|  |
| --- |
| Experiment No. 8 |
| Creating GUI with python containing widgets such as labels, textbox, radio, checkboxes and custom dialog boxes |
| Date of Performance: 18/03/2024 |
| Date of Submission: 25/03/2024 |

**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’.

**Program:**

import tkinter as tk

from tkinter import messagebox

def stop\_application():

root.destroy()

def submit\_form():

name = entry\_name.get()

email = entry\_email.get()

age = entry\_age.get()

# Check if any language is selected

if language\_var.get() == "":

messagebox.showerror("Error", "Please select your favorite programming language.")

return

# Get the selected language

language = language\_var.get()

# Get the selected gender

gender = gender\_var.get()

# Get the checked hobbies

hobbies = [hobby for hobby, var in hobbies\_vars.items() if var.get()]

# Display submitted information

message = f"Name: {name}\nEmail: {email}\nAge: {age}\nLanguage: {language}\nGender: {gender}\nHobbies: {', '.join(hobbies)}"

messagebox.showinfo("Form Submitted", message)

root = tk.Tk()

root.title("Comprehensive GUI Application")

# Styling

root.configure(bg="#f0f0f0")

# Labels

label\_name = tk.Label(root, text="Name:", bg="#f0f0f0")

label\_name.grid(row=0, column=0, padx=10, pady=5, sticky="e")

label\_email = tk.Label(root, text="Email:", bg="#f0f0f0")

label\_email.grid(row=1, column=0, padx=10, pady=5, sticky="e")

label\_age = tk.Label(root, text="Age:", bg="#f0f0f0")

label\_age.grid(row=2, column=0, padx=10, pady=5, sticky="e")

label\_language = tk.Label(root, text="Favorite Programming Language:", bg="#f0f0f0")

label\_language.grid(row=3, column=0, padx=10, pady=5, sticky="e")

label\_gender = tk.Label(root, text="Gender:", bg="#f0f0f0")

label\_gender.grid(row=4, column=0, padx=10, pady=5, sticky="e")

label\_hobbies = tk.Label(root, text="Hobbies:", bg="#f0f0f0")

label\_hobbies.grid(row=5, column=0, padx=10, pady=5, sticky="e")

# Entry Widgets

entry\_name = tk.Entry(root)

entry\_name.grid(row=0, column=1, padx=10, pady=5)

entry\_email = tk.Entry(root)

entry\_email.grid(row=1, column=1, padx=10, pady=5)

entry\_age = tk.Entry(root)

entry\_age.grid(row=2, column=1, padx=10, pady=5)

# Dropdown Menu

languages = ["Python", "Java", "C++", "JavaScript", "Ruby", "Other"]

language\_var = tk.StringVar()

language\_var.set("") # Default value

dropdown\_language = tk.OptionMenu(root, language\_var, \*languages)

dropdown\_language.grid(row=3, column=1, padx=10, pady=5, sticky="ew")

# Radio Buttons

gender\_var = tk.StringVar()

gender\_var.set("Male") # Default value

radio\_male = tk.Radiobutton(root, text="Male", variable=gender\_var, value="Male", bg="#f0f0f0")

radio\_male.grid(row=4, column=1, padx=10, pady=5, sticky="w")

radio\_female = tk.Radiobutton(root, text="Female", variable=gender\_var, value="Female", bg="#f0f0f0")

radio\_female.grid(row=4, column=1, padx=10, pady=5, sticky="e")

# Checkboxes

hobbies\_list = ["Reading", "Gaming", "Traveling", "Music", "Sports"]

hobbies\_vars = {}

for i, hobby in enumerate(hobbies\_list):

var = tk.BooleanVar()

checkbox = tk.Checkbutton(root, text=hobby, variable=var, bg="#f0f0f0")

checkbox.grid(row=6+i, column=1, padx=10, pady=2, sticky="w")

hobbies\_vars[hobby] = var

# Buttons

submit\_button = tk.Button(root, text="Submit", command=submit\_form, bg="#008000", fg="#ffffff")

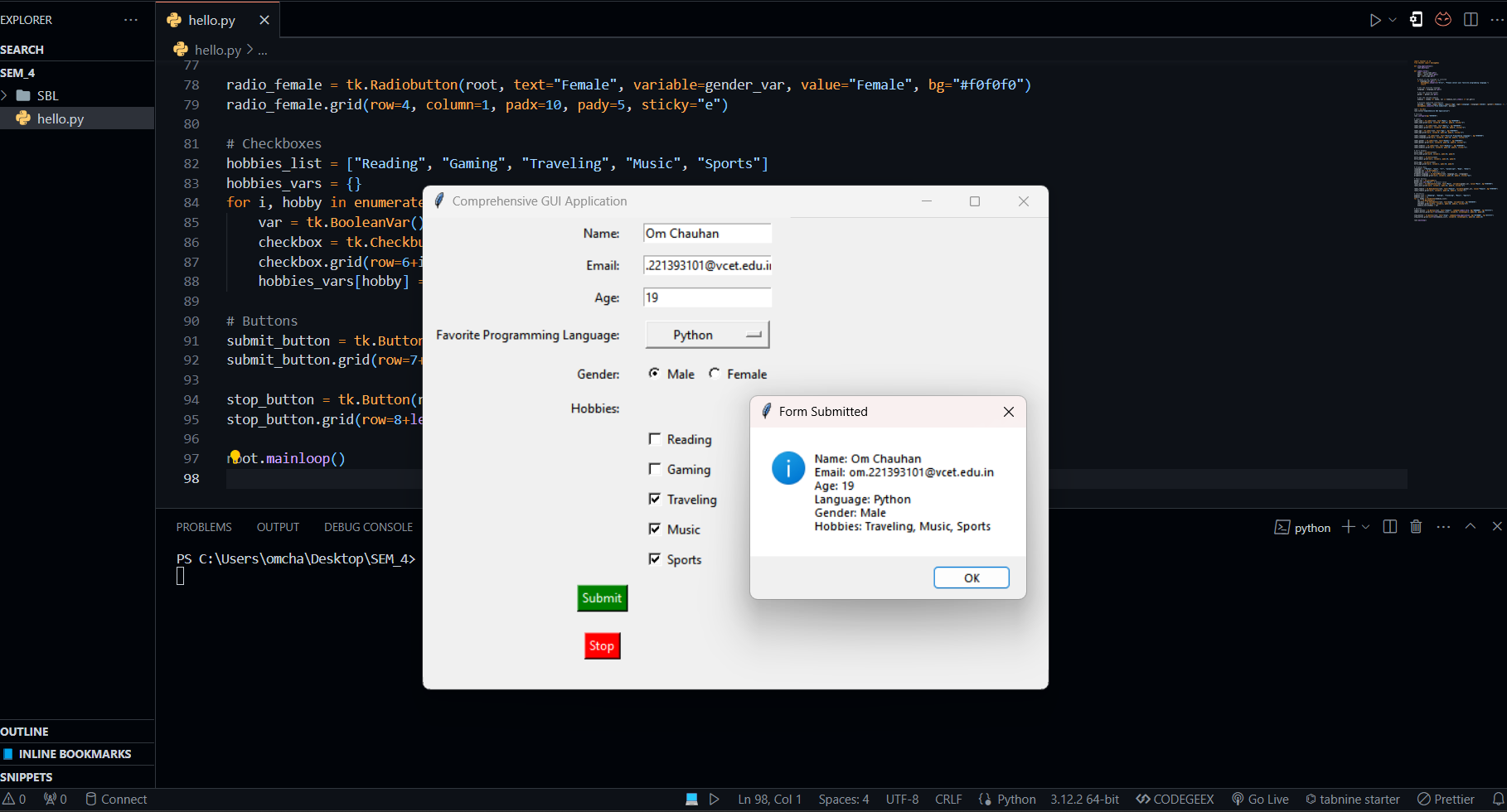
submit\_button.grid(row=7+len(hobbies\_list), column=0, columnspan=2, padx=10, pady=10)

stop\_button = tk.Button(root, text="Stop", command=stop\_application, bg="#ff0000", fg="#ffffff")

stop\_button.grid(row=8+len(hobbies\_list), column=0, columnspan=2, padx=10, pady=10)

root.mainloop()

**Output:**



**Conclusion:**

Experiment No. 8 provided a thorough exploration of creating GUI applications using Python's tkinter library. By incorporating a range of widgets including labels, textboxes, radio buttons, checkboxes, and custom dialog boxes, an interactive form was developed. This experiment served as an effective introduction to GUI development in Python, highlighting its versatility and user-friendly nature. The hands-on experience offered valuable insights into the capabilities of tkinter and its practical applications in building graphical interfaces for Python programs.