Laboratory Activity 4 - Introduction to GUI Development using Pycharm	
Titong, Lee Ivan B.	10/20/2024
Course/Section BSCPE21S4	Prof. Maria Rizette Sayo

Python

#registration.py

```python
# registration.py
import sys
from PyQt5.QtWidgets import QWidget, QApplication, QLabel, QLineEdit,
QPushButton, QVBoxLayout, QFormLayout

class RegistrationForm(QWidget):
    def __init__(self):
        super().__init__()
        self.title = "Account Registration"
        self.initUI()

    def initUI(self):
        self.setWindowTitle(self.title)
        self.setGeometry(100, 100, 400, 300)  # Set the window size
        self.setFixedSize(400, 300)  # Fixed size for the window

        # Create layout
        layout = QFormLayout()

        # Create labels and text fields
        self.first_name = QLineEdit(self)
        self.last_name = QLineEdit(self)
        self.username = QLineEdit(self)
        self.password = QLineEdit(self)
        self.password.setEchoMode(QLineEdit.Password)  # Hide password
input
        self.email = QLineEdit(self)
        self.contact_number = QLineEdit(self)

        # Add fields to the layout
        layout.addRow(QLabel("First Name:"), self.first_name)
        layout.addRow(QLabel("Last Name:"), self.last_name)
        layout.addRow(QLabel("Username:"), self.username)
        layout.addRow(QLabel("Password:"), self.password)
        layout.addRow(QLabel("Email Address:"), self.email)
        layout.addRow(QLabel("Contact Number:"), self.contact_number)

        # Create buttons
```

```
        self.submit_button = QPushButton("Submit", self)
        self.clear_button = QPushButton("Clear", self)

        # Connect buttons to functions
        self.submit_button.clicked.connect(self.submit_form)
        self.clear_button.clicked.connect(self.clear_form)

        # Add buttons to the layout
        layout.addRow(self.submit_button, self.clear_button)

        # Set the layout for the widget
        self.setLayout(layout)

        # Center the window on the screen
        self.center()

        self.show()

    def center(self):
        # Center the window on the screen
        qr = self.frameGeometry()
        cp = QApplication.desktop().availableGeometry().center()
        qr.moveCenter(cp)
        self.move(qr.topLeft())

    def submit_form(self):
        # Logic to handle form submission
        print("Form Submitted")
        # Here you can add code to process the data

    def clear_form(self):
        # Clear all text fields
        self.first_name.clear()
        self.last_name.clear()
        self.username.clear()
        self.password.clear()
        self.email.clear()
        self.contact_number.clear()

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

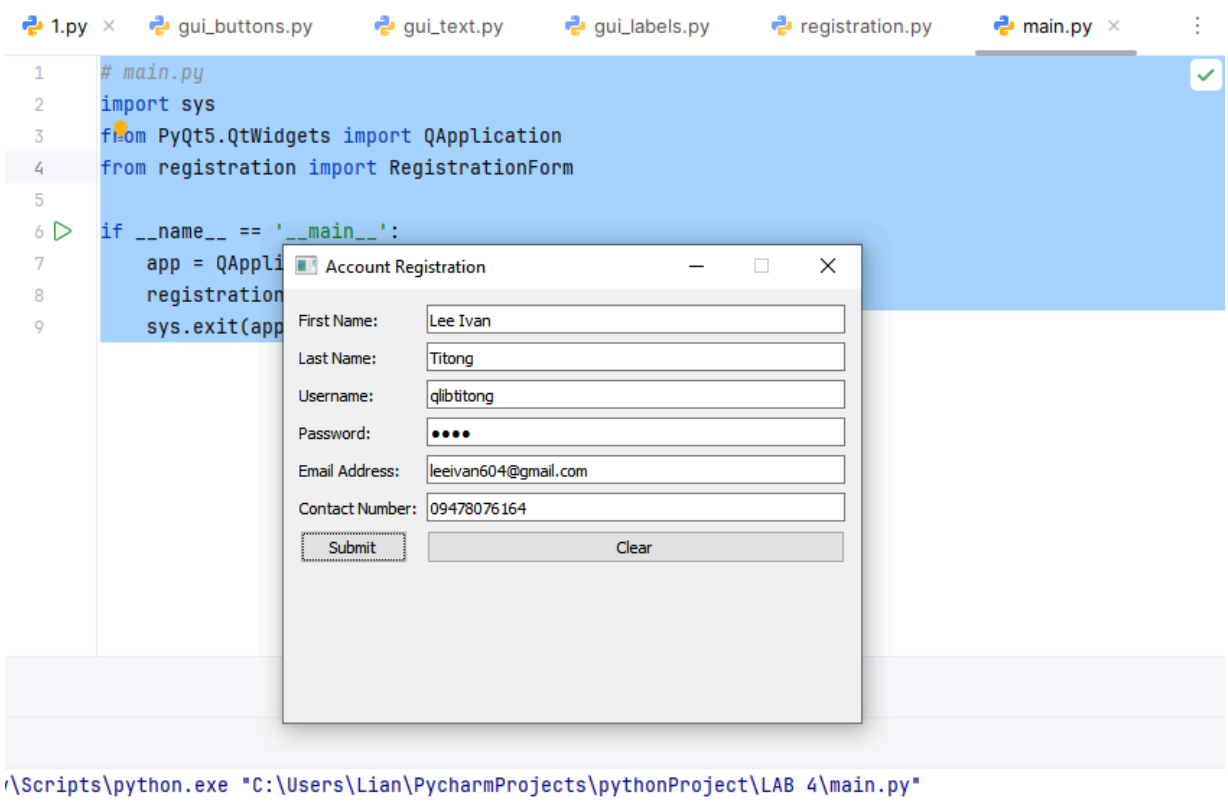
Python

```python
#main.py

# main.py
import sys
from PyQt5.QtWidgets import QApplication
from registration import RegistrationForm

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

#OUTPUT



Questions

1. Common GUI Applications:

- **Word processors:** Used for creating and editing text documents (e.g., Microsoft Word, Google Docs).
- **Web browsers:** Used for accessing and viewing web pages (e.g., Google Chrome, Firefox).
- **Spreadsheets:** Used for organizing and analyzing data (e.g., Microsoft Excel, Google Sheets).

2. Why People Use Them:

- **Easy to use:** They have a friendly interface.
- **Efficient:** They help you do things faster.
- **Versatile:** They can be used for many different tasks.

3. PyCharm's Role:

- PyCharm is a tool that helps programmers make GUI applications.
- It makes it easier to design and code GUIs.
- Without tools like PyCharm, making GUIs would be much harder.

4. Platforms for GUIs:

- **Desktop:** For computers (e.g., Windows, macOS).
- **Web:** For websites (e.g., online apps).
- **Mobile:** For smartphones and tablets.

5. Purpose of Code:

- `app = QApplication(sys.argv)`: Starts the application.
- `ex = App()`: Creates the main window.
- `sys.exit(app.exec_())`: Keeps the application running until you close it.

Conclusion

In conclusion, we successfully developed an Object-Oriented GUI Application for a simple Account Registration System using PyQt5, structured into two main files: main.py and registration.py. The application features a well-organized user interface utilizing a QFormLayout, ensuring that labels and corresponding text fields are properly aligned. The window is centered on the screen for an improved user experience, and it includes input fields for essential registration details such as first name, last name, username, password, email address, and contact number, each accompanied by a label. Functionality is enhanced with a "Submit" button for handling form submissions and a "Clear" button to reset the input fields, while the password field is designed to hide input for security. This implementation meets all specified requirements and serves as a solid foundation for potential enhancements, such as data validation or backend integration.