import sys

from PyQt5.QtWidgets import \*

from PyQt5.QtCore import QDate

from PyQt5 import uic

log\_class = uic.loadUiType('C:/pyqt\_practice/log.ui')[0]

record\_class = uic.loadUiType('C:/pyqt\_practice/record.ui')[0]

main\_class = uic.loadUiType('C:/pyqt\_practice/main.ui')[0]

day1 = []

day2 = []

day3 = []

start\_date = QDate.currentDate()

class Log(QDialog, log\_class):

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

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

self.setupUi(self)

self.setWindowTitle("운동 로그 조회")

self.date\_edit.setDate(start\_date)

self.date\_edit.setDateRange(start\_date, start\_date.addDays(2))

global day1

if day1:

log\_count = 0

for log in day1:

self.log\_table.insertRow(log\_count)

self.log\_table.setItem(log\_count, 0, QTableWidgetItem(log.get("date").toString("yyyy-MMM-dd")))

self.log\_table.setItem(log\_count, 1, QTableWidgetItem(log.get("category")))

self.log\_table.setItem(log\_count, 2, QTableWidgetItem(str(log.get("time"))+"분"))

chk\_box = QCheckBox("인증하기")

self.log\_table.setCellWidget(log\_count, 3, chk\_box)

log\_count += 1

else

self.info\_label.setText("\*선택 날짜에 해당하는 운동 로그가 없습니다.\*")

self.date\_edit.dateChanged.connect(self.addToTable)

def addToTable(self):

global day1, day2, day3

for i in range(self.log\_table.rowCount()):

self.log\_table.removeRow(0)

if self.date\_edit.date() == start\_date:

choosed\_day = day1

elif self.date\_edit.date() == start\_date.addDays(1):

choosed\_day = day2

else:

choosed\_day = day3

if choosed\_day:

log\_count = 0

for log in choosed\_day:

self.log\_table.insertRow(log\_count)

self.log\_table.setItem(log\_count, 0, QTableWidgetItem(log.get("date").toString("yyyy-MMM-dd")))

self.log\_table.setItem(log\_count, 1, QTableWidgetItem(log.get("category")))

self.log\_table.setItem(log\_count, 2, QTableWidgetItem(str(log.get("time"))+"분"))

chk\_box = QCheckBox()

self.log\_table.setCellWidget(log\_count, 3, chk\_box)

log\_count += 1

else:

self.info\_label.setText("\*선택 날짜에 해당하는 운동 로그가 없습니다.\*")

class Record(QDialog, record\_class):

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

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

self.setupUi(self)

self.setWindowTitle("운동 계획 기록하기")

#선택할 수 있는 날짜의 범위를 시작일 부터 3일로 제한하기

self.date\_edit.setDate(start\_date)

self.date\_edit.setDateRange(start\_date, start\_date.addDays(2))

self.record\_btn.clicked.connect(self.recordLog)

def recordLog(self):

global day1, day2, day3

new\_log = {"category":self.category.currentText(), "date":self.date\_edit.date(),"time":self.time.value()}

if new\_log.get("date") == start\_date:

day1.append(new\_log)

elif new\_log.get("date") == start\_date.addDays(1):

day2.append(new\_log)

else:

day3.append(new\_log)

ans = QMessageBox.question(self, "운동 계획 기록 완료!",

"운동 계획이 추가되었습니다.\n운동 계획 로그를 조회하러 가시겠습니

까?”, QMessageBox.Yes|QMessageBox.No, QMessageBox.Yes)

if ans == QMessageBox.No:

self.close()

else:

self.close()

log\_window = Log()

log\_window.show()

log\_window.exec()

class Main(QMainWindow, main\_class):

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

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

self.setupUi(self)

self.setWindowTitle("작심삼일 운동 챌린지")

self.log\_action.triggered.connect(self.logBtnClicked)

self.record\_btn.clicked.connect(self.recordBtnClicked)

self.log\_btn.clicked.connect(self.logBtnClicked)

def recordBtnClicked(self):

record\_window = Record()

record\_window.show()

record\_window.exec()

def logBtnClicked(self):

log\_window = Log()

log\_window.show()

log\_window.exec()

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

app = QApplication(sys.argv)

main\_window = Main()

main\_window.show()

app.exec\_()