**Lab Exercise 9 – Two Number Calculator in QML and PyQT**

**Lab Exercise: Creating a Two-Number Calculator with PyQt and QML**

Creating a lab exercise for building a two-number calculator using PyQt and QML is a great way to introduce learners to creating interactive applications with graphical user interfaces. In this exercise, learners will create a simple calculator that can perform basic arithmetic operations (addition, subtraction, multiplication, division) on two numbers entered by the user. Here's a step-by-step lab exercise:

Objective: Create a PyQt application with a QML-based user interface that allows users to perform arithmetic operations on two numbers.

**Requirements:**

* PyQt5 should be installed.
* Basic understanding of Python and PyQt5.

**Instructions:**

* In this lab exercise, learners will create a two-number calculator application using PyQt and QML. Follow the steps below:
* Create the QML File:
* Create a QML file named calculator.qml with the following content:

import QtQuick 2.15

import QtQuick.Controls 2.15

ApplicationWindow {

visible: true

width: 320

height: 480

title: "Two-Number Calculator"

Rectangle {

width: parent.width

height: parent.height

Column {

spacing: 10

TextField {

id: num1Input

width: parent.width

placeholderText: "Enter first number"

}

TextField {

id: num2Input

width: parent.width

placeholderText: "Enter second number"

}

Row {

spacing: 10

Button {

text: "Add"

onClicked: resultText.text = (parseFloat(num1Input.text) + parseFloat(num2Input.text)).toString()

}

Button {

text: "Subtract"

onClicked: resultText.text = (parseFloat(num1Input.text) - parseFloat(num2Input.text)).toString()

}

Button {

text: "Multiply"

onClicked: resultText.text = (parseFloat(num1Input.text) \* parseFloat(num2Input.text)).toString()

}

Button {

text: "Divide"

onClicked: {

if (parseFloat(num2Input.text) === 0) {

resultText.text = "Error: Division by zero"

} else {

resultText.text = (parseFloat(num1Input.text) / parseFloat(num2Input.text)).toString()

}

}

}

}

Label {

id: resultText

width: parent.width

text: ""

}

}

}

}

**Create the Python Script:**

Create a Python script, e.g., calculator\_app.py, and add the following code:

import sys

from PyQt5.QtCore import QUrl

from PyQt5.QtWidgets import QApplication

from PyQt5.QtQml import QQmlApplicationEngine

app = QApplication(sys.argv)

engine = QQmlApplicationEngine()

engine.load(QUrl.fromLocalFile("calculator.qml"))

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

**Run the Application:**

Open a terminal or command prompt, navigate to the directory containing the Python script and QML file, and run the Python script:

python calculator\_app.py

The PyQt5 application should open and display a simple two-number calculator with a QML-based user interface.