Activity Name 5 - Introduction to Event Handling in GUI Development	
San Juan, Edson Ray E.	10/21/2024
CPE21S4	Prof. Maria Rizette Sayo

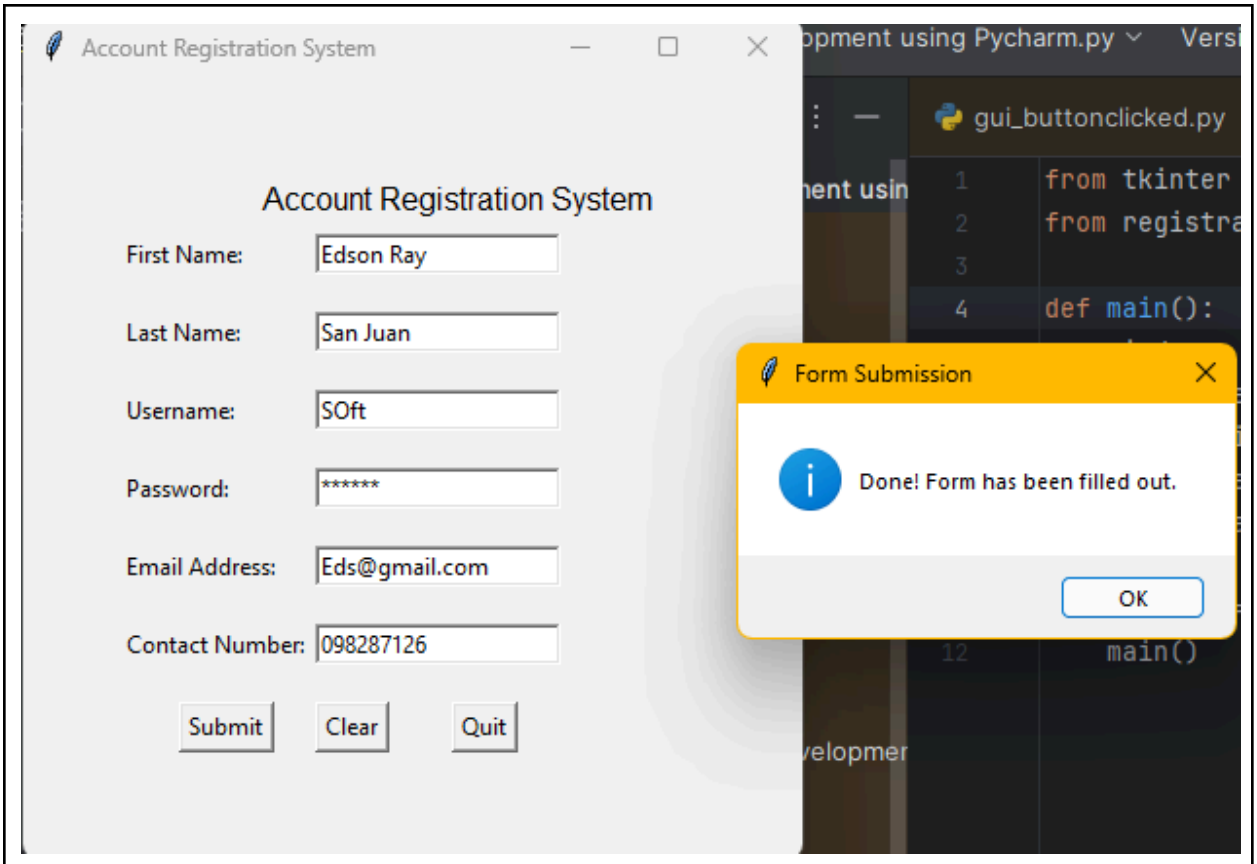
6. Supplementary Activity:

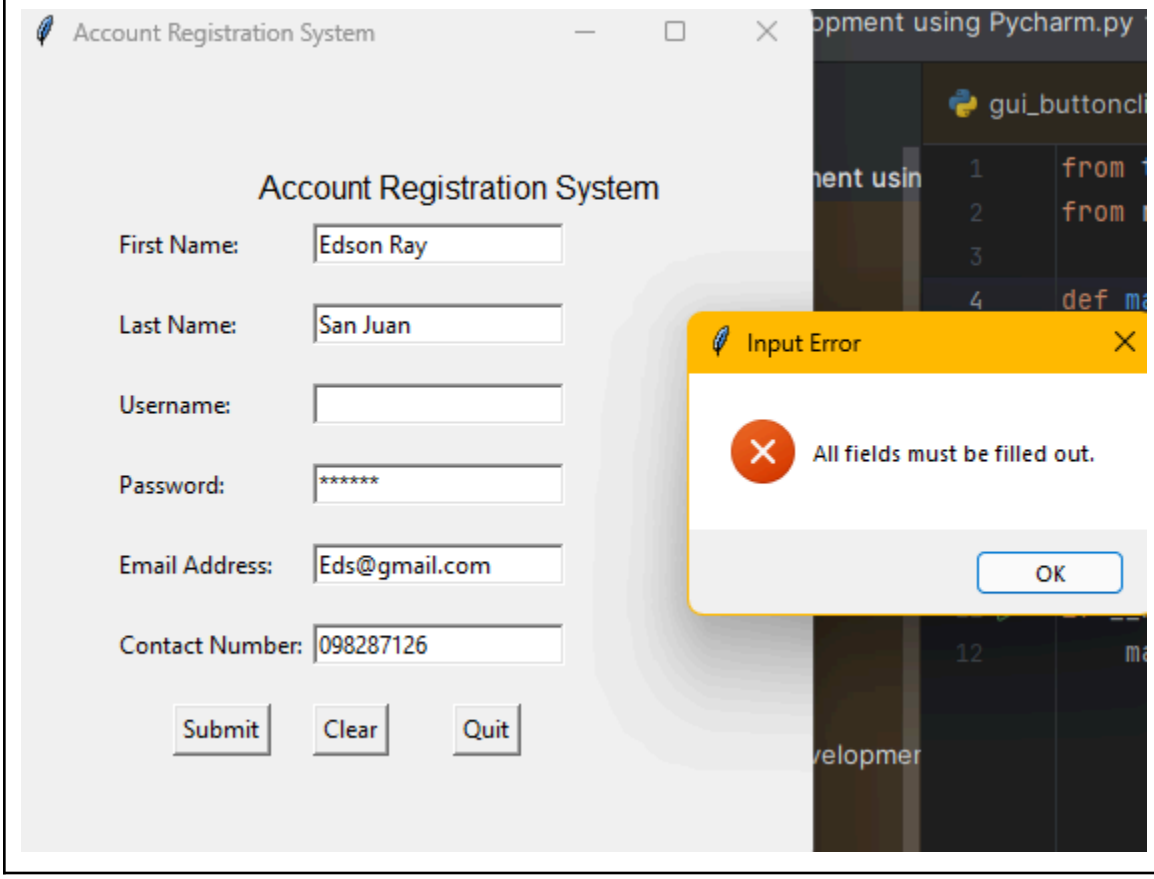
Task

Using the simple Account Registration system created in the previous laboratory activity. Create the functionality that will register the account by saving it to a Database using sqlitedict or a .csv or .txt file.

```
# Check for empty inputs
if not first_name or not last_name or not username or not password or not email_address or not contact_number:
    messagebox.showerror( title: "Input Error",  message: "All fields must be filled out.")
else:
    # Write to CSV
    with open('AccReg.csv', mode='a', newline='') as file:
        csv_writer = csv.writer(file)
        csv_writer.writerow([f'First Name: {first_name}'])
        csv_writer.writerow([f'Last Name: {last_name}', ])
        csv_writer.writerow([f'Username: {username}'])
        csv_writer.writerow([f'Password: {password}'])
        csv_writer.writerow([f'Email Address: {email_address}'])
        csv_writer.writerow([f'Contact Number: {contact_number}'])
        # new line
        csv_writer.writerow([])
```

The GUI program should not allow the registration to proceed if there is a field that has an empty value and notify of the missing values using a message box. Once the registration is successful a message should appear that would inform the user that the registration was successful. Use the appropriate symbol in making the message box.


Account Registration System
Account Registration System
First Name: Edson Ray
Last Name: San Juan
Username: SOft
Password: ******
Email Address: Eds@gmail.com
Contact Number: 098287126
Submit
Clear
Quit
Form Submission
Done! Form has been filled out.
OK
gui_buttonclicked.py
from tkinter
from registra
def main():
main()


Account Registration System
Account Registration System
First Name: Edson Ray
Last Name: San Juan
Username:
Password: ******
Email Address: Eds@gmail.com
Contact Number: 098287126
Submit
Clear
Quit
Input Error
All fields must be filled out.
OK

```
# Check for empty inputs
if not first_name or not last_name or not username or not password or not email_address or not contact_number:
    messagebox.showerror( title: "Input Error",  message: "All fields must be filled out.")
else:
    # Write to CSV
    with open('AccReg.csv', mode='a', newline='') as file:
```

```
# Confirmation dialog
messagebox.showinfo( title: "Form Submission",  message: "Done! Form has been filled out.")
```

Questions

1. What are the other signals available in PyQt5? (give at least 3 and describe each)
    a. **clicked()**: This signal is sent out when you click a button in your app. For example, if you have a "Submit" button, clicking it can make something happen, like saving information or displaying a message.
    b. **textChanged(str)**: This signal is triggered whenever the text in a text box (like a QLineEdit) changes. It helps your program keep track of what the user is typing, so you can do things like check if their input is valid right away.
    c. **valueChanged(int)**: This signal occurs when you change the value on a slider or spin box. It tells your app what the new number is, which you can use to update something else in the interface, like adjusting a display or recalculating a value based on the slider position.

2. Why do you think that event handling in Python is divided into signals and slots?
    a. Firstly, it allows for loose coupling, enabling different parts of the program to communicate without being tightly linked, which makes maintenance easier. Second, a single slot can respond to multiple signals, promoting code reuse and reducing duplication. Third, this separation improves clarity by making it easy to see which events trigger specific actions.

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?
    a. Message boxes enhance user experience in a GUI application in several key ways. They clearly communicate important information, such as warnings or confirmations, helping users understand what's happening.

4. What is Error-handling and how was it applied in the task performed?
    a. In the given code, it checks for empty required fields (first name, last name, username, password, email, contact number). If any field is empty, it shows an error message with messagebox.showerror(). If all fields are filled, it saves the data to a CSV file and displays a confirmation message with messagebox.showinfo().

5. What maybe the reasons behind the need to implement error handling?
    a. Error handling helps maintain data integrity by ensuring only valid data is processed. It also makes debugging easier by allowing developers to quickly identify issues. Additionally, it improves security by safeguarding sensitive information. Error handling lets developers manage how the program reacts to problems and strengthens the application's ability to handle unexpected situations effectively.

7. Conclusion:

In conclusion, building the account registration system with PyQt5 really helped me understand how signals and slots make a GUI interactive. I added error handling and message boxes to improve the user experience and make sure people entered valid info. This project taught me how to deal with user mistakes and showed me why writing clear and organized code is important.