

WEEK 13 : Calculator using PyQt6

Task : Implement calculator using pyqt, make individual logical commits using git, and push to Git Hub with proper documentation.

Code :

Main Window :

```
class Calculator(QMainWindow):
```

```
    def __init__(self):
```

```
        super().__init__()
```

```
        self.display = QLineEdit()
```

```
        self.buttons = {}
```

```
        for number in range(10):
```

```
            self.buttons[number] = QPushButton(str(number))
```

```
        for operator in '+-*/=':
```

```
            self.buttons[operator] = QPushButton(operator)
```

```
        self.layout = QGridLayout()
```

```
        self.layout.addWidget(self.display, 0, 0, 1, 4)
```

```
        for row in range(4):
```

```
            for column in range(4):
```

```
                self.layout.addWidget(self.buttons[row * 4 + column], row + 1, column)
```

```
        self.setLayout(self.layout)
```

```
        self.setWindowTitle('Calculator')
```

```
for button in self.buttons.values():  
    button.clicked.connect(self.on_button_clicked)
```

```
def on_button_clicked(self, button):
```

```
    text = button.text()
```

```
    if text == '=':
```

```
        expression = self.display.text()
```

```
        result = eval(expression)
```

```
        self.display.setText(str(result))
```

```
    else:
```

```
        self.display.setText(self.display.text() + text)
```

Application :

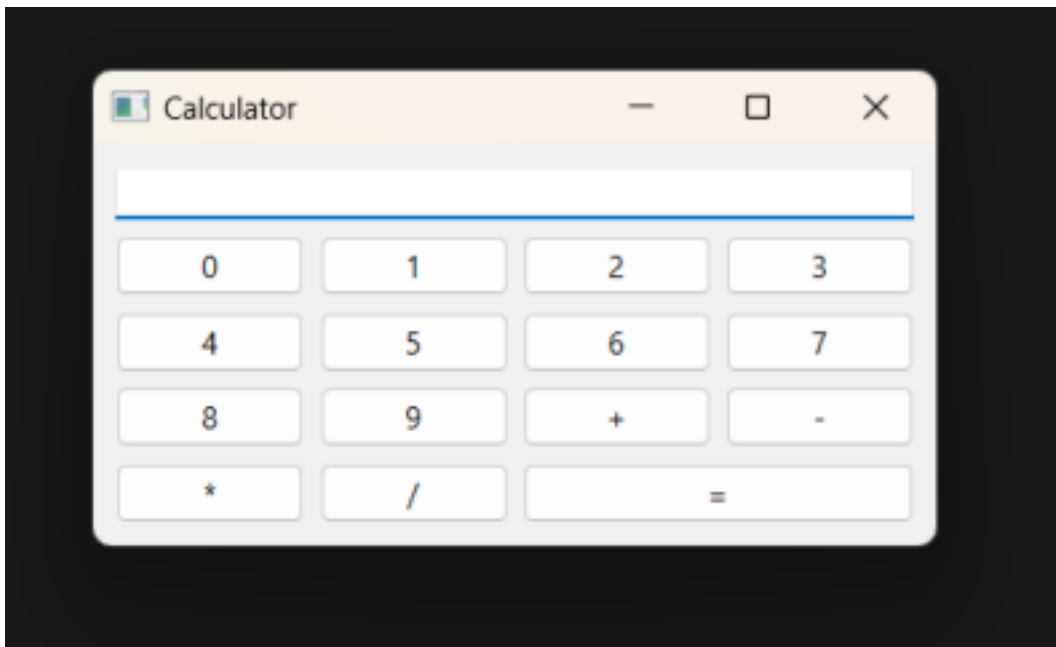
```
app = QApplication(sys.argv)
```

```
calculator = Calculator()
```

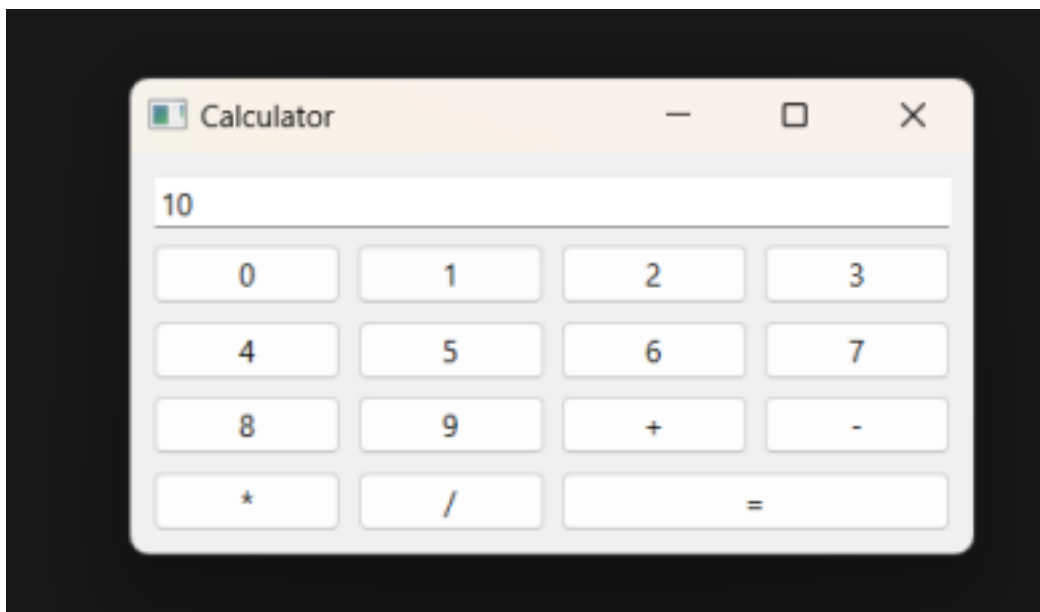
```
calculator.show()
```

```
app.exec()
```

Output :



Performing 5+5 :



Conclusion : Through this practical i learned how to work with PyQt and learned the concept of building a GUI application using it

