**Lab Exercise 18- Horizontal Translation Animation using PySide6**

Here's an example of a registration form that includes text boxes, radio buttons, and a dropdown list, created using PySide:

Here is a lab exercise that demonstrates a horizontal translation animation using PySide:

import sys

from PySide2.QtWidgets import QApplication, QMainWindow, QPushButton

from PySide2.QtCore import QPoint, QPropertyAnimation

class MainWindow(QMainWindow):

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

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

self.setGeometry(100, 100, 300, 200)

self.setWindowTitle('PySide Horizontal Translation Animation')

self.button = QPushButton('Animate Me', self)

self.button.setGeometry(50, 50, 100, 30)

self.button.clicked.connect(self.start\_animation)

def start\_animation(self):

start\_point = self.button.pos()

end\_point = QPoint(200, self.button.y())

self.anim = QPropertyAnimation(self.button, b"pos")

self.anim.setDuration(1000)

self.anim.setStartValue(start\_point)

self.anim.setEndValue(end\_point)

self.anim.start()

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

app = QApplication(sys.argv)

window = MainWindow()

window.show()

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

In this lab exercise, a horizontal translation animation is created using PySide. The button moves horizontally from its initial position to the target position within the window.

To test the lab exercise, save the code to a file named horizontal\_translation\_animation.py and run it using the following command:

python horizontal\_translation\_animation.py

Upon running the script, a window will appear with a button. Clicking the button will trigger the horizontal translation animation, moving the button horizontally from its initial position to the target position. You can customize the animation by modifying the coordinates and duration in the setStartValue, setEndValue, and setDuration methods according to your requirements.