Laboratory Activity No. 4	
CALVIN EARL PLANTA	10/14/2024
CPE21S4 / CPE009B	Prof. Sayo

6. Supplementary Activity

Task

Create an Object-Oriented GUI Application for a simple Account Registration System with the following required information:
first name, last name, username, password, email address, contact number.
Requirements:
• The GUI must be centered on your screen.
• The GUI Components should be organized according to the order of information required using Absolute Positioning.
• The position of the components should be automatically computed based on the top component.
• All the text fields should be accompanied with their corresponding label on the left side while the text field is on the
right side.
• There should be a program title other than the Window Title.
• There should be a submit button and clear button at the bottom (submit button on the left, clear button on the right).
• The program should be launched on main.py while the GUI Codes should be on a separate file called
registration.py

registration.py file

Python

```python
import sys
from PyQt5.QtWidgets import QWidget, QLineEdit, QPushButton,
QApplication, QLabel
from PyQt5.QtGui import QIcon


class App(QWidget):

   def __init__(self):
       super().__init__()
       self.title = "PyQt Registration"
       self.x = 200
       self.y = 200
       self.width = 300
```

```
        self.height = 300
        self.initUI()

    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(self.x, self.y, self.width, self.height)
        self.setWindowIcon(QIcon('pythonico.ico'))

        self.textboxlbl = QLabel("Registration", self)
        self.textboxlbl.move(120, 20)

        self.textboxlbl2 = QLabel("First name:", self)
        self.textboxlbl2.move(20, 70)
        self.textboxentry = QLineEdit(self)
        self.textboxentry.move(80, 70)

        self.textboxlbl3 = QLabel("Last name:", self)
        self.textboxlbl3.move(20, 100)
        self.textboxentry2 = QLineEdit(self)
        self.textboxentry2.move(80, 100)

        self.textboxlbl4 = QLabel("Username:", self)
        self.textboxlbl4.move(20, 130)
        self.textboxentry3 = QLineEdit(self)
        self.textboxentry3.move(80, 130)

        self.textboxlbl5 = QLabel("Password:", self)
        self.textboxlbl5.move(20, 160)
        self.textboxentry4 = QLineEdit(self)
        self.textboxentry4.move(80, 160)
        self.textboxentry4.setEchoMode(QLineEdit.Password)

        self.textboxlbl6 = QLabel("Email:", self)
        self.textboxlbl6.move(20, 190)
        self.textboxentry5 = QLineEdit(self)
        self.textboxentry5.move(80, 190)

        self.button = QPushButton('Submit', self)
        self.button.clicked.connect(self.submit)
        self.button.move(50, 230)

        self.result_label = QLabel("", self)
        self.result_label.move(80, 280)

        self.button2 = QPushButton('Clear', self)
```

```
        self.button2.move(150, 230)
        self.button2.clicked.connect(self.clear)

    def submit(self):
        first_name = self.textboxentry.text()
        last_name = self.textboxentry2.text()
        username = self.textboxentry3.text()
        password = self.textboxentry4.text()
        email = self.textboxentry5.text()
        self.result_label.setText("Registration successful!")
    def clear(self):
        self.textboxentry.clear()
        self.textboxentry2.clear()
        self.textboxentry3.clear()
        self.textboxentry4.clear()
        self.textboxentry5.clear()


if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = App()
    sys.exit(app.exec_())
```

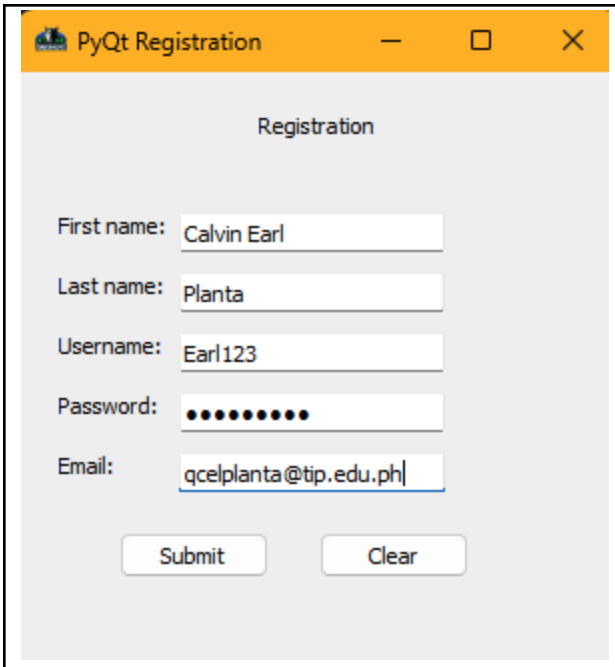
main.py file

Python

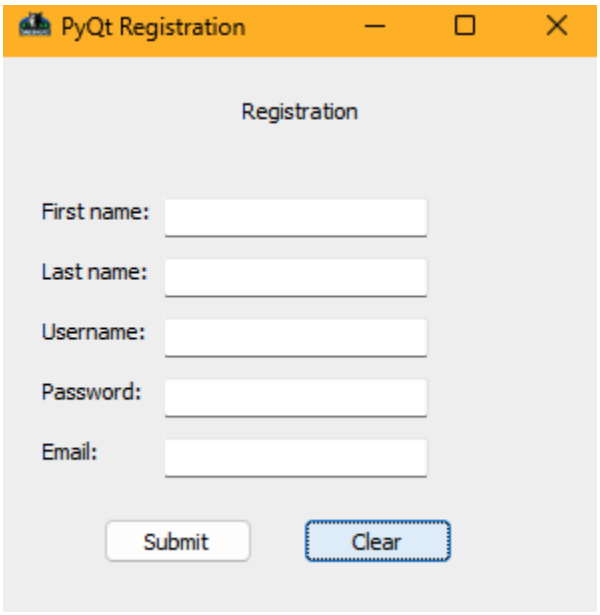
```
import sys
from PyQt5.QtWidgets import QWidget, QLineEdit, QPushButton,
QApplication, QLabel
from PyQt5.QtGui import QIcon
from registration import App

if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = App()
    ex.show()
    sys.exit(app.exec_())
```

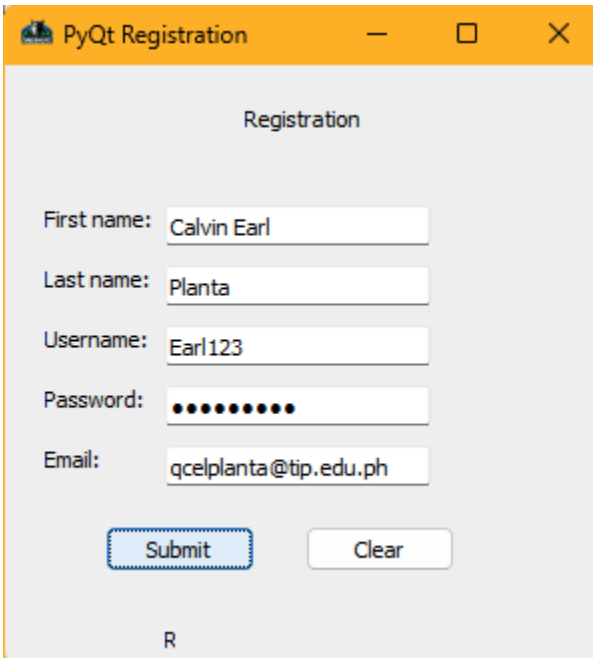
Output


PyQt Registration
Registration
First name:
Calvin Earl
Last name:
Planta
Username:
Earl123
Password:
Email:
qcelplanta@tip.edu.ph
Submit
Clear

clear


PyQt Registration
Registration
First name:
Last name:
Username:
Password:
Email:
Submit
Clear

submit



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)
    - **Typical GUI programs that office workers, students, and home users utilize include web browsers such as Microsoft Office Suite and Firefox, Safari, and Google Chrome programs such as Outlook, Word, Excel, and PowerPoint. Furthermore, a lot of people communicate with Linux distributions and other GUI-based operating systems such as Windows and MacOS.**
2. Based from your answer in question 1, why do you think home users, students, and office employees use those GUI programs?
    - **Due to the fact that these GUI programs provide an easy-to-use interface, easy to use computer systems, generate and edit documents, and access the internet Office professionals, students, and home users all use them. As a result, productivity rises. effectiveness as well as the whole computer experience.**
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?
    - **First off, PyCharm has a GUI Designer that lets programmers create graphically as well as create GUI apps using a drag-and-drop interface. Second, Code Completion is offered by PyCharm. It makes feasible suggestions to developers so they can write code more quickly. variables, methods, and classes completions. Lastly, Code Inspection is**

a feature of PyCharm that examines the code for possible mistakes. alerts, as well as upgrades. This function aids developers in finding and fixing problems at an early stage, making their graphical user interface (GUI) more dependable and robust. Finally, PyCharm offers Debugging Tools so that programmers may troubleshoot more successfully with their GUI application. These resources enable developers to specify go through their code, check variables, and set breakpoints, which make it simpler to determine and resolve problems

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)
    - **One can build and implement a graphical user interface (GUI) on multiple platforms, such as Windows. OS X, Linux, mobile devices (iOS and Android), and the internet as a whole programs, as well as on cloud servers. Certain GUI applications could be made on particular platforms because of the extensive use and compatibility of Windows, the svelte design and user-friendly interface, the flexibility and open-source nature of Linux, and the growing the necessity for accessibility on the go, the requirement for accessibility on the web from any location, and the features of accessibility, security, and scalability of the cloud.**
5. What is the purpose of app = QApplication(sys.argv), ex = App(), and sys.exit(app.exec_())?
    - **sys.exit(app.exec_()), ex = App(), and app = QApplication(sys.argv) are used to construct a launch the main application window, launch the Qt application instance, and initiate the application's event loop, respectively), which when combined allow the GUI program to operate and react to user exchanges.**

7. Conclusion

We talked about using Python and the Tkinter package to create a GUI application for a basic account registration system. We discussed a number of topics related to developing GUI applications during our talk, such as code organization, event processing, as well as input verification. The completed implementation produced an operational Account Registration System and complied with the requirements.