Assignment No : 3

Title: GUI in Python

Problem Statement: Design a user interface in Python

Learning Objectives:

✓ To design a user interface in Python

✓ To learn simplicity, user centric approach of a GUI in designing

Learning Outcomes:

A simple GUI designed using Tkinter library in Python.

Requirements:

Tkinter - standard GUI library for Python

Implementation Steps:

1. import the Python GUI Tkinter module: from tkinter import *

2. Define a function called show_successful_popup

. This function is intended to

display a "Thank You" message with the user's name when called.

- Get the user's name from an Entry widget (enter_1) using enter_1.get().
- Create a new window (Top level) for the "Thank You" message, set its size and
- Add a Label widget with the "Thank You" message that includes the user's name.
- Add a "Close" Button widget to close the "Thank You" window.

3. Create the main window (base) for the feedback form, set its size and title.

```
base = Tk()
base.geometry("550x550")
base.title('Feedback form')
```

4. Create labels for "Name," "Middle Name," “Last Name” "Branch," "Courses," and

5. Create a dropdown menu (OptionMenu) for selecting the branch.

6. Create entry widgets for the user's name ,middle name ,last name

7. Create radio buttons for class as F.E, S.E, T.E, B.E.

8. Create checkboxes for selecting Corses HCI, CN,DBMS, SPOS

9. Create a "Submit" button that, when clicked, calls the show_successful_popup.

10. Start the main GUI event loop to handle user interactions.

base.mainloop()

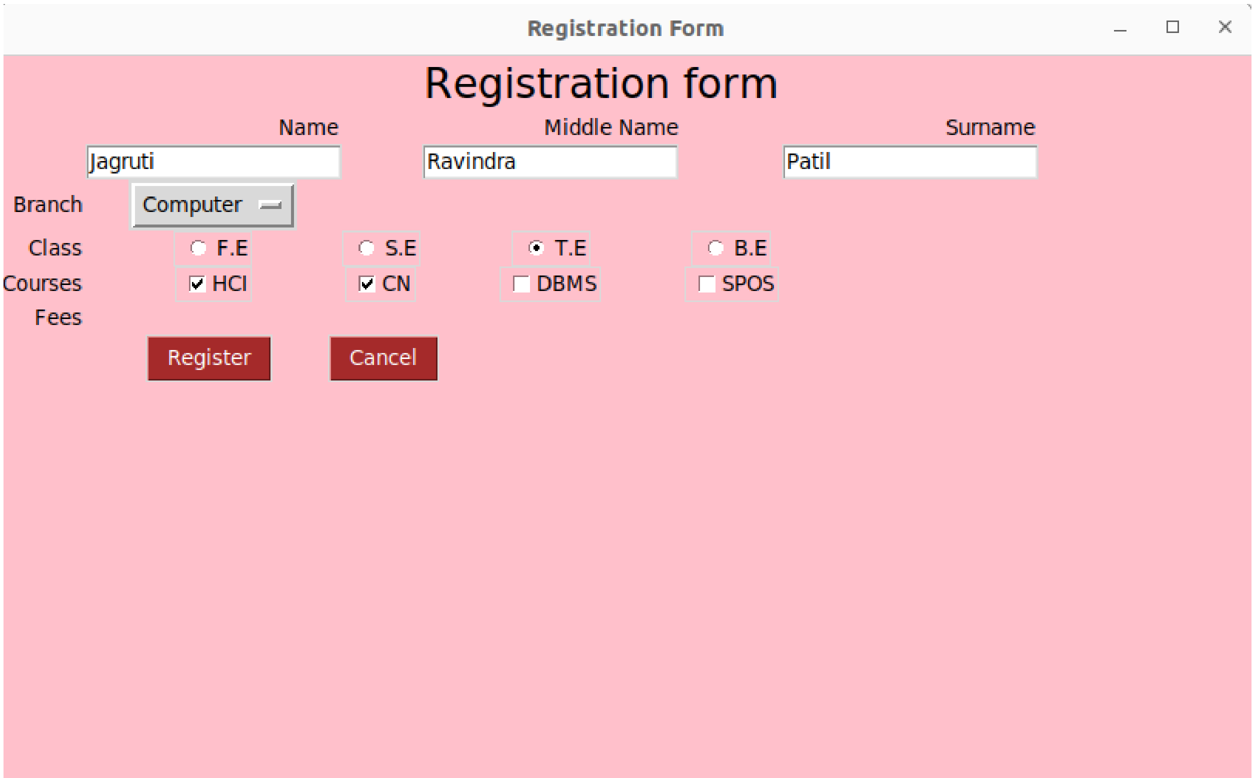
Implementation:

```
from tkinter import *
from tkinter import messagebox

def show_popup(title, message):
messagebox.showinfo(title, message)

def show_successful_popup():
name = entry_1.get()
middle_name = entry_6.get()
surname = entry_7.get()
branch = var_branch.get()

courses_selected = [course_names[i] for i, var in enumerate(course_vars) if var.get() == 1]
courses_text = ", ".join(courses_selected)

# Calculate fees based on the number of selected courses
fees = len(courses_selected) * 1000

if name and middle_name and surname :
registration_info = f"Name: {name} {middle_name} {surname}\nBranch: {branch}\nCourses: {courses_text}\nFees: {fees} USD"
success_message = f"Congratulations! Registration Successful!\n\n{registration_info}"

show_popup("Successful Registration", success_message)
else:
show_popup("Registration Error", "Please fill in all the required fields.")

root = Tk()
root.geometry('600x400')
root.title("Registration Form")
root.configure(bg='pink')

# Labels for the form
Label(root, text="Registration form", justify="center", font=("bold", 20), bg='pink').grid(row=0, column=3, columnspan=2)

Label(root, text="Name", bg='pink').grid(row=1, column=1, sticky='e')
Label(root, text="Middle Name", bg='pink').grid(row=1, column=3, sticky='e')
Label(root, text="Surname", bg='pink').grid(row=1, column=5, sticky='e')

Label(root, text="", bg='pink').grid(row=2, column=5, sticky='e')

Label(root, text="Branch", bg='pink').grid(row=4, column=0, sticky='e')
Label(root, text="Class", bg='pink').grid(row=5, column=0, sticky='e')
Label(root, text="Courses", bg='pink').grid(row=6, column=0, sticky='e')
Label(root, text="Fees", bg='pink').grid(row=7, column=0, sticky='e')
```

```
# Entry fields
entry_1 = Entry(root)
entry_1.grid(row=2, column=1)
entry_6 = Entry(root)
entry_6.grid(row=2, column=3)
entry_7 = Entry(root)
entry_7.grid(row=2, column=5)

# Branch Dropdown
branches = ["Computer", "IT", "Mechanical", "Electrical", "Civil"]
var_branch = StringVar(root)
var_branch.set(branches[0])
OptionMenu(root, var_branch, *branches).grid(row=4, column=1)

# Class Radiobuttons
class_var = StringVar()
classes = ["F.E", "S.E", "T.E", "B.E"]
for i, cls in enumerate(classes):
Radiobutton(root, text=cls, variable=class_var, value=cls, bg='pink').grid(row=5, column=1 + i)

# Courses Checkboxes
course_vars = [IntVar() for _ in range(4)]
course_names = ["HCI", "CN", "DBMS", "SPOS"]
for i, course_var in enumerate(course_vars):
Checkbutton(root, text=course_names[i], variable=course_var, bg='pink').grid(row=6, column=1 + i)

# Calculate and display fees
fees_label = Label(root, text="", bg='pink')
fees_label.grid(row=7, column=1)

# Register and Cancel buttons
Button(root, text='Register', bg='brown', fg='white', command=show_successful_popup).grid(row=8, column=0, columnspan=3)
Button(root, text='Cancel', bg='brown', fg='white').grid(row=8, column=1, columnspan=3)

root.mainloop()
```

Output: Images of GUI as

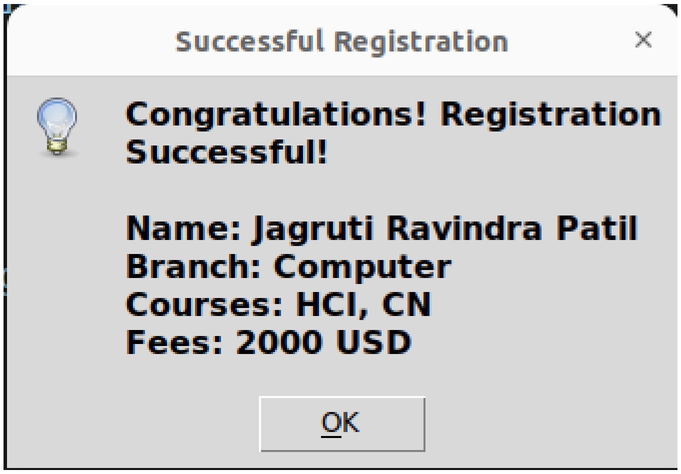
User Interface I:



Registration form

Name: Jagruti
Middle Name: Ravindra
Surname: Patil
Branch: Computer
Class: ○ F.E ○ S.E ● T.E ○ B.E
Courses: ☑ HCI ☑ CN ☐ DBMS ☐ SPOS
Fees

User Interface II:



Congratulations! Registration Successful!

Name: Jagruti Ravindra Patil
Branch: Computer
Courses: HCI, CN
Fees: 2000 USD

Conclusion: GUI created using Python Tkinter standard GUI library for Python