**Lab Exercise 13- Bar Chart using PyQt**

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

Creating a lab exercise on graphs and charts in PyQt involves demonstrating how to display various types of graphs and charts using libraries like Matplotlib and Pyqtgraph. Below is a lab exercise that guides students through creating a PyQt application that displays a simple bar chart using Matplotlib.

Objective: Create a PyQt application that displays a bar 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

import matplotlib.pyplot as plt

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

class BarChartApp(QMainWindow):

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

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

self.initUI()

def initUI(self):

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

self.setWindowTitle('Bar 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(plt.figure())

layout.addWidget(self.canvas)

self.draw\_bar\_chart() # Create and display the bar chart

def draw\_bar\_chart(self):

data = {'Apples': 5, 'Bananas': 2, 'Cherries': 7, 'Grapes': 4}

labels = data.keys()

values = data.values()

# Create the bar chart

plt.bar(labels, values)

plt.xlabel('Fruits')

plt.ylabel('Quantity')

plt.title('Fruit Distribution')

# Display the bar chart on the canvas

self.canvas.draw()

Create a function to run the application:

def run\_app():

app = QApplication(sys.argv)

ex = BarChartApp()

ex.show()

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

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

run\_app()

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

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

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