**Lab Exercise 17- Complete Registration form using PySide6**

Here's an example of a registration form that includes text boxes, radio buttons, and a dropdown list, created using PySide:

import sys

from PySide6.QtWidgets import QApplication, QWidget, QLabel, QLineEdit, QPushButton, QVBoxLayout, QHBoxLayout, QFormLayout, QComboBox, QRadioButton, QButtonGroup, QMessageBox

class RegistrationForm(QWidget):

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

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

self.setWindowTitle('Registration Form')

self.layout = QVBoxLayout()

self.form\_layout = QFormLayout()

self.line\_edit\_name = QLineEdit()

self.line\_edit\_email = QLineEdit()

self.combo\_box\_gender = QComboBox()

self.combo\_box\_gender.addItems(["Male", "Female", "Other"])

self.radio\_group = QButtonGroup()

self.radio\_button\_student = QRadioButton("Student")

self.radio\_button\_professional = QRadioButton("Professional")

self.radio\_group.addButton(self.radio\_button\_student)

self.radio\_group.addButton(self.radio\_button\_professional)

self.form\_layout.addRow("Name:", self.line\_edit\_name)

self.form\_layout.addRow("Email:", self.line\_edit\_email)

self.form\_layout.addRow("Gender:", self.combo\_box\_gender)

radio\_layout = QHBoxLayout()

radio\_layout.addWidget(self.radio\_button\_student)

radio\_layout.addWidget(self.radio\_button\_professional)

self.form\_layout.addRow("Occupation:", radio\_layout)

self.button\_register = QPushButton('Register')

self.button\_register.clicked.connect(self.on\_register)

self.layout.addLayout(self.form\_layout)

self.layout.addWidget(self.button\_register)

self.setLayout(self.layout)

def on\_register(self):

name = self.line\_edit\_name.text()

email = self.line\_edit\_email.text()

gender = self.combo\_box\_gender.currentText()

if self.radio\_button\_student.isChecked():

occupation = "Student"

elif self.radio\_button\_professional.isChecked():

occupation = "Professional"

else:

occupation = ""

if name and email and occupation:

QMessageBox.information(self, "Registration Successful", f"Registration successful!\nName: {name}\nEmail: {email}\nGender: {gender}\nOccupation: {occupation}")

else:

QMessageBox.warning(self, "Registration Error", "Please fill in all the required fields.")

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

form = RegistrationForm()

form.show()

sys.exit(app.exec\_())

In this example, the registration form includes fields for name, email, gender (selectable from a dropdown list), and occupation (selectable using radio buttons). The user can fill in these fields and click the "Register" button. If all the required fields are filled, a message box will appear indicating successful registration; otherwise, an error message will be displayed.

To run this script, save it to a file named registration\_form.py, and run it using the following command:

python registration\_form.py

This will open the registration form window where you can enter the registration details. You can further enhance this form by adding more fields and implementing additional validation checks as needed.