**Lab Exercise 12- Read JSON using PyQt**

**Lab Exercise: Read JSON file with PyQt and QML and Present on QTable**

Creating a PyQt project to read and display JSON data involves several steps. In this example, I'll provide a simple PyQt application that loads JSON data from a file and displays it in a table widget. You'll need to have PyQt5 installed. If you haven't already installed it, you can use pip to install it:

pip install PyQt5

Create a JSON file as given below:

Test.json

{

    "name": "John Doe",

    "age": 30,

    "city": "New York",

    "email": "johndoe@example.com",

    "hobbies": ["Reading", "Hiking", "Cooking"],

    "is\_student": false

}

Now, you can create a PyQt application to read and display JSON data. Here's a basic example:

import sys

import json

from PyQt5.QtWidgets import QApplication, QMainWindow, QTableWidget, QTableWidgetItem, QVBoxLayout, QPushButton, QWidget, QFileDialog

class JSONViewer(QMainWindow):

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

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

self.initUI()

def initUI(self):

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

self.setWindowTitle('JSON Viewer')

# Create a central widget

central\_widget = QWidget(self)

self.setCentralWidget(central\_widget)

# Create a table widget to display JSON data

self.table = QTableWidget(self)

central\_layout = QVBoxLayout()

central\_layout.addWidget(self.table)

# Create a load button

self.load\_button = QPushButton('Load JSON', self)

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

central\_layout.addWidget(self.load\_button)

central\_widget.setLayout(central\_layout)

def load\_json(self):

options = QFileDialog.Options()

file\_name, \_ = QFileDialog.getOpenFileName(self, "Open JSON File", "", "JSON Files (\*.json);;All Files (\*)", options=options)

if file\_name:

with open(file\_name, 'r') as json\_file:

data = json.load(json\_file)

# Set the table widget rows and columns

self.table.setRowCount(len(data))

self.table.setColumnCount(2)

# Populate the table with JSON data

for row, (key, value) in enumerate(data.items()):

key\_item = QTableWidgetItem(str(key))

value\_item = QTableWidgetItem(str(value))

self.table.setItem(row, 0, key\_item)

self.table.setItem(row, 1, value\_item)

# Set headers for the table

self.table.setHorizontalHeaderLabels(['Key', 'Value'])

def main():

app = QApplication(sys.argv)

ex = JSONViewer()

ex.show()

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

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

main()

Save this code in a Python file (e.g., json\_viewer.py) and run it. This PyQt application provides a simple user interface with a "Load JSON" button. When you click the button, you can select a JSON file, and the application will display the JSON data in a table format.