**Lab Exercise 14- Pie Chart using PyQt**

**Lab Exercise: Creating a Pie Chart in PyQt**

Creating a lab exercise for a pie chart in PyQt involves guiding students through building a PyQt application that displays a pie chart using the Matplotlib library. Below is a lab exercise that demonstrates how to create such an application:

**Objective: Create a PyQt application that displays a pie chart using Matplotlib.**

**Requirements:**

* PyQt5: You should have PyQt5 installed.
* Matplotlib: Install Matplotlib using pip install matplotlib.

**Instructions:**

Import the required modules:

import sys

from PyQt5.QtWidgets import QApplication, QMainWindow, QVBoxLayout, QWidget

from matplotlib.backends.backend\_qt5agg import FigureCanvasQTAgg as FigureCanvas

from matplotlib.figure import Figure

**Create a class for the PyQt application:**

class PieChartApp(QMainWindow):

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

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

self.initUI()

def initUI(self):

self.setGeometry(100, 100, 800, 600)

self.setWindowTitle('Pie Chart Example')

central\_widget = QWidget(self)

self.setCentralWidget(central\_widget)

layout = QVBoxLayout()

central\_widget.setLayout(layout)

# Create a FigureCanvas to embed the Matplotlib plot

self.canvas = FigureCanvas(Figure())

layout.addWidget(self.canvas)

self.draw\_pie\_chart() # Create and display the pie chart

def draw\_pie\_chart(self):

data = [25, 40, 15, 20] # Sample data for the pie chart

labels = ['Category A', 'Category B', 'Category C', 'Category D']

ax = self.canvas.figure.add\_subplot(111)

ax.pie(data, labels=labels, autopct='%1.1f%%', startangle=90)

ax.axis('equal')

self.canvas.draw()

**Create a function to run the application:**

def run\_app():

app = QApplication(sys.argv)

ex = PieChartApp()

ex.show()

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

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

run\_app()

Save this code in a Python file, for example, pie\_chart\_app.py.

Instruct the students to run the program, which will create a PyQt application displaying a simple pie chart. Students can modify the data list and labels list to experiment with different data for the pie chart.

This exercise demonstrates how to integrate Matplotlib with PyQt to display a pie chart, and students can use this as a starting point to explore more advanced charting and graphing capabilities in their PyQt applications.