**Extra Lab 5– Using Qt Charts in PySide6**

**Objective**

Learn how to use **Qt Charts** in PySide6 to create and display **line charts, bar charts, and pie charts** in a Qt application.

**Step 1: Understanding Qt Charts**

Qt Charts provides an easy way to visualize data using different types of charts, such as:

* **QLineSeries** → Line Charts
* **QBarSeries** → Bar Charts
* **QPieSeries** → Pie Charts
* **QScatterSeries** → Scatter Plots

**Step 2: Implementing a Simple Line Chart**

This example creates a **line chart** with dummy data.

import sys

from PySide6.QtWidgets import QApplication, QMainWindow, QVBoxLayout, QWidget

from PySide6.QtCharts import QChart, QChartView, QLineSeries

from PySide6.QtCore import QPointF

class LineChartApp(QMainWindow):

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

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

self.setWindowTitle("Qt Charts - Line Chart Example")

self.setGeometry(100, 100, 600, 400)

# Create a line series and add data points

series = QLineSeries()

data\_points = [(0, 5), (1, 15), (2, 25), (3, 20), (4, 30)]

for x, y in data\_points:

series.append(QPointF(x, y))

# Create a chart and add the series

chart = QChart()

chart.addSeries(series)

chart.setTitle("Simple Line Chart")

chart.createDefaultAxes() # Auto-create X and Y axes

# Create a ChartView to display the chart

chart\_view = QChartView(chart)

chart\_view.setRenderHint(chart\_view.renderHints()) # Enable smooth rendering

# Set up the layout

central\_widget = QWidget()

layout = QVBoxLayout()

layout.addWidget(chart\_view)

central\_widget.setLayout(layout)

self.setCentralWidget(central\_widget)

if \_\_name\_\_ == "\_\_main\_\_":

app = QApplication(sys.argv)

window = LineChartApp()

window.show()

sys.exit(app.exec())

**Step 3: Explanation**

1. **Creating a QLineSeries**
   * series = QLineSeries() initializes the series.
   * series.append(QPointF(x, y)) adds data points to the chart.
2. **Setting Up the Chart (QChart)**
   * chart = QChart() creates a chart.
   * chart.addSeries(series) adds the line series to the chart.
   * chart.createDefaultAxes() automatically sets X and Y axes.
3. **Displaying the Chart (QChartView)**
   * QChartView(chart) creates a widget to display the chart.
   * setRenderHint() ensures smooth rendering.

**Step 4: Implementing a Bar Chart**

A **bar chart** is useful for comparing discrete values.

from PySide6.QtCharts import QBarSet, QBarSeries, QChart, QChartView, QBarCategoryAxis

class BarChartApp(QMainWindow):

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

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

self.setWindowTitle("Qt Charts - Bar Chart Example")

self.setGeometry(100, 100, 600, 400)

# Create a bar set

set0 = QBarSet("Category A")

set0.append([10, 20, 30, 40, 50]) # Add values

# Create a bar series and add the set

series = QBarSeries()

series.append(set0)

# Create the chart

chart = QChart()

chart.addSeries(series)

chart.setTitle("Simple Bar Chart")

# Set X-axis labels

categories = ["Jan", "Feb", "Mar", "Apr", "May"]

axisX = QBarCategoryAxis()

axisX.append(categories)

chart.addAxis(axisX, Qt.AlignBottom)

series.attachAxis(axisX)

# Display the chart

chart\_view = QChartView(chart)

layout = QVBoxLayout()

layout.addWidget(chart\_view)

central\_widget = QWidget()

central\_widget.setLayout(layout)

self.setCentralWidget(central\_widget)

**Step 5: Implementing a Pie Chart**

A **pie chart** is useful for visualizing percentages.

from PySide6.QtCharts import QPieSeries

class PieChartApp(QMainWindow):

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

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

self.setWindowTitle("Qt Charts - Pie Chart Example")

self.setGeometry(100, 100, 600, 400)

# Create a pie series

series = QPieSeries()

series.append("A", 40)

series.append("B", 30)

series.append("C", 20)

series.append("D", 10)

# Create the chart

chart = QChart()

chart.addSeries(series)

chart.setTitle("Simple Pie Chart")

# Display the chart

chart\_view = QChartView(chart)

layout = QVBoxLayout()

layout.addWidget(chart\_view)

central\_widget = QWidget()

central\_widget.setLayout(layout)

self.setCentralWidget(central\_widget)

**Expected Output**

* **Line Chart:** A plotted line connecting data points.
* **Bar Chart:** A simple bar graph with category labels.
* **Pie Chart:** A pie chart with four labeled sections.