**Extra Lab 2 – Displaying an Image in a Fixed-Size QLabel While Maintaining Aspect Ratio**

**Objective:**

Learn how to **display an image inside a QLabel of fixed size** while **maintaining its aspect ratio** in PySide6.

**Prerequisites**

Ensure you have **PySide6** installed:

pip install PySide6

**Steps to Complete the Lab**

**Step 1: Create the PySide6 Application**

This program allows users to **select an image**, which is then displayed inside a **fixed-size QLabel** while maintaining its **original aspect ratio**.

import sys

from PySide6.QtWidgets import QApplication, QWidget, QLabel, QPushButton, QVBoxLayout, QFileDialog

from PySide6.QtGui import QPixmap

from PySide6.QtCore import Qt

class ImageDisplayApp(QWidget):

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

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

self.setWindowTitle("Fixed Size Image Display")

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

# Create a fixed-size QLabel

self.image\_label = QLabel("Select an Image")

self.image\_label.setFixedSize(300, 300) # Fixed size label

self.image\_label.setAlignment(Qt.AlignCenter)

self.image\_label.setStyleSheet("border: 2px solid black;")

# Button to load an image

self.load\_button = QPushButton("Load Image")

self.load\_button.clicked.connect(self.load\_image)

# Layout

layout = QVBoxLayout()

layout.addWidget(self.image\_label)

layout.addWidget(self.load\_button)

self.setLayout(layout)

def load\_image(self):

"""Opens a file dialog to select an image and displays it while maintaining aspect ratio."""

file\_path, \_ = QFileDialog.getOpenFileName(self, "Select Image", "", "Images (\*.png \*.jpg \*.jpeg \*.bmp)")

if file\_path:

pixmap = QPixmap(file\_path)

scaled\_pixmap = pixmap.scaled(self.image\_label.size(), Qt.KeepAspectRatio, Qt.SmoothTransformation)

self.image\_label.setPixmap(scaled\_pixmap)

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

app = QApplication(sys.argv)

window = ImageDisplayApp()

window.show()

sys.exit(app.exec())

**Explanation**

1. **Fixed QLabel Size**
   * The label is set to a **fixed size of 300x300 pixels** using:

self.image\_label.setFixedSize(300, 300)

* + This ensures the label does **not resize** based on the image.

1. **Maintaining Aspect Ratio**
   * The image is **rescaled** while preserving its aspect ratio:

scaled\_pixmap = pixmap.scaled(self.image\_label.size(), Qt.KeepAspectRatio, Qt.SmoothTransformation)

* + Qt.KeepAspectRatio ensures the image scales proportionally.
  + Qt.SmoothTransformation improves image quality during resizing.

1. **Loading Images Dynamically**
   * Users can **select any image** using a **file dialog**.
   * The selected image is displayed **inside the fixed-size QLabel**.

**Expected Behavior**

1. Click **"Load Image"** to select an image.
2. The image **resizes** to fit inside the **300x300** QLabel **without distortion**.
3. If the image is **larger**, it **shrinks while maintaining proportions**.
4. If the image is **smaller**, it appears **centered** inside the label.