-03-2024

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


def greet():
    name = entry.get()
```

```
    gender = gender_var.get()
    greeting_label.config(text=f"Hello, {name}! You are {gender}.")


# Create the main window
root = tk.Tk()
root.title("Simple GUI")


# Add a Label
label = tk.Label(root, text="Enter your name:")
label.pack()


# Add an Entry widget
entry = tk.Entry(root)
entry.pack()


# Add a Label for gender selection
gender_label = tk.Label(root, text="Select your gender:")
gender_label.pack()


# Variable to hold the selected gender
gender_var = tk.StringVar()


# Add Male and Female Radiobuttons
```

```
male_radio = tk.Radiobutton(root, text="Male", variable=gender_var, value="male")
male_radio.pack()

female_radio = tk.Radiobutton(root, text="Female", variable=gender_var, value="female")
female_radio.pack()

# Set default gender
gender_var.set("male")

# Add a Button
button = tk.Button(root, text="Greet", command=greet)
button.pack()

# Add a Label to display the greeting
greeting_label = tk.Label(root, text="")
greeting_label.pack()

# Start the GUI event loop
root.mainloop()
```

RESULT:

Conclusion: The study and implementation of the TKinter GUI package offer valuable insights into creating graphical user interfaces in Python. Through hands-on experience, developers gain proficiency in designing interactive applications with buttons, labels, and other widgets. Understanding TKinter enhances the ability to develop user-friendly software solutions tailored to diverse user needs. With its simplicity and versatility, TKinter empowers developers to build visually appealing and intuitive interfaces for their Python applications. The exploration of TKinter enriches programming skills, fostering creativity and innovation in GUI development. Through its extensive documentation and community support, TKinter serves as a valuable resource for developers seeking to create polished and functional graphical interfaces in Python.