ECEA 5347- Project 1

Implementation/Assumption Notes

Objective- Create a Qt user interface that will:

1. Display temperature & humidity read in from a pseudo sensor, respond to events (e.g. button presses) and send data to code (e.g. text fields)
2. Contain a Qt button that will have your main Python program read a single humidity/temperature reading from the pseudo sensor and then display the values on the UI
3. Use a simple SQL database like MariaDB or other extendable Python data structure to store incoming humidity/temperature value pairs along with a timestamp
4. Provide a Qt button that will read 10 values from the pseudo sensor with a one second delay between each reading (these values will be stored with their timestamps) – you can decide how this is displayed on the UI – showing data items read, showing completion, etc.
5. Provide a Qt button that will cause the main Python program to calculate and display on the UI from the last 10 (or less) temperature/humidity pairs read in – include minimum, maximum, and average values for each reading type
6. Include two fields to set temperature and humidity alarm values; if the main Python program sees any temperature or humidity read from the pseudo sensor exceeds the alarm value, indicate an alarm on the UI – these alarm fields should have default values at startup
7. Displays a Qt button on the UI that will close the UI and end the Python main program
8. Data from the pseudo sensor is humidity between 0 and 100%, and temperature between -20 and 100 degrees Fahrenheit.

Additional Notes

1. IDE is set up with: VSCode, Win10, PyQt5, Python.

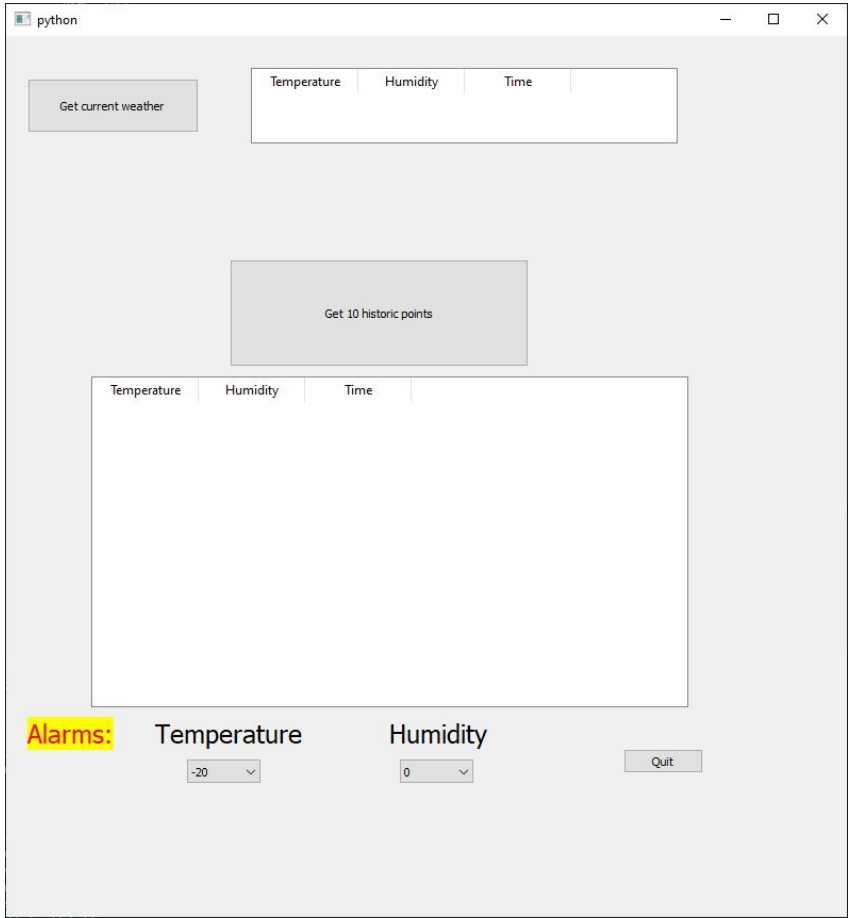
weatherApp.py

```
import sys
import random
import mysql.connector

from PyQt5.uic import loadUi
from PyQt5.QtSql import QSqlDatabase, QSqlQuery
from PyQt5.QtWidgets import QDialog, QApplication, QMainWindow, QMessageBox, QLab
el, QTableWidget, QTableWidgetItem
from datetime import datetime
from time import sleep
from PyQt5 import QtCore, QtGui, QtWidgets

class MainWindow(QDialog):
    def __init__(self):
        super(MainWindow, self).__init__()
        loadUi("buttontable.ui",self)
        self.button1.clicked.connect(self.button1_clicked)
        self.button2.clicked.connect(self.button2_clicked)


    def button1_clicked(self):
        mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="123",
        database="weather"
        )



        mycursor = mydb.cursor()
        tempin = str(random.randint(-20, 100))
        humin = str(random.randint(0, 100))
        now = datetime.now()
        timein = str(now.strftime("%H:%M:%S"))
        sql = "INSERT INTO weather (temperature, humidity, timestamp) VALUES (%s,
 %s, %s)"
        val = (tempin, humin, timein)
        mycursor.execute(sql, val)
        mydb.commit()

        msg = QMessageBox()
        msg.setWindowTitle("Temperature alert!")
        msg.setText("Temperature is high!")
        x = msg.exec()
```

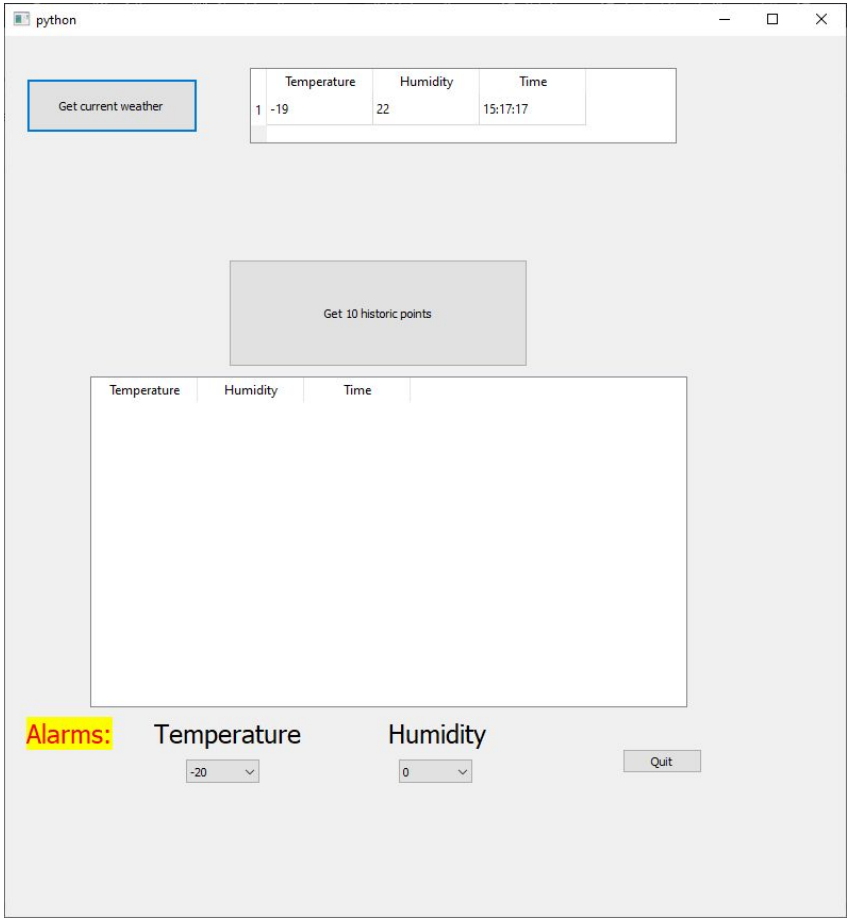
```
        mycursor.execute("SELECT * from weather ORDER BY id DESC LIMIT 1")
        myresult = mycursor.fetchall()

        self.tableWidget_2.setRowCount(1)
        tablerow=0
        for row in myresult:
            self.tableWidget_2.setItem(tablerow, 0, QtWidgets.QTableWidgetItem(ro
w[1]))
            self.tableWidget_2.setItem(tablerow, 1, QtWidgets.QTableWidgetItem(ro
w[2]))
            self.tableWidget_2.setItem(tablerow, 2, QtWidgets.QTableWidgetItem(ro
w[3]))
            tablerow+=1


    def button2_clicked(self):
        mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="123",
        database="weather"
        )

        mycursor = mydb.cursor()
        mycursor.execute("SELECT * from weather ORDER BY id DESC LIMIT 10")
        myresult = mycursor.fetchall()

        self.tableWidget.setRowCount(10)
        tablerow=0
        for row in myresult:
            self.tableWidget.setItem(tablerow, 0, QtWidgets.QTableWidgetItem(row[
1]))
            self.tableWidget.setItem(tablerow, 1, QtWidgets.QTableWidgetItem(row[
2]))
            self.tableWidget.setItem(tablerow, 2, QtWidgets.QTableWidgetItem(row[
3]))
            tablerow+=1

# main
app = QApplication(sys.argv)
mainwindow = MainWindow()
widget = QtWidgets.QStackedWidget()
widget.addWidget(mainwindow)
widget.show()
```

```
try:
    sys.exit(app.exec_())
except:
    print("Exiting")
```

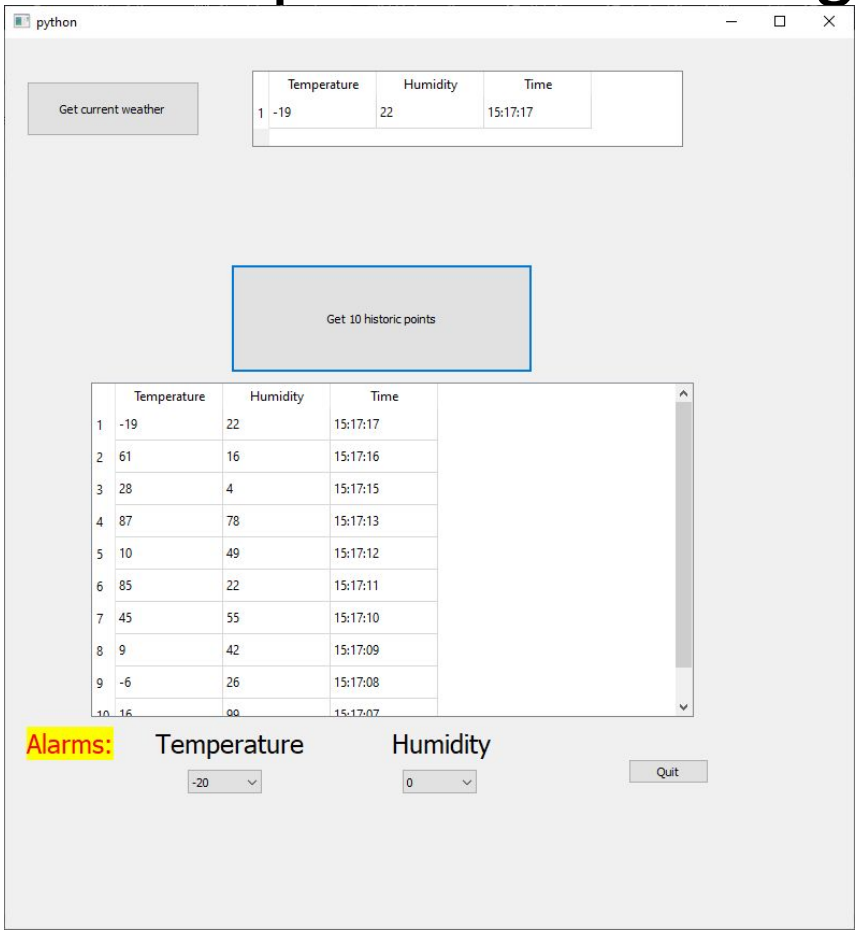
UI startup


python
Get current weather
Temperature
Humidity
Time
Get 10 historic points
Temperature
Humidity
Time
Alarms:
Temperature
-20
Humidity
0
Quit

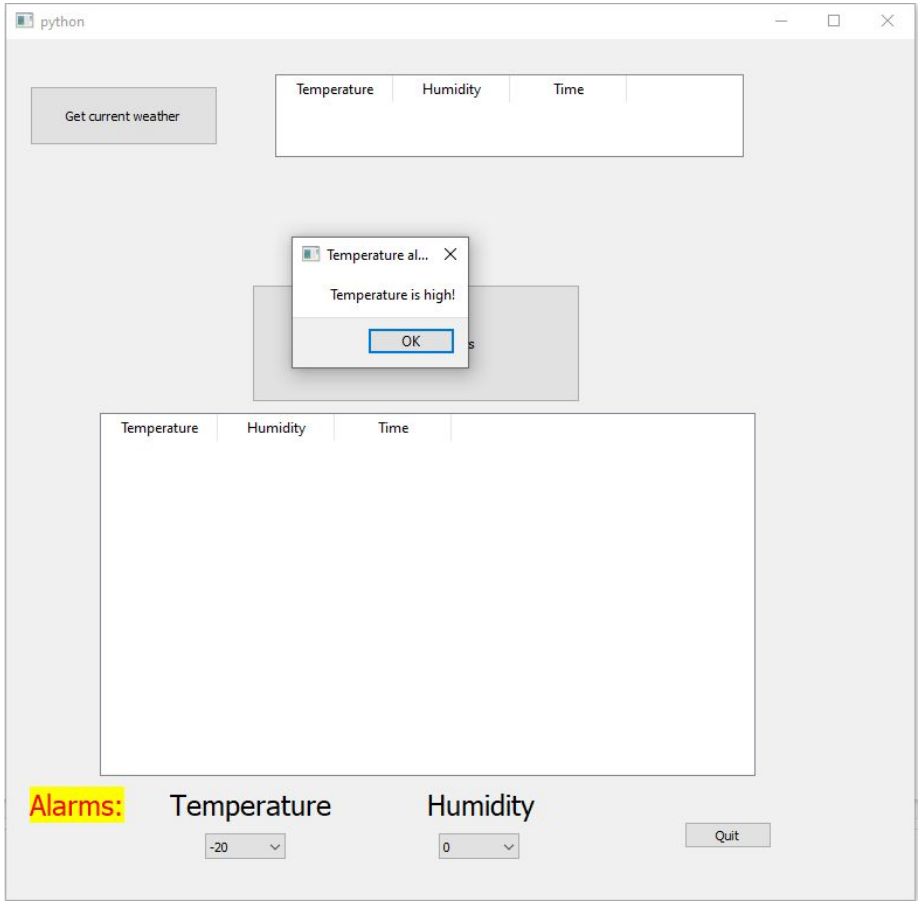
Single temperature reading



Ten temperature readings



Alarm


python
Get current weather
Temperature
Humidity
Time
Temperature al...
Temperature is high!
OK
Temperature
Humidity
Time
Alarms:
Temperature
-20
Humidity
0
Quit