Laboratory Activity 4 - Introduction to GUI Development using Pycharm	
BONIFACIO, NYKO ADREIN L.	10 / 14 / 2024
CPE-009B / CpE21S4	Prof. Sayo

TASK

CODE:

registration.py

registration.py × main.py

```python
from tkinter import *


2 usages
class RegistrationWindow:
    def __init__(self, win):

        self.Label1 = Label(win, text="Account Registration", font=("Impact", 25), bg="Orange", fg="White")
        self.Label1.place(x=80, y=10)
        fields = ["First Name:", "Last Name:", "Username:", "Password:", "Email Address:", "Contact Number:"]
        self.entries = []
        y_position = 90
        for i, field in enumerate(fields):
            Label(win, text=field, font=("Arial", 14), anchor='w').place(x=50, y=y_position)
            entry = Entry(win, bd=5)
            entry.place(x=200, y=y_position)
            self.entries.append(entry)
            y_position += 40
        self.submit_btn = Button(win, text="Submit", bd=4, bg="LightGreen", font=("Arial", 14), command=self.submit)
        self.submit_btn.place(x=100, y=y_position + 20)
        self.clear_btn = Button(win, text="Clear", bd=4, bg="LightCoral", font=("Arial", 14), command=self.clear)
        self.clear_btn.place(x=200, y=y_position + 20)

    1 usage
    def submit(self):
        for entry in self.entries:
            print(entry.get())
    1 usage
    def clear(self):
        for entry in self.entries:
            entry.delete( first: 0,  last: 'end')
2 usages
def center_window(window, width=400, height=400):
    screen_width = window.winfo_screenwidth()
    screen_height = window.winfo_screenheight()
    x_coordinate = int((screen_width / 2) - (width / 2))
    y_coordinate = int((screen_height / 2) - (height / 2))
    window.geometry(f"{width}x{height}+{x_coordinate}+{y_coordinate}")

```

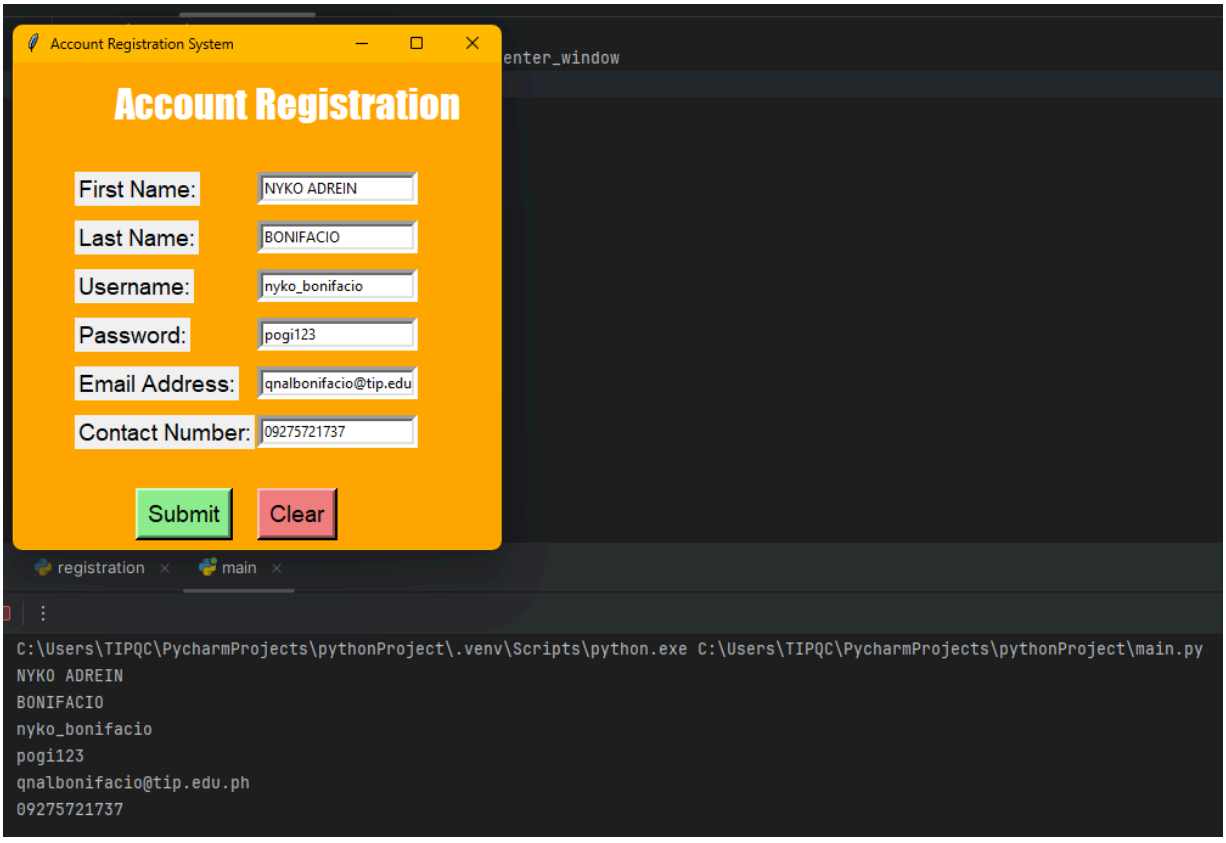
main.py

```python
from tkinter import Tk
from registration import RegistrationWindow, center_window

window = Tk()
app = RegistrationWindow(window)

center_window(window, width: 400, height: 400)
window.title("Account Registration System")
window.configure(bg='Orange')
window.mainloop()
```

OUTPUT:



QUESTIONS

1. What are the common GUI Applications that general end-users such as home users, students, and office employees use? (give at least 3 and describe each)

- Web browsers allow internet access and navigation, word processors enable easy document creation and editing, and spreadsheet software helps organize and analyze data in tables.

2. Based from your answer in question 1, why do you think home users, students, and office employees use those GUI programs?

- Users prefer these applications for their user-friendliness, productivity enhancements, collaboration features, and accessibility from various devices.

3. How does Pycharm help developers in making GUI applications, what would be the difference if developers made GUI programs without GUI Frameworks such as Pycharm or Tkinter?

- PyCharm provides an IDE with tools for code completion, debugging, and version control, streamlining development. Without GUI frameworks, creating components would be more complex and time-consuming.

4. What are the different platforms a GUI program may be created and deployed on? (Three is required then state why might a program be created on that specific platform)

- Windows is popular in business for its support of various tools, macOS caters to Apple users with native UI, and Linux offers open-source flexibility for niche audiences.

5. What is the purpose of app = QApplication(sys.argv) , ex = App() and sys.exit(app.exec_()) ?

- `app = QApplication(sys.argv)` initializes the app, `ex = App()` creates the main GUI instance, and `sys.exit(app.exec_())` starts the event loop and ensures a clean exit.

CONCLUSION

In conclusion, starting with basic and interactive application creation can be achieved by studying GUI development with PyCharm. Python graphical interface development is made easier by PyCharm's compatibility with Tkinter and PyQt. Through proficiency with buttons, windows, and other graphical user interface elements, we may enhance the usability of our programs. For new users, PyCharm's useful features like debugging and code suggestions facilitate the process. All things considered, it's a fantastic tool to have when beginning GUI creation.