

Activity Name #6 - Laboratory Activity 5 - Introduction to Event Handling in GUI Development

Fernandez, Don Eleazar T.

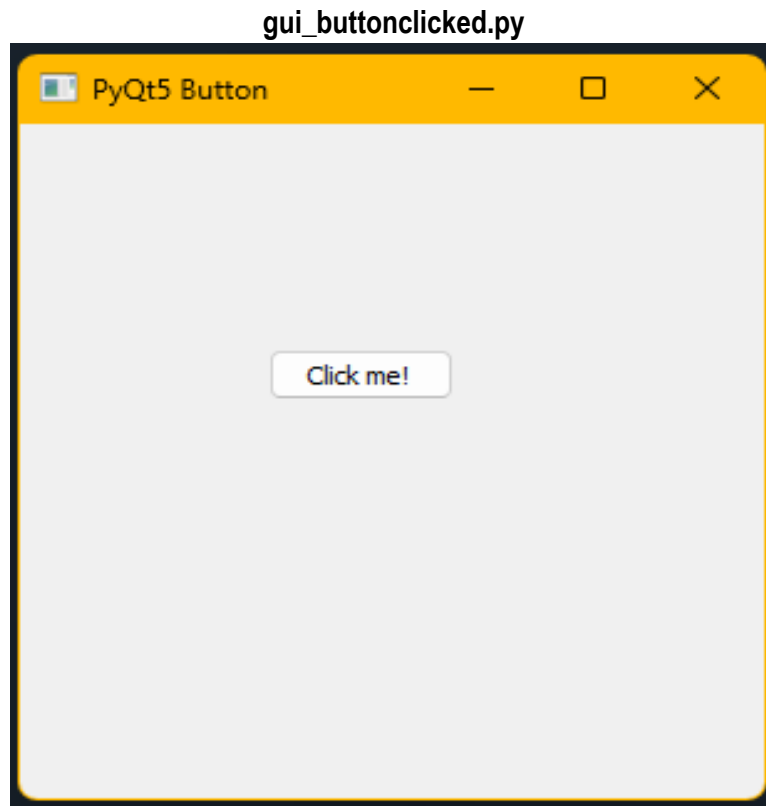
10/21/2024

CPE009/CPE21S4

Maria Rizette Sayo

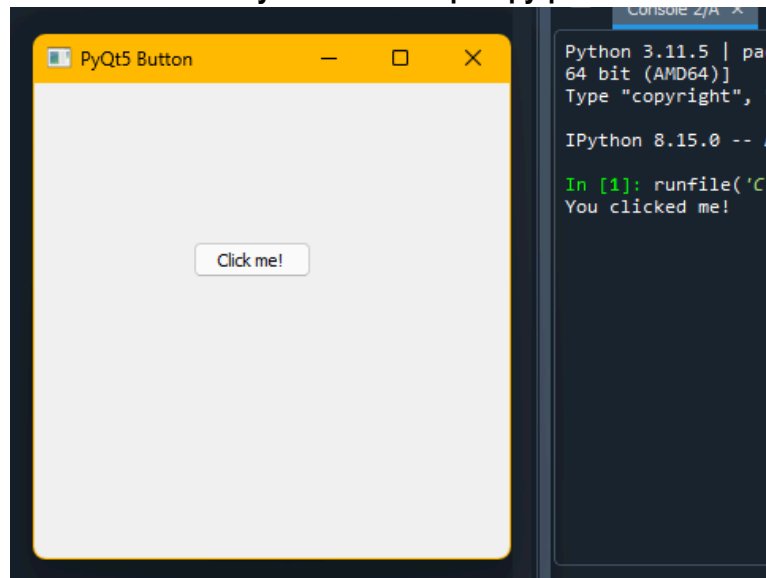
Procedure:

Event Handling:



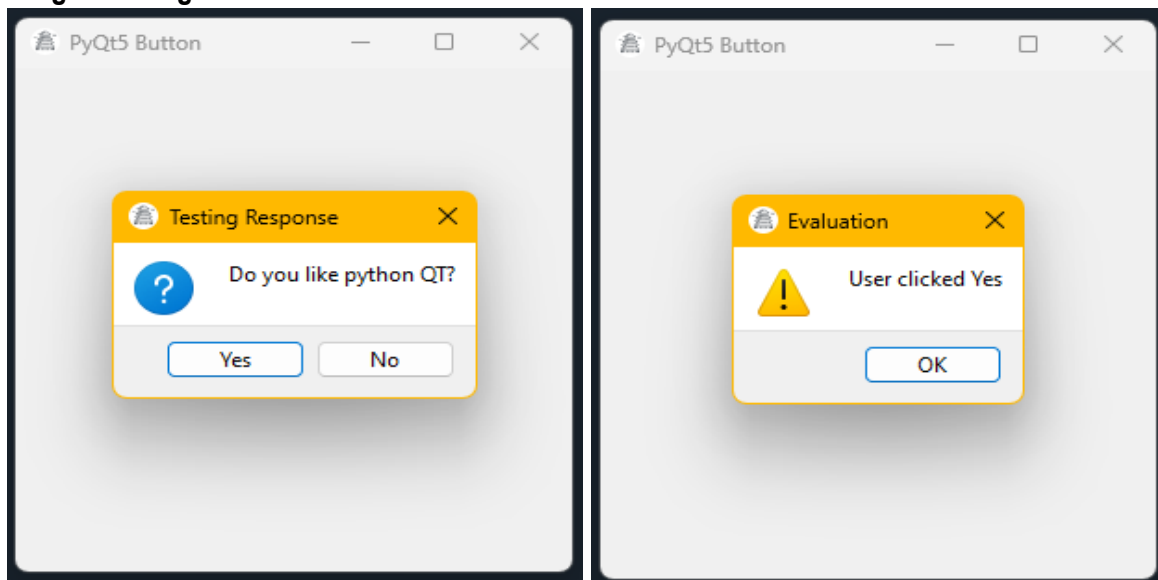
Observation: The program results in "You've hovered over me!" as the program button is hovered over by a mouse cursor. The program was successful.

`from PyQt5.QtCore import pyqtSlot`



Observation: The program results in “You’ve clicked me!” in the console as the program button was clicked. The program was successful.

Adding a Message Box:



Observation: After the program button “Click me!” was clicked, a message box popped up with a yes or no button. After the yes button was clicked the program had a successful run as intended.

Supplementary Activity:

```

import sys
import csv
import os
from PyQt5.QtWidgets import QWidget, QApplication, QLabel, QLineEdit, QPushButton, QMessageBox

class RegistrationWindow(QWidget):
    def __init__(self):
        super().__init__()
        self.init_ui()
        self.show()

    def init_ui(self):
        self.setWindowTitle('Account Registration')
        self.setGeometry(100, 100, 400, 400)
        self.fields = ['First Name', 'Last Name', 'Username', 'Password', 'Email Address', 'Contact Number']
        self.textbox = []

        for i, field in enumerate(self.fields):
            QLabel(field, self).move(50, 60 + i * 40)
            textbox = QLineEdit(self)
            textbox.move(200, 60 + i * 40)
            self.textbox.append(textbox)

        self.create_buttons()
    def create_buttons(self):
        QPushButton('Submit', self, clicked=self.register).move(100, 300)
        QPushButton('Clear', self, clicked=self.clear).move(200, 300)

    def clear(self):
        for textbox in self.textbox:
            textbox.clear()

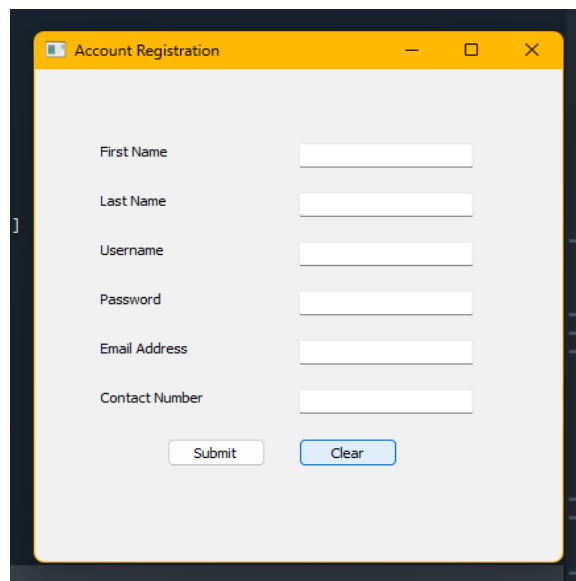
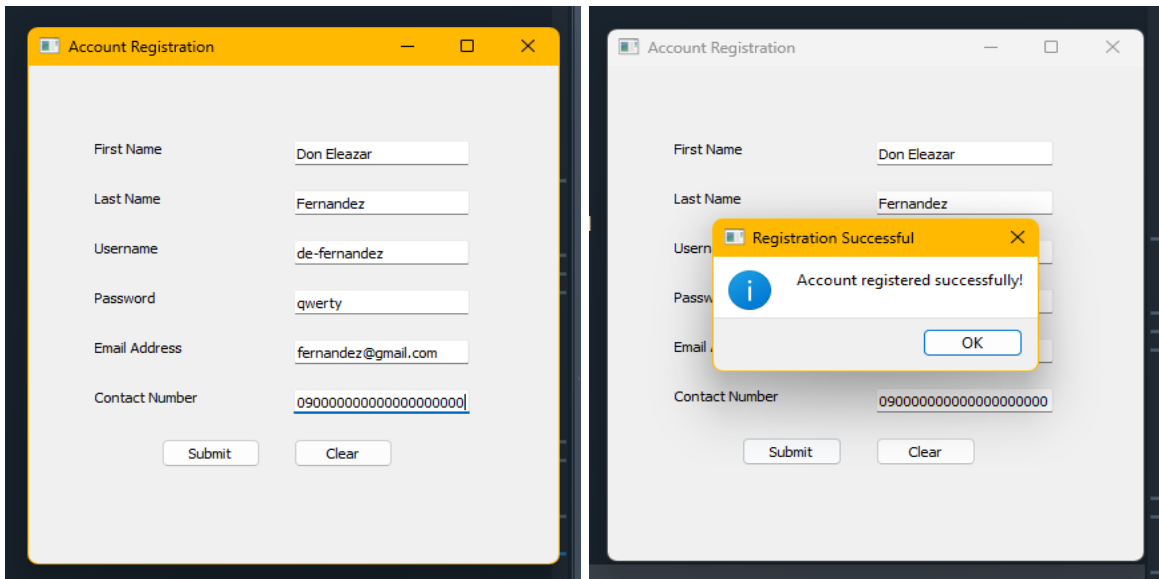
    def register(self):
        data = [field.text() for field in self.textbox]
        if any(not value for value in data):
            QMessageBox.warning(self, "Missing Information", "Please fill out all fields.", QMessageBox.Ok)
            return

        file_exists = os.path.isfile('registrations.csv')
        with open('registrations.csv', mode='a', newline='') as file:
            writer = csv.writer(file)
            if not file_exists:
                writer.writerow(self.fields)
            writer.writerow(data)

```

```
QMessageBox.information(self, "Registration Successful", "Account registered successfully!",
QMessageBox.Ok)
self.clear_fields())
```

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



Questions:

1. What are the other signals available in PyQt5? (give at least 3 and describe each)
 - In PyQt5, the signals available are: the “clicked” signal is sent when a button is clicked. The “textChanged” signal is sent when text in a text field changes. The “stateChanged” signal is sent when a checkbox is checked or unchecked.
2. Why do you think that event handling in Python is divided into signals and slots?
 - In PyQt5, event handling is divided into signals and slots as this allow different parts of the program to work together.
3. How can message boxes be used to provide a better User Experience or how can message boxes be used to make a GUI Application more user-friendly?
 - The message boxes can make a GUI Application more user friendly as there can be an interaction between the programmer and the user through the program.
4. What is Error-handling and how was it applied in the task performed?
 - The error handling is how a program manages problems during an execution. In the activity performed earlier, it shows an error message that informed me as a user that something was wrong, especially on the type error.
5. What maybe the reasons behind the need to implement error handling?
 - It is important because it helps users understand problems, prevents crashes, protects data, and much more.

Conclusion:

To conclude, the signals and slots in PyQt5 made my program more organized and linked user actions to the program. Also, the message boxes improve the user experience by giving clear feedback and helpful information. These made my program more interactive and easier to use. In the supplementary activity, I had errors to which it made me confused of how I would fix the entire code or start all over again, fortunately, I went well in the end.

I affirm that I will not give or receive any unauthorized help on this activity/exam and that all work will be my own.