import sys

from PyQt5.QtWidgets import (QApplication, QWidget, QLabel, QLineEdit,

QPushButton, QTreeView)

from PyQt5.QtGui import (QPixmap, QImage)

from PyQt5.QtCore import Qt

import requests

import io

from PIL import Image

class VideoDownloader(QWidget):

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

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

 self.initUI()

 def initUI(self):

 self.setWindowTitle("Video Downloader")

 self.setGeometry(300, 300, 800, 600)

 self.searchLabel = QLabel("Search for videos:", self)

 self.searchLabel.move(10, 80)

 self.searchLabel.setFont(QtGui.QFont('Arial', 15, QtGui.QFont.Bold))

 self.searchEntry = QLineEdit(self)

 self.searchEntry.move(230, 80)

 self.searchEntry.setFont(QtGui.QFont('Arial', 15, QtGui.QFont.Bold))

 self.searchEntry.setFixedWidth(400)

 self.searchButton = QPushButton("Search", self)

 self.searchButton.move(330, 130)

 self.searchButton.setFont(QtGui.QFont('Arial', 10, QtGui.QFont.Bold))

 self.searchButton.clicked.connect(self.searchVideosVk)

 self.resultTree = QTreeView(self)

 self.resultTree.move(10, 180)

 self.resultTree.setFixedWidth(600)

 self.resultTree.setFixedHeight(400)

 self.model = QStandardItemModel(self.resultTree)

 self.model.setHorizontalHeaderLabels(['Title', 'Thumbnail'])

 self.resultTree.setModel(self.model)

 def searchVideosVk(self):

 try:

 query = self.searchEntry.text()

 apiUrl = f"https://api.vk.com/method/video.search?q={query}&count=10&v=5.131"

 response = requests.get(apiUrl).json()

 results = response['response']['items']

 self.model.removeRows(0, self.model.rowCount())

 for result in results:

 thumbnail = requests.get(result['image']).content

 thumbnail = Image.open(io.BytesIO(thumbnail))

 thumbnail = thumbnail.resize((100, 100), Image.ANTIALIAS)

 thumbnail = QImage(thumbnail.tobytes(), thumbnail.width, thumbnail.height, QImage.Format\_RGBA8888)

 thumbnail = QPixmap.fromImage(thumbnail)

 item = QStandardItem(result['title'])

 self.model.appendRow([item, QStandardItem()])

 self.model.item(self.model.rowCount() - 1, 1).setData(thumbnail, Qt.DecorationRole)

 except Exception as e:

 QMessageBox.critical(self, "Error", str(e))

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

 app = QApplication(sys.argv)

 downloader = VideoDownloader()

 downloader.show()

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