**Handling User Input and Form Validation in PySide6-PyQt6**

Learn how to **capture user input** in Qt forms and implement **real-time validation** using PySide6. This exercise includes:

* **Basic input fields (QLineEdit, QComboBox, QSpinBox)**
* **Real-time validation (text length, email format, number range)**
* **Form submission with validation checks**

**1. Project Structure**

form\_validation/

│── main.py # Main application with form validation

│── requirements.txt # Dependencies (PySide6)

**2. Installing Dependencies**

Before running the project, install PySide6:

pip install PySide6

**3. Implementing a Qt Form with Validation (main.py)**

This script creates a **user registration form** with **real-time validation**.

import sys

import re

from PySide6.QtWidgets import (

QApplication, QWidget, QLabel, QLineEdit, QVBoxLayout,

QPushButton, QMessageBox, QComboBox, QSpinBox

)

from PySide6.QtGui import QIntValidator, QRegularExpressionValidator

from PySide6.QtCore import QRegularExpression

class UserForm(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.setWindowTitle("User Registration Form")

self.resize(400, 250)

layout = QVBoxLayout()

# Name Input

self.name\_label = QLabel("Full Name:")

self.name\_input = QLineEdit()

self.name\_input.setPlaceholderText("Enter your name")

self.name\_input.textChanged.connect(self.validate\_name)

layout.addWidget(self.name\_label)

layout.addWidget(self.name\_input)

# Email Input

self.email\_label = QLabel("Email Address:")

self.email\_input = QLineEdit()

self.email\_input.setPlaceholderText("Enter your email")

email\_regex = QRegularExpression(r"^[\w\.-]+@[\w\.-]+\.\w+$")

self.email\_input.setValidator(QRegularExpressionValidator(email\_regex))

self.email\_input.textChanged.connect(self.validate\_email)

layout.addWidget(self.email\_label)

layout.addWidget(self.email\_input)

# Age Input

self.age\_label = QLabel("Age:")

self.age\_input = QSpinBox()

self.age\_input.setRange(18, 100) # Accept only ages between 18 and 100

layout.addWidget(self.age\_label)

layout.addWidget(self.age\_input)

# Country Selection

self.country\_label = QLabel("Country:")

self.country\_input = QComboBox()

self.country\_input.addItems(["Select", "USA", "Canada", "UK", "India", "Australia"])

layout.addWidget(self.country\_label)

layout.addWidget(self.country\_input)

# Submit Button

self.submit\_button = QPushButton("Submit")

self.submit\_button.clicked.connect(self.submit\_form)

layout.addWidget(self.submit\_button)

self.setLayout(layout)

def validate\_name(self):

"""Ensure the name field is not empty."""

if len(self.name\_input.text()) < 3:

self.name\_input.setStyleSheet("border: 1px solid red;")

else:

self.name\_input.setStyleSheet("")

def validate\_email(self):

"""Ensure the email format is valid."""

email\_text = self.email\_input.text()

if re.match(r"^[\w\.-]+@[\w\.-]+\.\w+$", email\_text):

self.email\_input.setStyleSheet("")

else:

self.email\_input.setStyleSheet("border: 1px solid red;")

def submit\_form(self):

"""Validate the form before submission."""

if len(self.name\_input.text()) < 3:

QMessageBox.warning(self, "Validation Error", "Name must be at least 3 characters long.")

return

if not re.match(r"^[\w\.-]+@[\w\.-]+\.\w+$", self.email\_input.text()):

QMessageBox.warning(self, "Validation Error", "Enter a valid email address.")

return

if self.country\_input.currentText() == "Select":

QMessageBox.warning(self, "Validation Error", "Please select a country.")

return

QMessageBox.information(self, "Success", "Form submitted successfully!")

if \_\_name\_\_ == "\_\_main\_\_":

app = QApplication(sys.argv)

window = UserForm()

window.show()

sys.exit(app.exec())

**4. Features & Functionalities**

✅ **Real-time validation** for name and email fields.  
✅ **Regex-based validation** for email format.  
✅ **Number validation** for age input (18-100).  
✅ **Dropdown validation** for country selection.  
✅ **Form submission check** with message pop-ups.

**5. Next Steps & Improvements**

* ✅ Add **phone number validation** with regex.
* ✅ Use **QValidator** for custom validation.
* ✅ Add **password input with strength validation**.