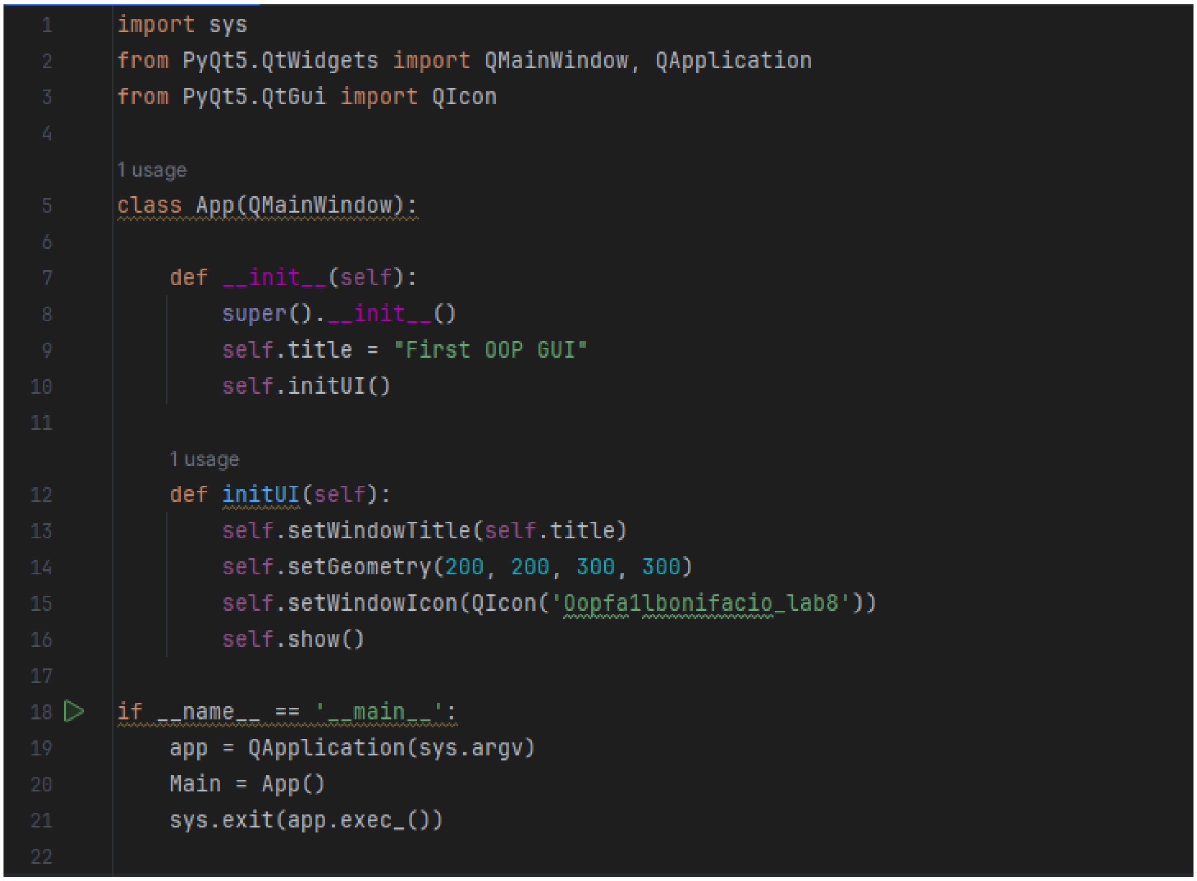
Laboratory Activity No. 4 – Introduction to GUI Development using Pycharm	
Bonifacio, Redj Guillian F.	17/10/2024
CPE009 / CPE21S1	Sayo, Maria Rizette

5. Procedure:

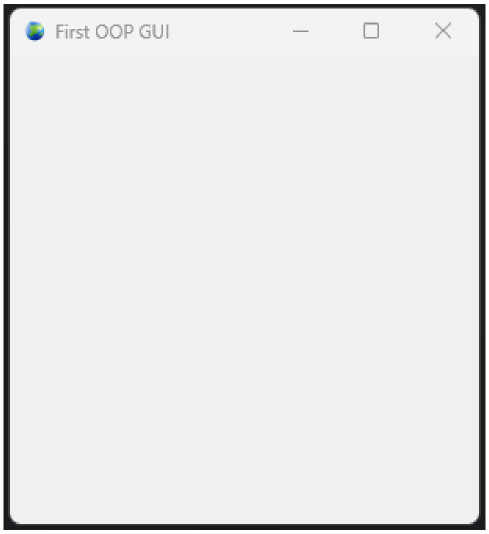
gui_window.py

Source Code



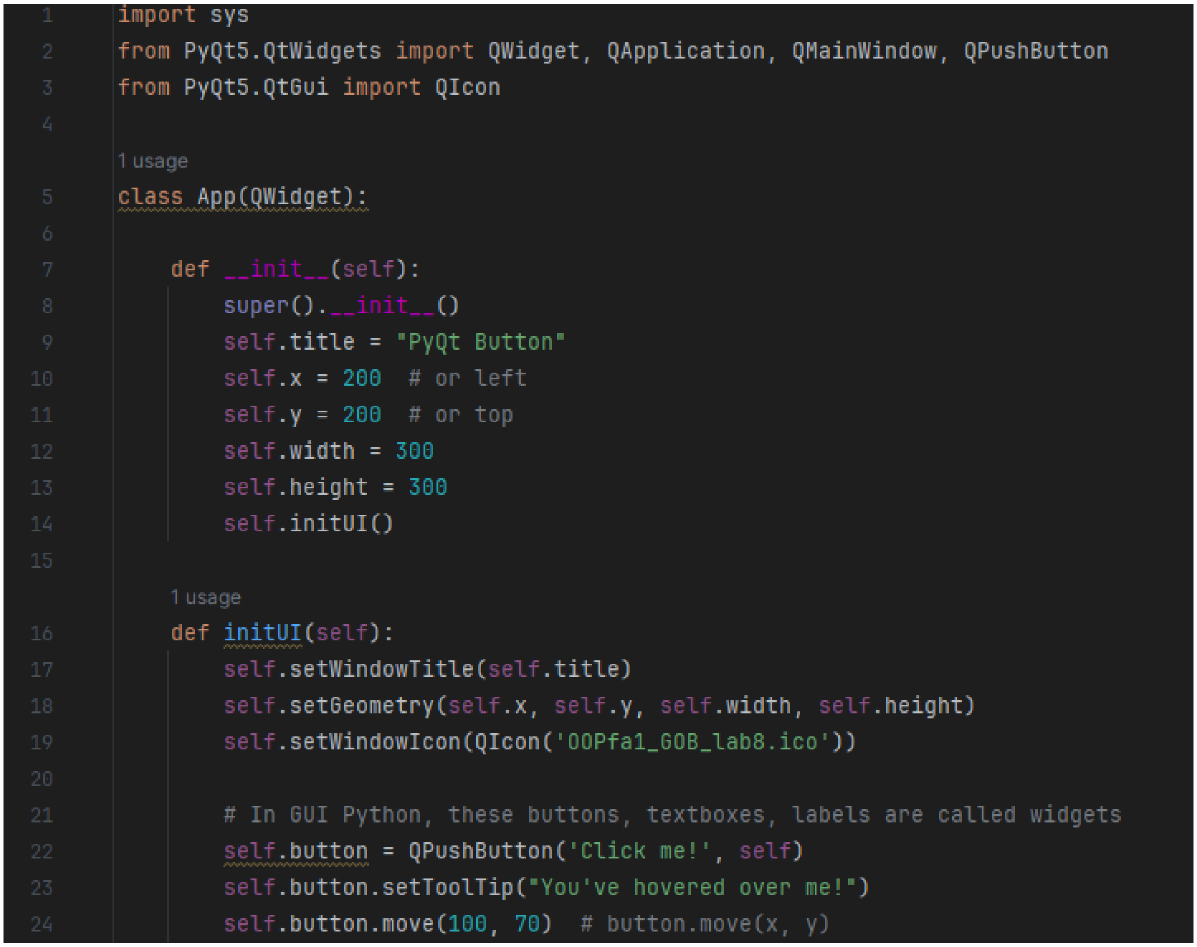
```python
import sys
from PyQt5.QtWidgets import QMainWindow, QApplication
from PyQt5.QtGui import QIcon

1 usage
class App(QMainWindow):

    def __init__(self):
        super().__init__()
        self.title = "First OOP GUI"
        self.initUI()

    1 usage
    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(200, 200, 300, 300)
        self.setWindowIcon(QIcon('Oopfa1lbonifacio_lab8'))
        self.show()

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

```

Output

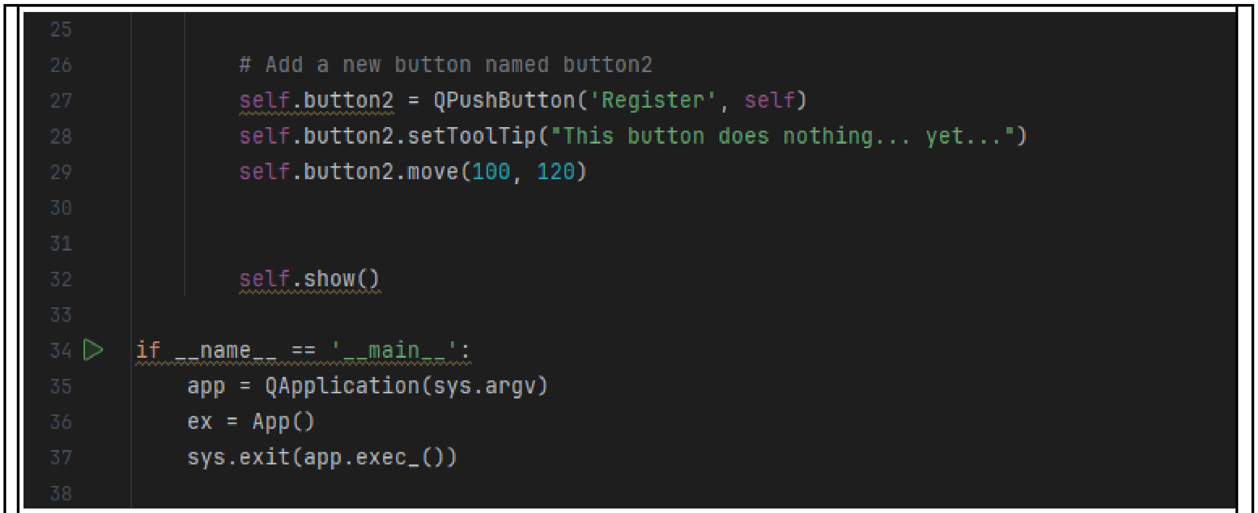


gui_buttons.py

Code

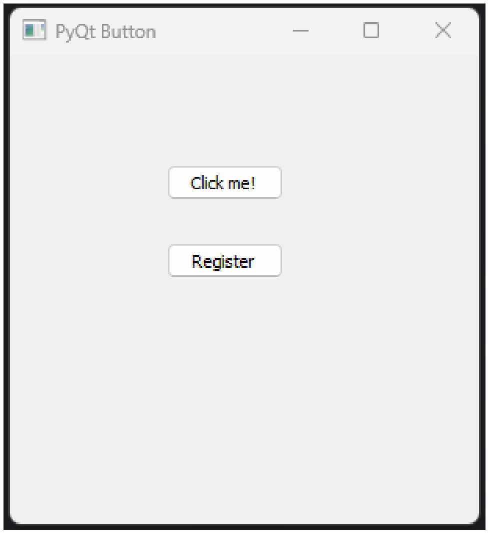


```
import sys
from PyQt5.QtWidgets import QWidget, QApplication, QMainWindow, QPushButton
from PyQt5.QtGui import QIcon

1 usage
class App(QWidget):

    def __init__(self):
        super().__init__()
        self.title = "PyQt Button"
        self.x = 200  # or left
        self.y = 200  # or top
        self.width = 300
        self.height = 300
        self.initUI()

    1 usage
    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(self.x, self.y, self.width, self.height)
        self.setWindowIcon(QIcon('OOPfa1_GOB_lab8.ico'))

        # In GUI Python, these buttons, textboxes, labels are called widgets
        self.button = QPushButton('Click me!', self)
        self.button.setToolTip("You've hovered over me!")
        self.button.move(100, 70)  # button.move(x, y)
```



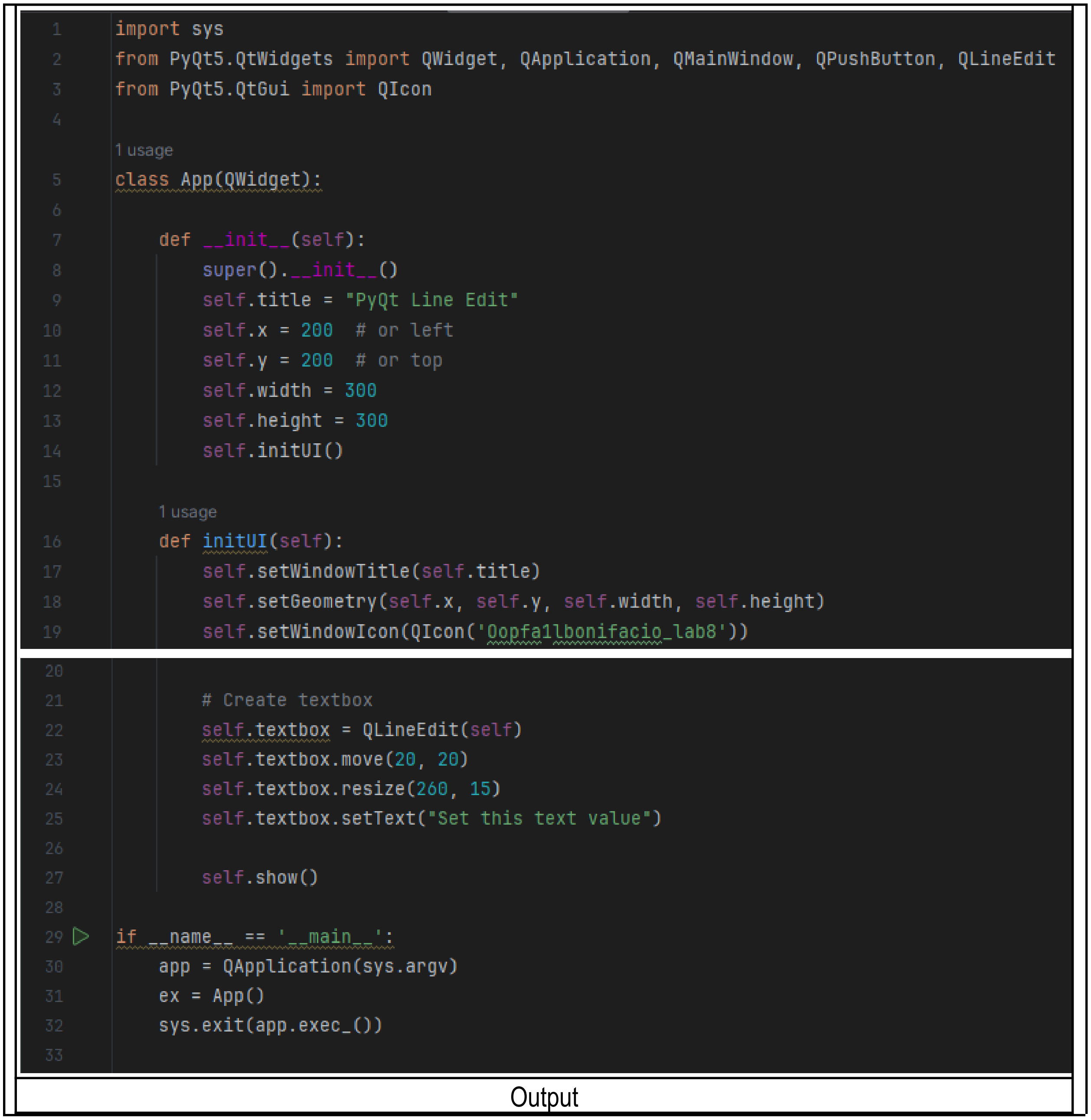
```python

        # Add a new button named button2
        self.button2 = QPushButton('Register', self)
        self.button2.setToolTip("This button does nothing... yet...")
        self.button2.move(100, 120)


        self.show()

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

```

Output



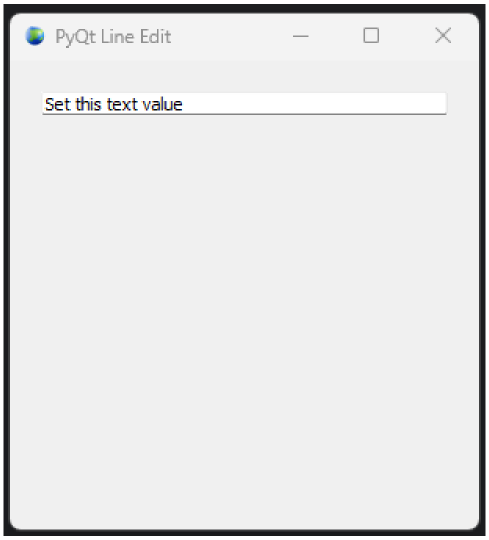
gui_text.py

Code



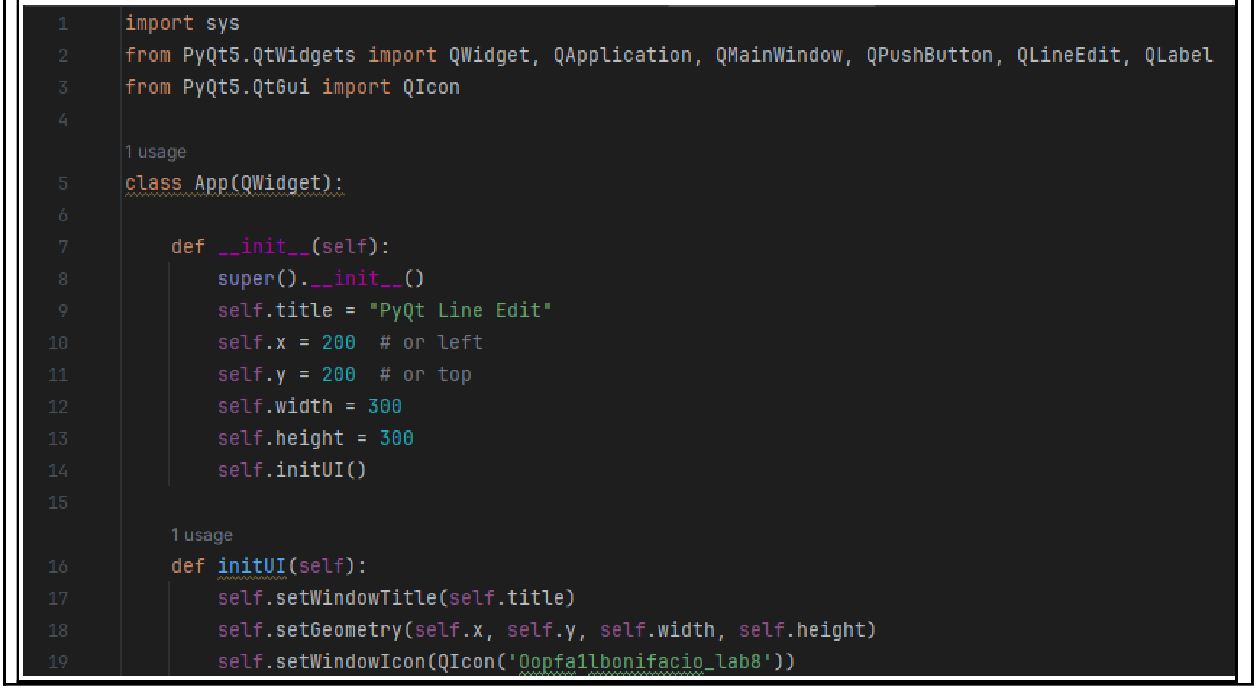
```
import sys
from PyQt5.QtWidgets import QWidget, QApplication, QMainWindow, QPushButton, QLineEdit
from PyQt5.QtGui import QIcon

1 usage
class App(QWidget):

    def __init__(self):
        super().__init__()
        self.title = "PyQt Line Edit"
        self.x = 200  # or left
        self.y = 200  # or top
        self.width = 300
        self.height = 300
        self.initUI()

    1 usage
    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(self.x, self.y, self.width, self.height)
        self.setWindowIcon(QIcon('Oopfa1lbonifacio_lab8'))

        # Create textbox
        self.textbox = QLineEdit(self)
        self.textbox.move(20, 20)
        self.textbox.resize(260, 15)
        self.textbox.setText("Set this text value")

        self.show()

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

Output

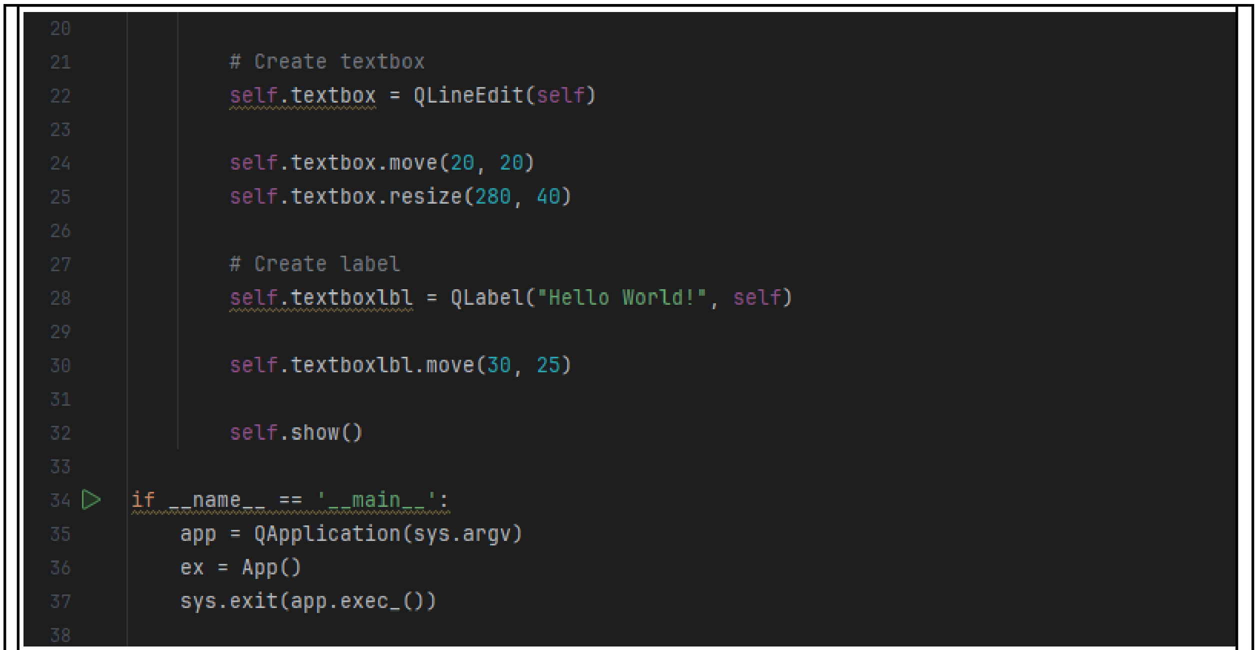


Set this text value

gui_labels.py
Code

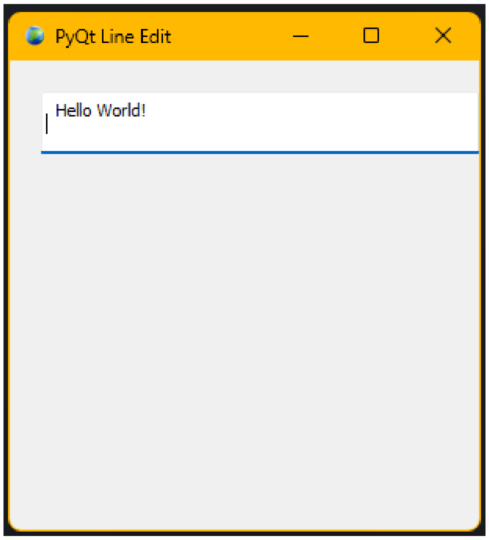


```
import sys
from PyQt5.QtWidgets import QWidget, QApplication, QMainWindow, QPushButton, QLineEdit, QLabel
from PyQt5.QtGui import QIcon

1 usage
class App(QWidget):

    def __init__(self):
        super().__init__()
        self.title = "PyQt Line Edit"
        self.x = 200  # or left
        self.y = 200  # or top
        self.width = 300
        self.height = 300
        self.initUI()

    1 usage
    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(self.x, self.y, self.width, self.height)
        self.setWindowIcon(QIcon('Oopfa1lbonifacio_lab8'))
```



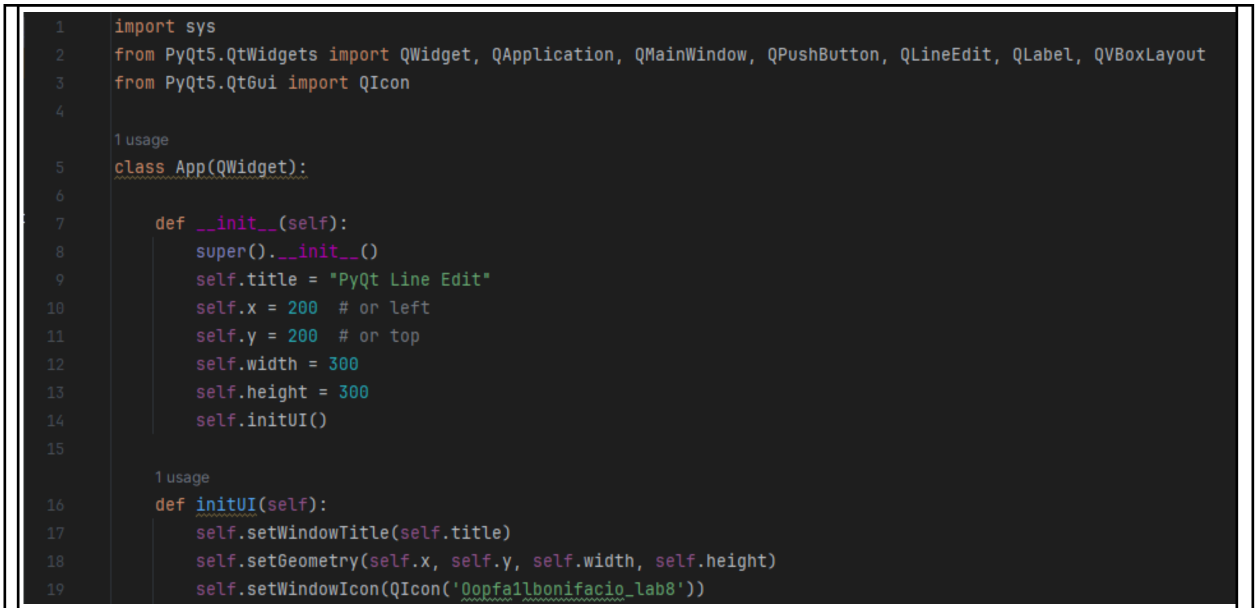
```python

        # Create textbox
        self.textbox = QLineEdit(self)

        self.textbox.move(20, 20)
        self.textbox.resize(280, 40)

        # Create label
        self.textboxlbl = QLabel("Hello World!", self)

        self.textboxlbl.move(30, 25)

        self.show()

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

```

Output

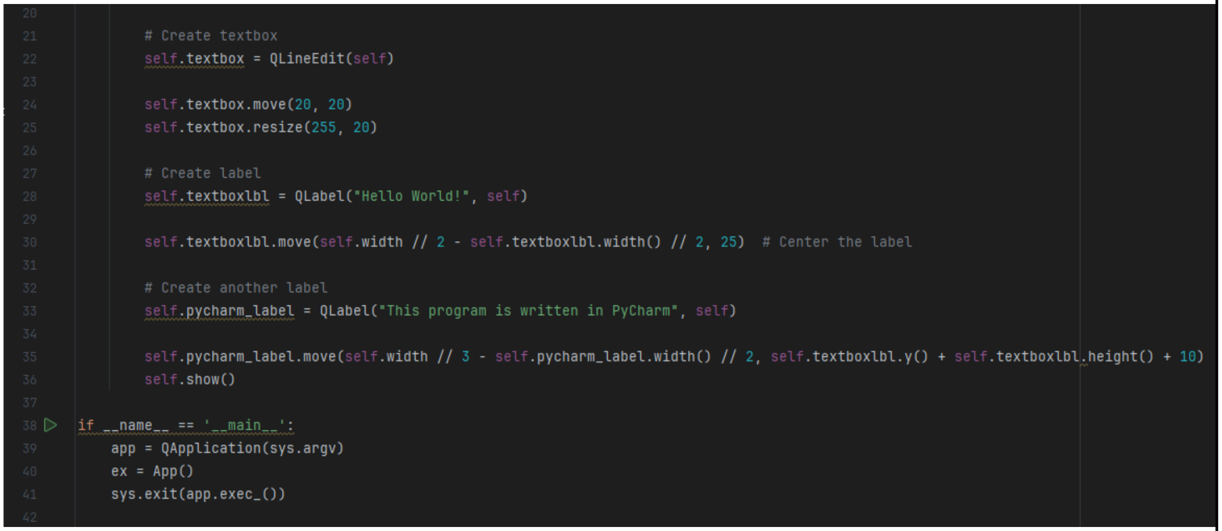


testing.py

Code

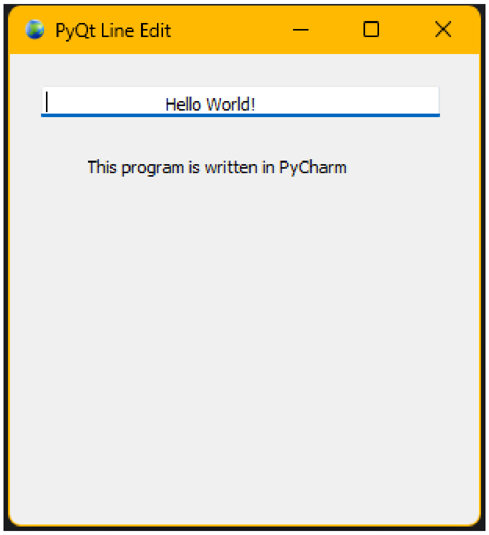


```python
import sys
from PyQt5.QtWidgets import QWidget, QApplication, QMainWindow, QPushButton, QLineEdit, QLabel, QVBoxLayout
from PyQt5.QtGui import QIcon

1 usage
class App(QWidget):

    def __init__(self):
        super().__init__()
        self.title = "PyQt Line Edit"
        self.x = 200  # or left
        self.y = 200  # or top
        self.width = 300
        self.height = 300
        self.initUI()

    1 usage
    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(self.x, self.y, self.width, self.height)
        self.setWindowIcon(QIcon('Oopfa1lbonifacio_lab8'))
```



```python

        # Create textbox
        self.textbox = QLineEdit(self)

        self.textbox.move(20, 20)
        self.textbox.resize(255, 20)

        # Create label
        self.textboxlbl = QLabel("Hello World!", self)

        self.textboxlbl.move(self.width // 2 - self.textboxlbl.width() // 2, 25)  # Center the label

        # Create another label
        self.pycharm_label = QLabel("This program is written in PyCharm", self)

        self.pycharm_label.move(self.width // 3 - self.pycharm_label.width() // 2, self.textboxlbl.y() + self.textboxlbl.height() + 10)
        self.show()

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

Output



PyQt Line Edit

Hello World!

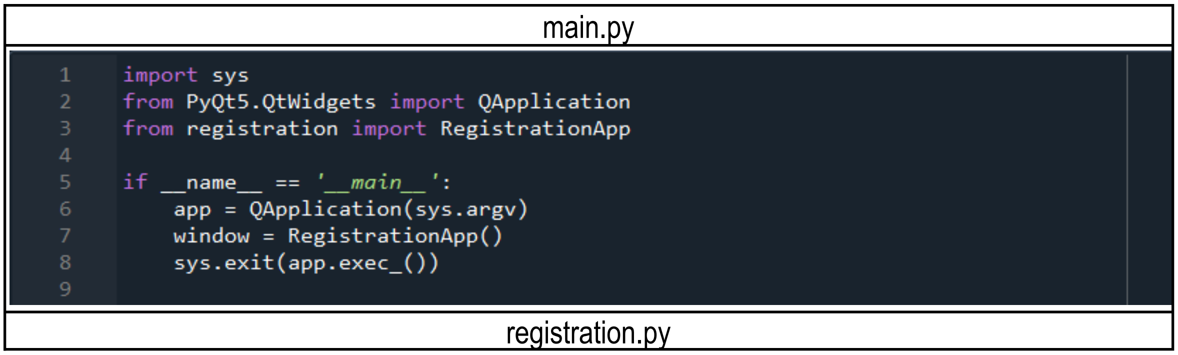
This program is written in PyCharm

6. Supplementary Activity:

Account Registration System

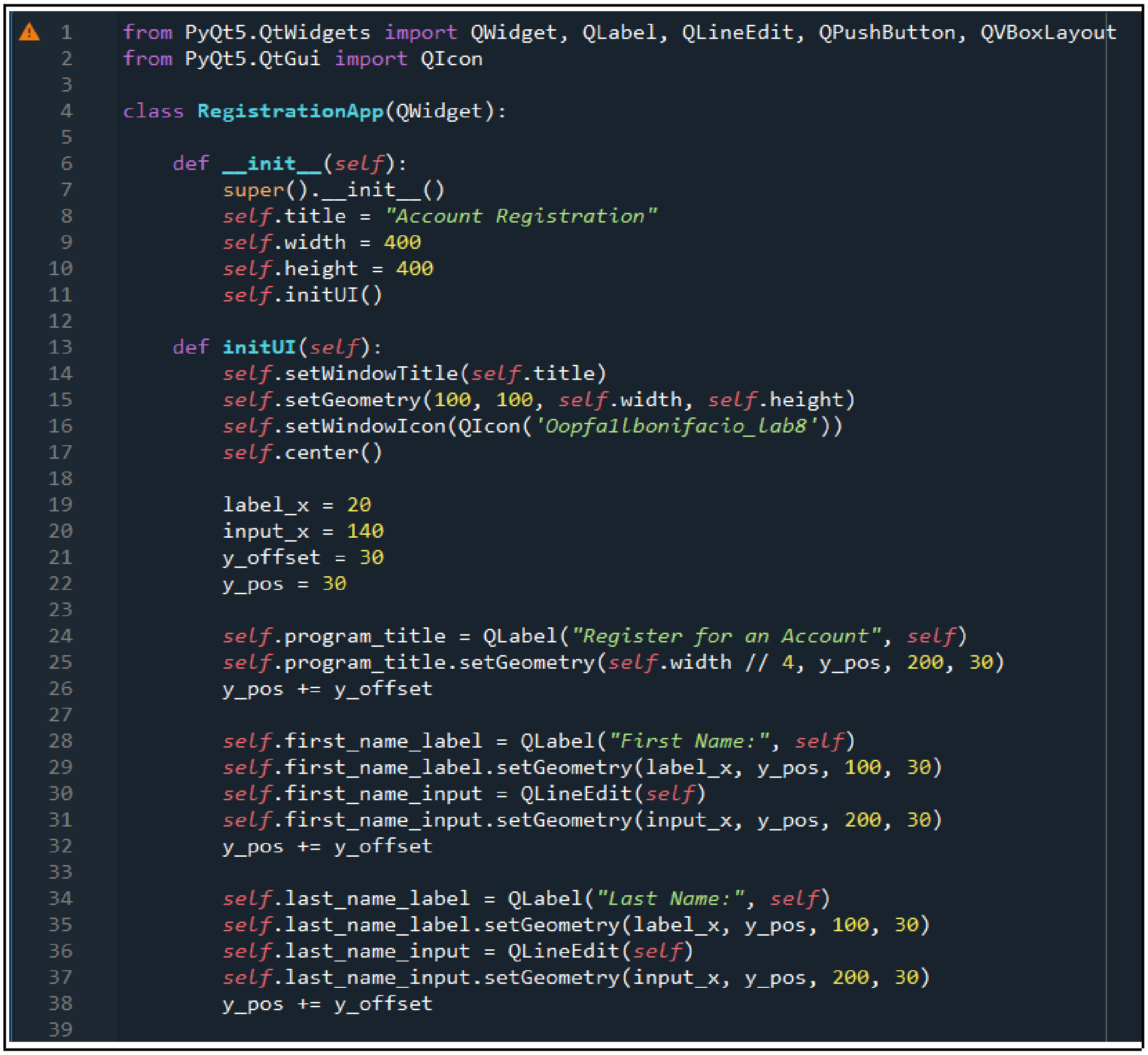
Source Code

main.py

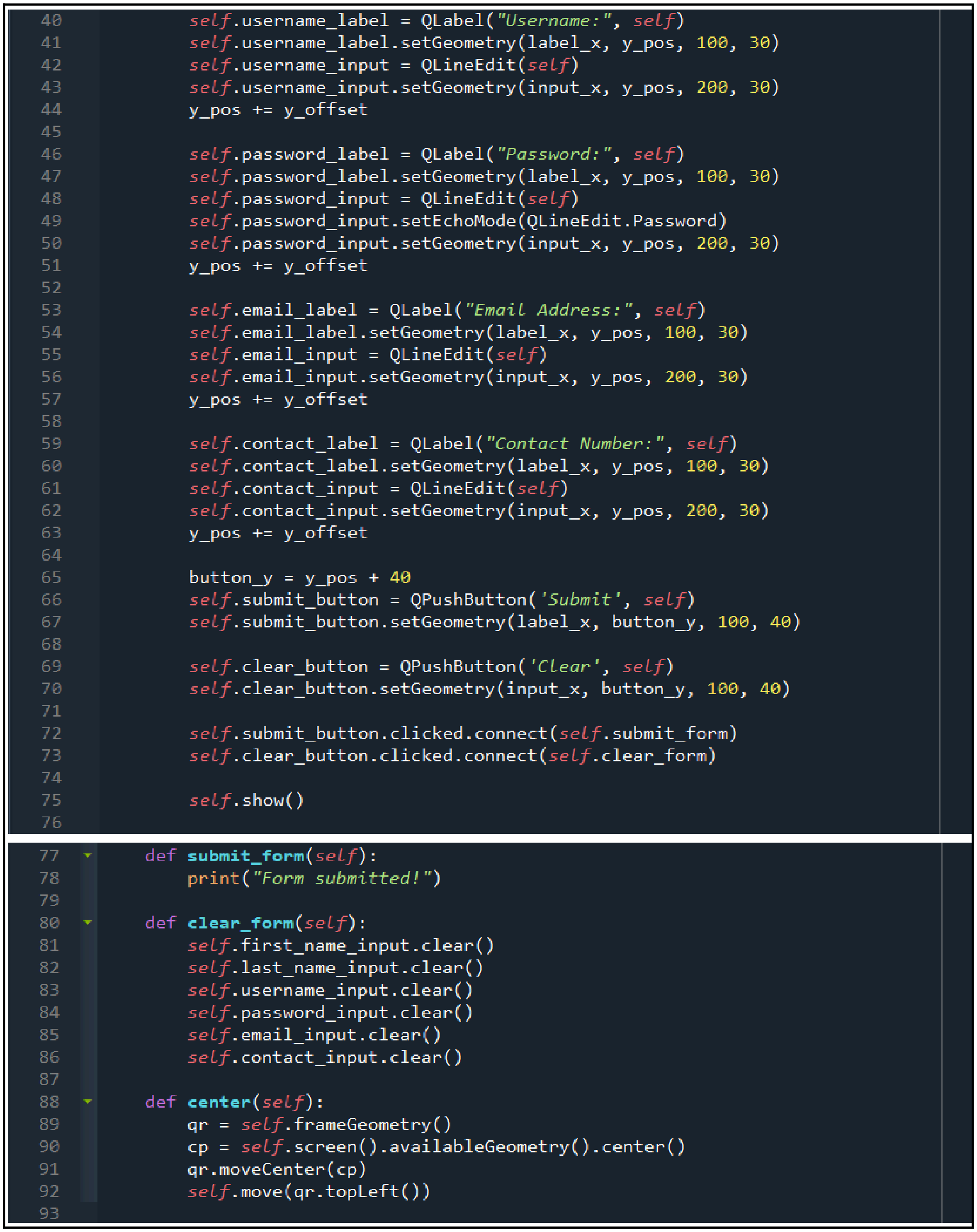


```python
import sys
from PyQt5.QtWidgets import QApplication
from registration import RegistrationApp

if __name__ == '__main__':
    app = QApplication(sys.argv)
    window = RegistrationApp()
    sys.exit(app.exec_())

```

registration.py

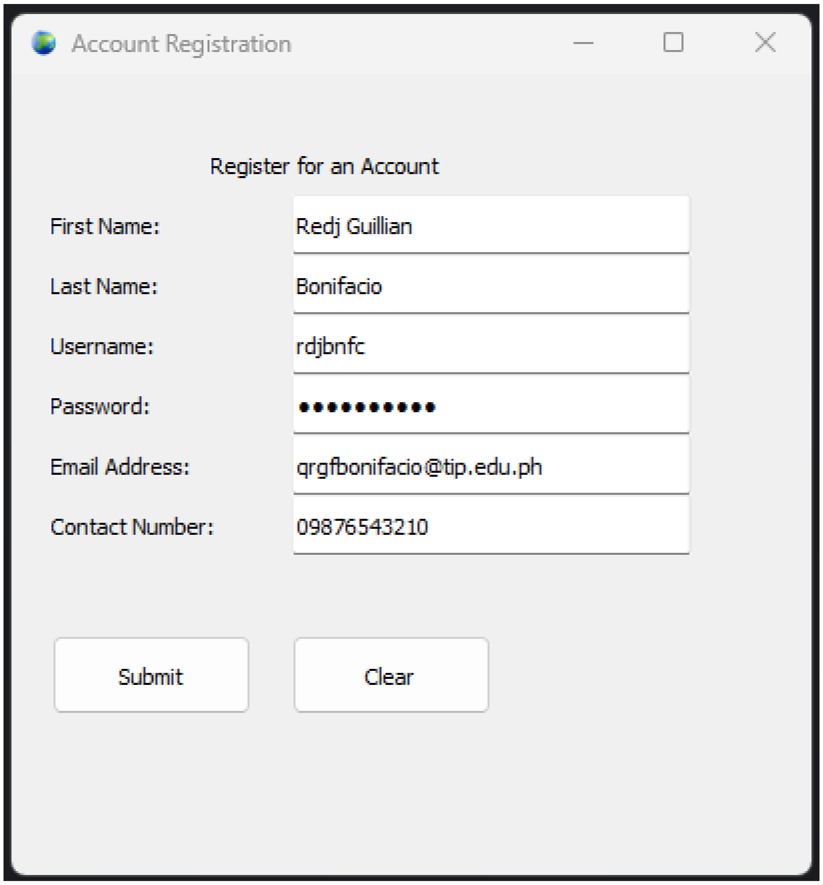


```python
from PyQt5.QtWidgets import QWidget, QLabel, QLineEdit, QPushButton, QVBoxLayout
from PyQt5.QtGui import QIcon

class RegistrationApp(QWidget):

    def __init__(self):
        super().__init__()
        self.title = "Account Registration"
        self.width = 400
        self.height = 400
        self.initUI()

    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(100, 100, self.width, self.height)
        self.setWindowIcon(QIcon('Oopfa1lbonifacio_lab8'))
        self.center()

        label_x = 20
        input_x = 140
        y_offset = 30
        y_pos = 30

        self.program_title = QLabel("Register for an Account", self)
        self.program_title.setGeometry(self.width // 4, y_pos, 200, 30)
        y_pos += y_offset

        self.first_name_label = QLabel("First Name:", self)
        self.first_name_label.setGeometry(label_x, y_pos, 100, 30)
        self.first_name_input = QLineEdit(self)
        self.first_name_input.setGeometry(input_x, y_pos, 200, 30)
        y_pos += y_offset

        self.last_name_label = QLabel("Last Name:", self)
        self.last_name_label.setGeometry(label_x, y_pos, 100, 30)
        self.last_name_input = QLineEdit(self)
        self.last_name_input.setGeometry(input_x, y_pos, 200, 30)
        y_pos += y_offset

```



```python
        self.username_label = QLabel("Username:", self)
        self.username_label.setGeometry(label_x, y_pos, 100, 30)
        self.username_input = QLineEdit(self)
        self.username_input.setGeometry(input_x, y_pos, 200, 30)
        y_pos += y_offset

        self.password_label = QLabel("Password:", self)
        self.password_label.setGeometry(label_x, y_pos, 100, 30)
        self.password_input = QLineEdit(self)
        self.password_input.setEchoMode(QLineEdit.Password)
        self.password_input.setGeometry(input_x, y_pos, 200, 30)
        y_pos += y_offset

        self.email_label = QLabel("Email Address:", self)
        self.email_label.setGeometry(label_x, y_pos, 100, 30)
        self.email_input = QLineEdit(self)
        self.email_input.setGeometry(input_x, y_pos, 200, 30)
        y_pos += y_offset

        self.contact_label = QLabel("Contact Number:", self)
        self.contact_label.setGeometry(label_x, y_pos, 100, 30)
        self.contact_input = QLineEdit(self)
        self.contact_input.setGeometry(input_x, y_pos, 200, 30)
        y_pos += y_offset

        button_y = y_pos + 40
        self.submit_button = QPushButton('Submit', self)
        self.submit_button.setGeometry(label_x, button_y, 100, 40)

        self.clear_button = QPushButton('Clear', self)
        self.clear_button.setGeometry(input_x, button_y, 100, 40)

        self.submit_button.clicked.connect(self.submit_form)
        self.clear_button.clicked.connect(self.clear_form)

        self.show()

    def submit_form(self):
        print("Form submitted!")

    def clear_form(self):
        self.first_name_input.clear()
        self.last_name_input.clear()
        self.username_input.clear()
        self.password_input.clear()
        self.email_input.clear()
        self.contact_input.clear()

    def center(self):
        qr = self.frameGeometry()
        cp = self.screen().availableGeometry().center()
        qr.moveCenter(cp)
        self.move(qr.topLeft())

```

Output



Account Registration

Register for an Account

First Name: Redj Guillian

Last Name: Bonifacio

Username: rdjbnfc

Password: •••••••••••

Email Address: qrgfbonifacio@tip.edu.ph

Contact Number: 09876543210

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)

Microsoft Word or Google Docs

- Word processors allow users to create, edit, and format text documents. Home users use them for personal tasks, students for assignments, and office employees for formal documents

Web Browsers (Chrome, Firefox)

- Web browsers allow users to access the internet. They are used by home users for browsing, students for research, and office employees for work-related tasks such as email and document sharing

File Explorer (Windows Explorer)

- File explorers help users navigate their file systems, open files, and manage directories. Home users, students, and office workers use them to organize personal, school, or work documents

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

   It makes interaction with the program itself easier. The use of GUI programs are intuitive, which does not need any experience in programming to operate the program. Additionally, for students and office employees, programs such Google Docs and web browsers support collaboration and sharing of work. It can make doing the task more efficient and manageable as it simplifies common tasks such document creation, web browsing, and file management

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?

   People use these GUI programs because they're easy to navigate and handle everyday tasks like writing documents or browsing the web effortlessly. They also make collaboration simple and work smoothly across different devices and platforms.

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?

   From personal experience, PyCharm makes building GUI apps much easier by providing tools for debugging and working with frameworks like PyQt. Without it, I'd have to deal with a lot of tedious, low-level tasks that slow down development

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

   GUI programs can be developed for Windows, macOS, or Linux. Windows is widely used, macOS is preferred for creative tools, and Linux is favored for its flexibility among developers

7. Conclusion:

In this lab, I learned about GUI development using PyCharm and explored key components like buttons and text fields. I found that PyCharm makes the development process easier and discovered the benefits of GUI frameworks. This knowledge will help us as we move on to object-oriented programming and more complex projects.