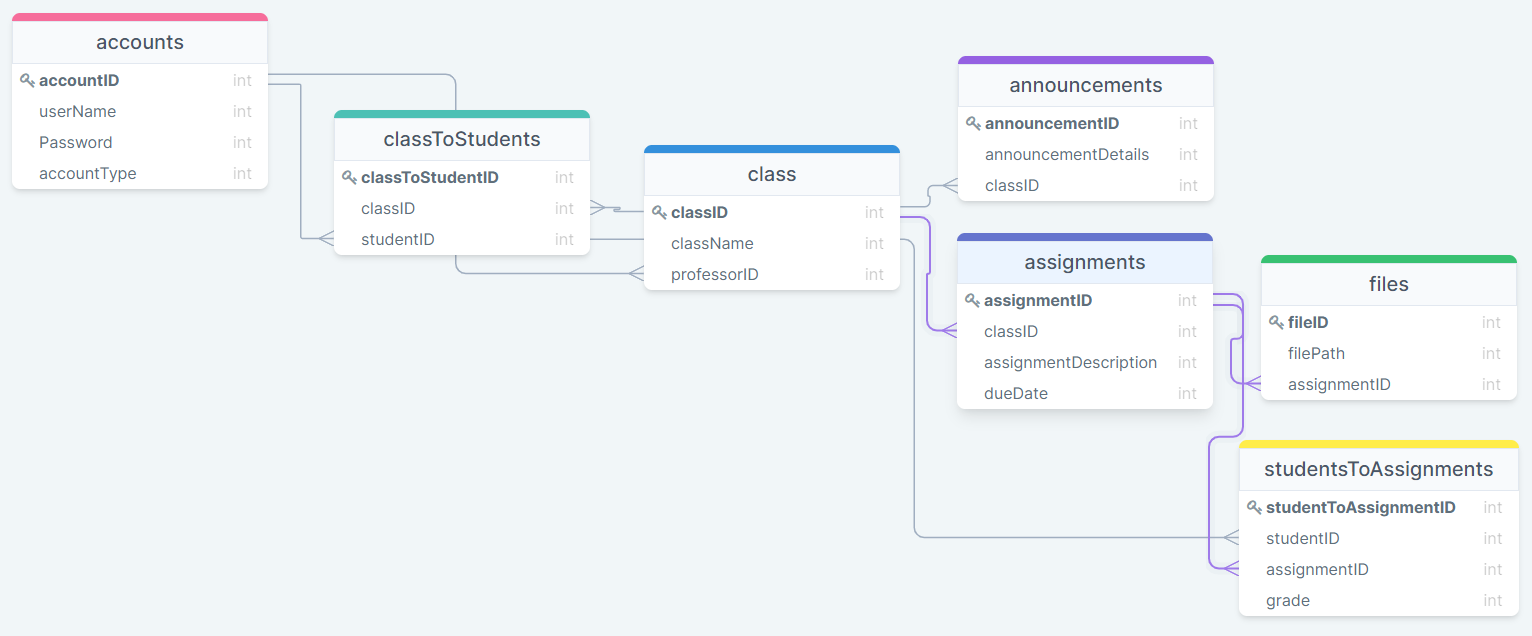
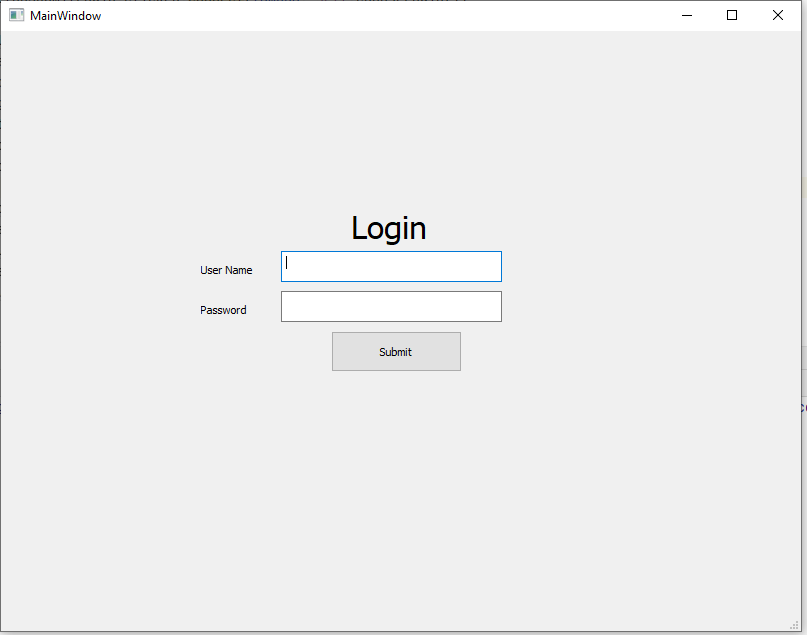
Final Project

DB Schema:



Screen Shots:



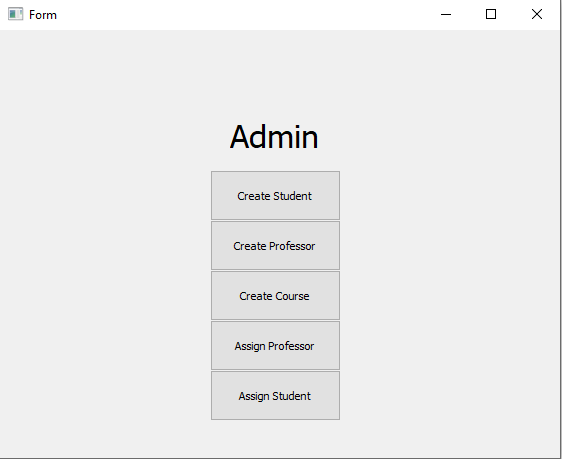
The main login window is where you enter your username and password.

If you are a student the submit button will take you to the student view.

If you are a professor the submit button will take you to the professor view.

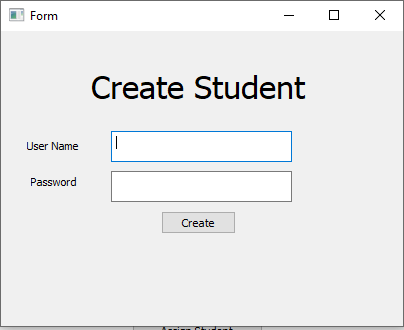
If you are a admin the submit button will take you to the admin view.

If you enter the wrong username or password the submit button will take you nowhere.

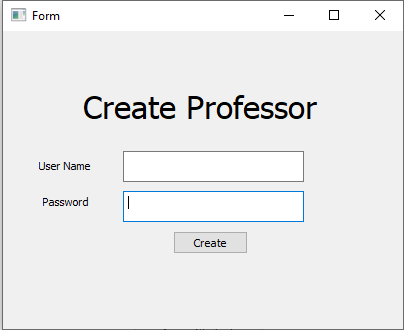


The admin form has options to

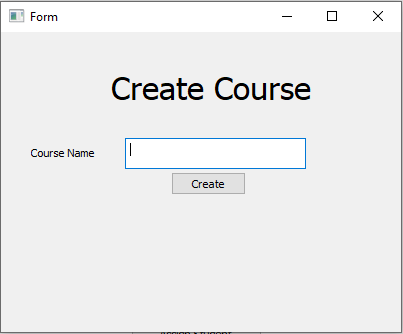
1. Create a student
2. Create a professor
3. Create a course
4. Assign a professor to a course
5. Assign a student to a course



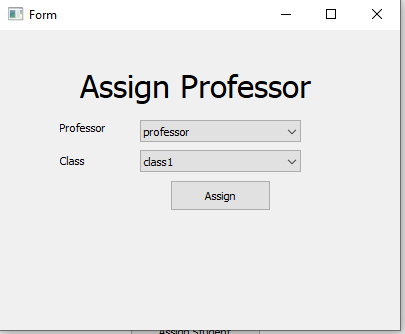
The create student form allows you to add a student to the database



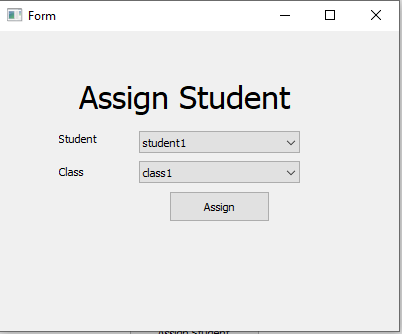
The create professor form allows you to add a professor to the database



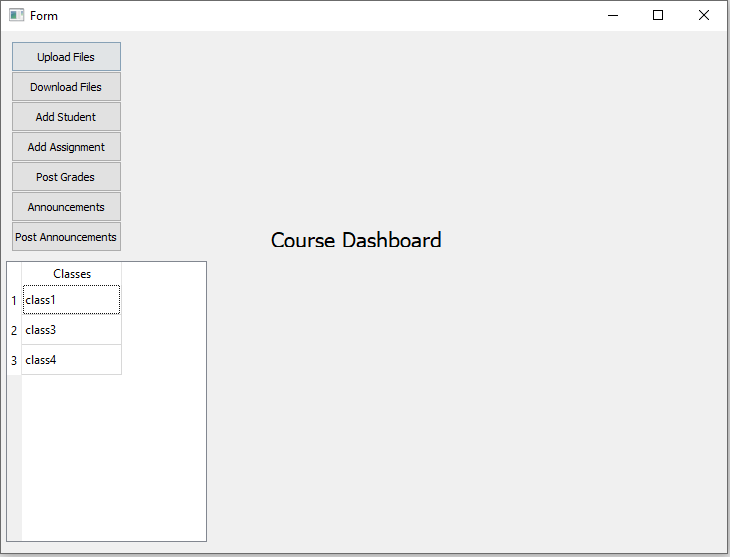
The create course allows you to create a course and add it to the database



The assign professor allows you to pick any professor and any class and add that professor as teacher of the class



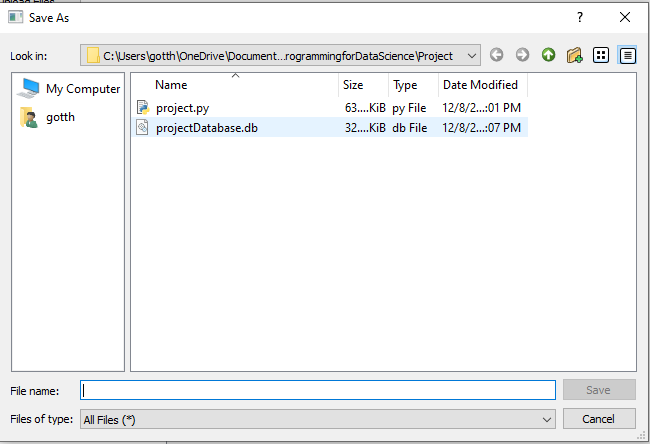
Assign student allows you to select any student and any class and add the student to the class in the database



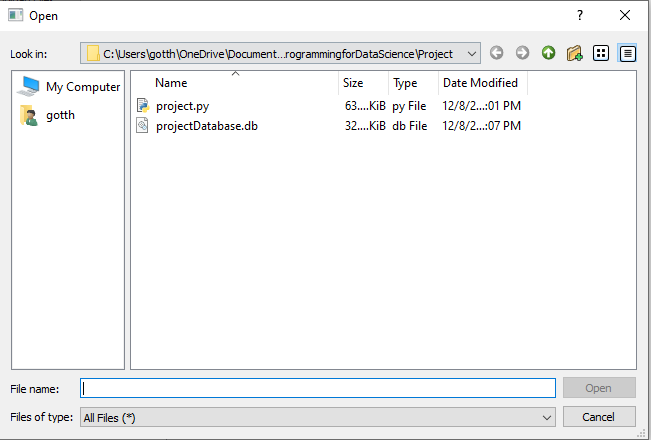
The professor view shows what classes the professor is currently teaching.

The professor also has the options to:

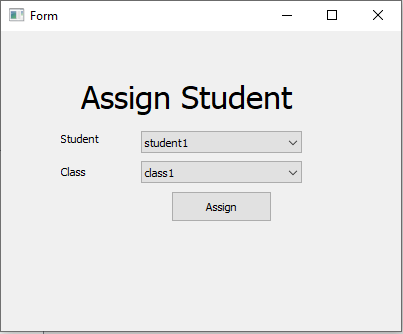
1. Upload a file
2. Download a file
3. Add a student to a class
4. Add an assignment to one of his classes
5. Add grades to one of his assignments
6. View any announcements in any of his classes
7. Post any announcements in any of his classes



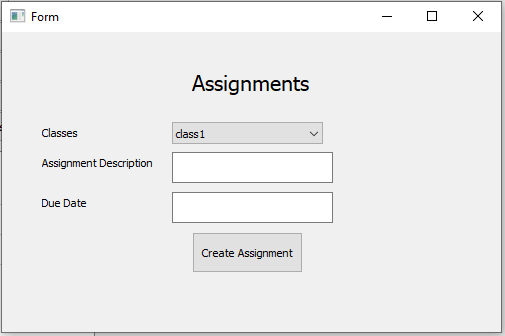
The upload allows you to upload a file to the database and the project folder



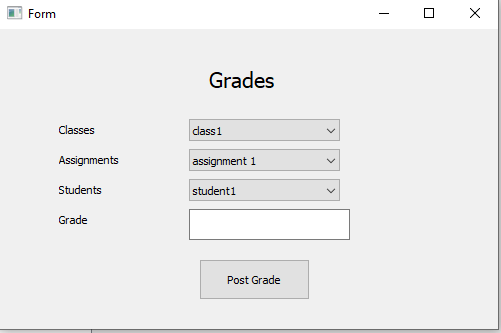
The download button allows you to open any file in the project



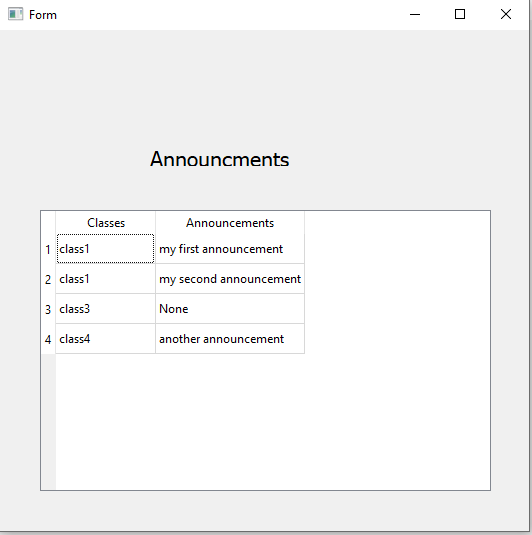
The professor is able to add a student to any one of his classes



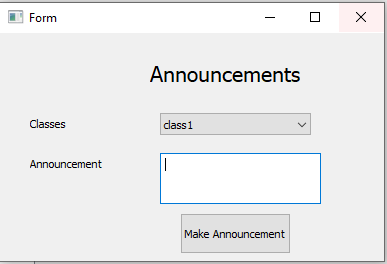
The professor is able to add an assignment to any one of his classes along with the due date



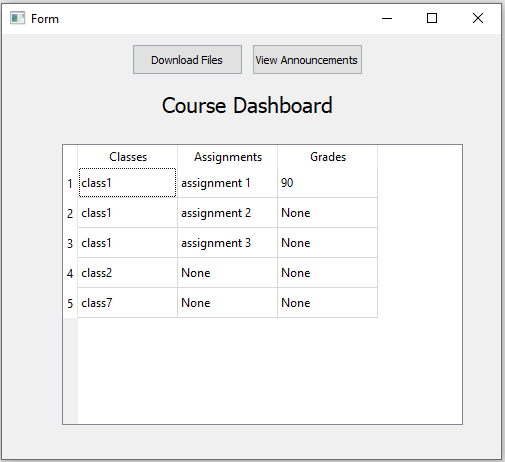
The professor is able to add grades to any one of his classes assignments for any student in that class



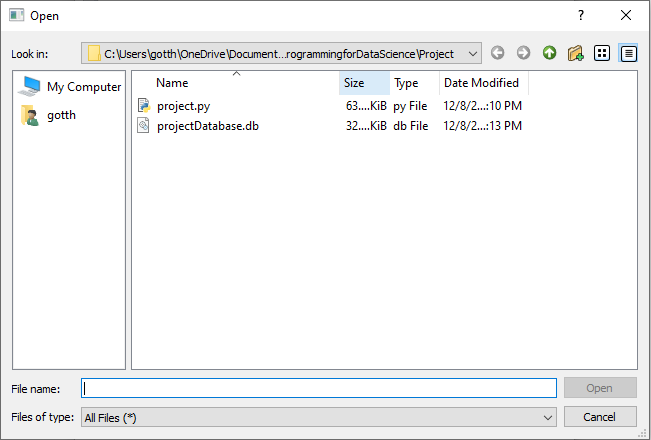
The professor is able to look at any announcement made in any class he is assigned to



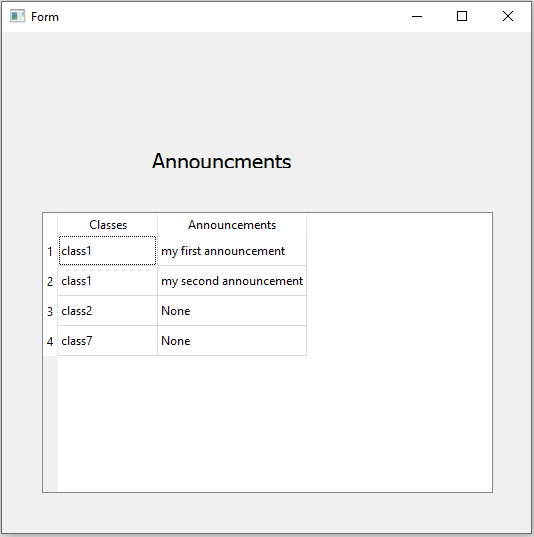
The professor is able to make an announcement to all of his students in any class he teaches



The student is able to see all of his classes, all of his assignments, and all of the grades his assignments currently have gotten for the course

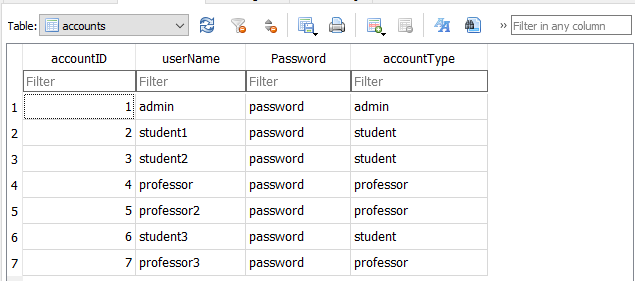


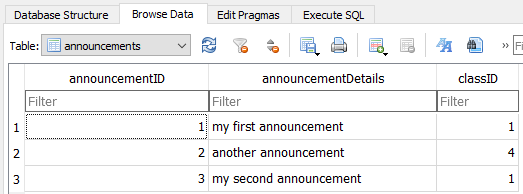
The student is able to download the files the professor has placed in the folder

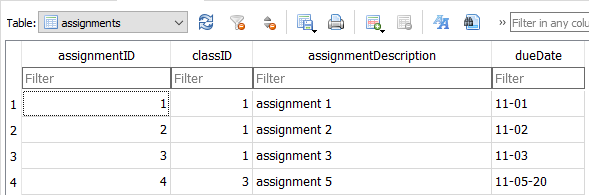


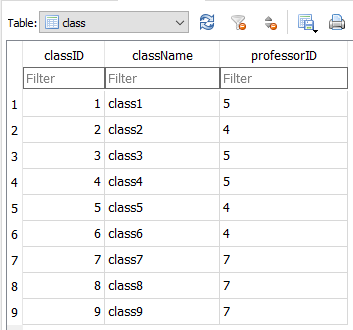
The student is able to see any announcement from any of his professors in any of his classes

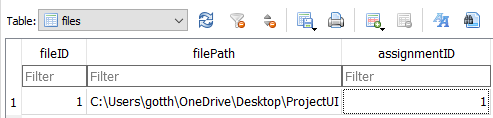
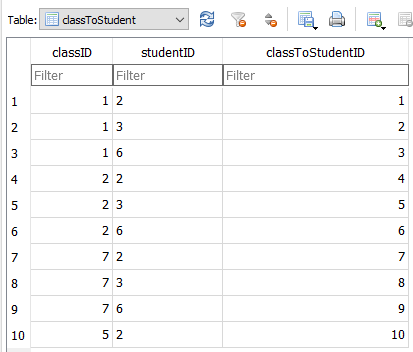
The data currently in the database is

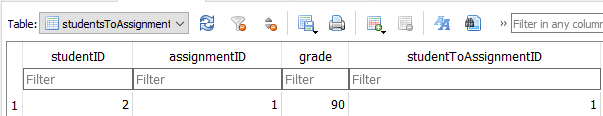












The project code starts here:

import sqlite3  
from PyQt5 import QtCore, QtGui, QtWidgets  
from PyQt5.QtGui import QPixmap  
from PyQt5.QtWidgets import QApplication, QMainWindow, QFileDialog, QLabel, QDialog  
import sys  
  
  
class Ui\_MainWindow(object):  
 def setupUi(self, MainWindow):  
  
 *#create database* self.database = DatabaseHelper()  
 self.database.CreateTables()  
 self.database.close()  
  
 MainWindow.setObjectName(**"MainWindow"**)  
 MainWindow.resize(800, 600)  
 self.centralwidget = QtWidgets.QWidget(MainWindow)  
 self.centralwidget.setObjectName(**"centralwidget"**)  
 self.Login = QtWidgets.QLabel(self.centralwidget)  
 self.Login.setGeometry(QtCore.QRect(350, 160, 101, 71))  
 font = QtGui.QFont()  
 font.setPointSize(24)  
 self.Login.setFont(font)  
 self.Login.setObjectName(**"Login"**)  
 self.textUserName = QtWidgets.QTextEdit(self.centralwidget)  
 self.textUserName.setGeometry(QtCore.QRect(280, 220, 221, 31))  
 self.textUserName.setObjectName(**"textUserName"**)  
 self.textPassword = QtWidgets.QTextEdit(self.centralwidget)  
 self.textPassword.setGeometry(QtCore.QRect(280, 260, 221, 31))  
 self.textPassword.setObjectName(**"textPassword"**)  
 *# self.comboUserType = QtWidgets.QComboBox(self.centralwidget)  
 # self.comboUserType.setGeometry(QtCore.QRect(520, 220, 151, 31))  
 # self.comboUserType.setObjectName("comboUserType")* self.buttonSubmitLogin = QtWidgets.QPushButton(self.centralwidget)  
 self.buttonSubmitLogin.setGeometry(QtCore.QRect(330, 300, 131, 41))  
 self.buttonSubmitLogin.setObjectName(**"buttonSubmitLogin"**)  
 self.labelUserName = QtWidgets.QLabel(self.centralwidget)  
 self.labelUserName.setGeometry(QtCore.QRect(200, 230, 71, 16))  
 self.labelUserName.setObjectName(**"labelUserName"**)  
 self.labelPassword = QtWidgets.QLabel(self.centralwidget)  
 self.labelPassword.setGeometry(QtCore.QRect(200, 270, 71, 16))  
 self.labelPassword.setObjectName(**"labelPassword"**)  
 MainWindow.setCentralWidget(self.centralwidget)  
 self.menubar = QtWidgets.QMenuBar(MainWindow)  
 self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 22))  
 self.menubar.setObjectName(**"menubar"**)  
 MainWindow.setMenuBar(self.menubar)  
 self.statusbar = QtWidgets.QStatusBar(MainWindow)  
 self.statusbar.setObjectName(**"statusbar"**)  
 MainWindow.setStatusBar(self.statusbar)  
  
 self.retranslateUi(MainWindow)  
 QtCore.QMetaObject.connectSlotsByName(MainWindow)  
  
  
 *#MainWindow.show()* try:  
 self.buttonSubmitLogin.clicked.connect(lambda: self.chooseLogin())  
 except:  
 print(**"The main window went critically wrong"**)  
  
 def retranslateUi(self, MainWindow):  
 \_translate = QtCore.QCoreApplication.translate  
 MainWindow.setWindowTitle(\_translate(**"MainWindow"**, **"MainWindow"**))  
 self.Login.setText(\_translate(**"MainWindow"**, **"Login"**))  
 self.buttonSubmitLogin.setText(\_translate(**"MainWindow"**, **"Submit"**))  
 self.labelUserName.setText(\_translate(**"MainWindow"**, **"User Name"**))  
 self.labelPassword.setText(\_translate(**"MainWindow"**, **"Password"**))  
  
 def chooseLogin(self):  
 database = DatabaseHelper()  
 global userName  
 userName = self.textUserName.toPlainText()  
 *#print(userName)* query = **"SELECT accountType from accounts where userName = '"** + userName + **"'"** global accountType  
 accountType = database.select(query)  
  
 queryPassword = **"Select password from accounts where userName = '"** + userName + **"'"** password = database.select(queryPassword)  
 if(password[0][0] == self.textPassword.toPlainText()):  
 *#print(accountType[0][0])* if(accountType[0][0] == **"admin"**):  
 self.showAdmin()  
 elif(accountType[0][0] == **"professor"**):  
 self.showProfessor(userName)  
 elif(accountType[0][0] == **"student"**):  
 self.showStudent()  
 else:  
 print(**"This is a bad login, try again"**)  
 else:  
 print(**"I'm sorry the password was wrong"**)  
  
 def showAdmin(self):  
 *#print("clicked")* self.adminWindow = QtWidgets.QMainWindow()  
 self.adminUI = Ui\_FormAdmin()  
 self.adminUI.setupUi(self.adminWindow)  
 self.adminWindow.show()  
  
 def showStudent(self):  
 *#print("clicked")* self.FormStudentView = QtWidgets.QWidget()  
 self.studentViewui = Ui\_FormStudentView()  
 self.studentViewui.setupUi(self.FormStudentView)  
 self.FormStudentView.show()  
  
 def showProfessor(self, userName):  
 *#print("clicked")* self.FormProfessorView = QtWidgets.QMainWindow()  
 self.professorViewUI = Ui\_FormProfessorView()  
 self.professorViewUI.setupUi(self.FormProfessorView, userName)  
 self.FormProfessorView.show()  
  
 *# def printMessage(self):  
 # print("Clicked")  
  
#done*class Ui\_FormAdmin(object):  
 def setupUi(self, FormAdmin):  
 FormAdmin.setObjectName(**"FormAdmin"**)  
 FormAdmin.resize(560, 428)  
 self.labelAdmin = QtWidgets.QLabel(FormAdmin)  
 self.labelAdmin.setGeometry(QtCore.QRect(230, 80, 101, 51))  
 font = QtGui.QFont()  
 font.setPointSize(24)  
 self.labelAdmin.setFont(font)  
 self.labelAdmin.setObjectName(**"labelAdmin"**)  
 self.buttonCreateStudent = QtWidgets.QPushButton(FormAdmin)  
 self.buttonCreateStudent.setGeometry(QtCore.QRect(210, 140, 131, 51))  
 self.buttonCreateStudent.setObjectName(**"buttonCreateStudent"**)  
 self.buttonCreateProfessor = QtWidgets.QPushButton(FormAdmin)  
 self.buttonCreateProfessor.setGeometry(QtCore.QRect(210, 190, 131, 51))  
 self.buttonCreateProfessor.setObjectName(**"buttonCreateProfessor"**)  
 self.buttonAssignCourse = QtWidgets.QPushButton(FormAdmin)  
 self.buttonAssignCourse.setGeometry(QtCore.QRect(210, 290, 131, 51))  
 self.buttonAssignCourse.setObjectName(**"buttonAssignCourse"**)  
 self.buttonCreateCourse = QtWidgets.QPushButton(FormAdmin)  
 self.buttonCreateCourse.setGeometry(QtCore.QRect(210, 240, 131, 51))  
 self.buttonCreateCourse.setObjectName(**"buttonCreateCourse"**)  
 self.buttonAssignStudent = QtWidgets.QPushButton(FormAdmin)  
 self.buttonAssignStudent.setGeometry(QtCore.QRect(210, 340, 131, 51))  
 self.buttonAssignStudent.setObjectName(**"buttonAssignStudent"**)  
  
 self.buttonCreateStudent.clicked.connect(lambda: self.showCreateStudent())  
 self.buttonCreateProfessor.clicked.connect(lambda: self.showCreateProfessor())  
 self.buttonAssignCourse.clicked.connect(lambda: self.showAssignProfessor())  
 self.buttonCreateCourse.clicked.connect(lambda: self.showCreateCourse())  
 self.buttonAssignStudent.clicked.connect(lambda: self.showAssignStudent())  
  
 self.retranslateUi(FormAdmin)  
 QtCore.QMetaObject.connectSlotsByName(FormAdmin)  
  
 def retranslateUi(self, FormAdmin):  
 \_translate = QtCore.QCoreApplication.translate  
 FormAdmin.setWindowTitle(\_translate(**"FormAdmin"**, **"Form"**))  
 self.labelAdmin.setText(\_translate(**"FormAdmin"**, **"Admin"**))  
 self.buttonCreateStudent.setText(\_translate(**"FormAdmin"**, **"Create Student"**))  
 self.buttonCreateProfessor.setText(\_translate(**"FormAdmin"**, **"Create Professor"**))  
 self.buttonAssignCourse.setText(\_translate(**"FormAdmin"**, **"Assign Professor"**))  
 self.buttonCreateCourse.setText(\_translate(**"FormAdmin"**, **"Create Course"**))  
 self.buttonAssignStudent.setText(\_translate(**"FormAdmin"**, **"Assign Student"**))  
  
 def showCreateStudent(self):  
 *# print("clicked")* self.FormCreateStudent = QtWidgets.QMainWindow()  
 self.createStudentUI = Ui\_FormCreateStudent()  
 self.createStudentUI.setupUi(self.FormCreateStudent)  
 self.FormCreateStudent.show()  
  
 def showCreateProfessor(self):  
 *#print("clicked")* self.FormCreateProfessor = QtWidgets.QMainWindow()  
 self.createProfessorUI = Ui\_FormCreateProfessor()  
 self.createProfessorUI.setupUi(self.FormCreateProfessor)  
 self.FormCreateProfessor.show()  
  
 def showCreateCourse(self):  
 *#print("clicked")* self.FormCreateCourse = QtWidgets.QMainWindow()  
 self.createCourseUI = Ui\_FormCreateCourse()  
 self.createCourseUI.setupUi(self.FormCreateCourse)  
 self.FormCreateCourse.show()  
  
 def showAssignStudent(self):  
 *#print("clicked")* self.FormAssignStudent = QtWidgets.QMainWindow()  
 self.assignStudentUI = Ui\_FormAssignStudent()  
 self.assignStudentUI.setupUi(self.FormAssignStudent)  
 self.FormAssignStudent.show()  
  
 def showAssignProfessor(self):  
 *#print("clicked")* self.FormAssignProfessor = QtWidgets.QMainWindow()  
 self.AssignProfessorUI = Ui\_FormAssignProfessor()  
 self.AssignProfessorUI.setupUi(self.FormAssignProfessor)  
 self.FormAssignProfessor.show()  
  
*#done*class Ui\_FormAssignProfessor(object):  
 def setupUi(self, FormAssignProfessor):  
 FormAssignProfessor.setObjectName(**"FormAssignProfessor"**)  
 FormAssignProfessor.resize(400, 300)  
 self.labelAssignProfessor = QtWidgets.QLabel(FormAssignProfessor)  
 self.labelAssignProfessor.setGeometry(QtCore.QRect(80, 10, 241, 91))  
 font = QtGui.QFont()  
 font.setPointSize(24)  
 self.labelAssignProfessor.setFont(font)  
 self.labelAssignProfessor.setObjectName(**"labelAssignProfessor"**)  
 self.comboAssignProfessor = QtWidgets.QComboBox(FormAssignProfessor)  
 self.comboAssignProfessor.setGeometry(QtCore.QRect(140, 90, 161, 22))  
 self.comboAssignProfessor.setObjectName(**"comboAssignProfessor"**)  
 self.labelAssignProfessorToClass = QtWidgets.QLabel(FormAssignProfessor)  
 self.labelAssignProfessorToClass.setGeometry(QtCore.QRect(60, 90, 47, 14))  
 self.labelAssignProfessorToClass.setObjectName(**"labelAssignProfessorToClass"**)  
 self.comboAssignClassToProfessor = QtWidgets.QComboBox(FormAssignProfessor)  
 self.comboAssignClassToProfessor.setGeometry(QtCore.QRect(140, 120, 161, 22))  
 self.comboAssignClassToProfessor.setObjectName(**"comboAssignClassToProfessor"**)  
 self.labelAssignClassToProfessor = QtWidgets.QLabel(FormAssignProfessor)  
 self.labelAssignClassToProfessor.setGeometry(QtCore.QRect(60, 120, 51, 20))  
 self.labelAssignClassToProfessor.setObjectName(**"labelAssignClassToProfessor"**)  
 self.buttonAssignProfessor = QtWidgets.QPushButton(FormAssignProfessor)  
 self.buttonAssignProfessor.setGeometry(QtCore.QRect(170, 150, 101, 31))  
 self.buttonAssignProfessor.setObjectName(**"buttonAssignProfessor"**)  
  
 self.retranslateUi(FormAssignProfessor)  
  
 self.buttonAssignProfessor.clicked.connect(lambda: self.assignProfessor())  
  
 QtCore.QMetaObject.connectSlotsByName(FormAssignProfessor)  
  
 def retranslateUi(self, FormAssignProfessor):  
 \_translate = QtCore.QCoreApplication.translate  
 FormAssignProfessor.setWindowTitle(\_translate(**"FormAssignProfessor"**, **"Form"**))  
 self.labelAssignProfessor.setText(\_translate(**"FormAssignProfessor"**, **"Assign Professor"**))  
 self.labelAssignProfessorToClass.setText(\_translate(**"FormAssignProfessor"**, **"Professor"**))  
 self.labelAssignClassToProfessor.setText(\_translate(**"FormAssignProfessor"**, **"Class"**))  
 self.buttonAssignProfessor.setText(\_translate(**"FormAssignProfessor"**, **"Assign"**))  
  
 database = DatabaseHelper()  
 query = **"SELECT userName from accounts where accountType = 'professor'"** answer = database.select(query)  
 for item in answer:  
 self.comboAssignProfessor.addItems(item)  
  
 classQuery = **"SELECT className from class"** classAnswer = database.select(classQuery)  
 for item in classAnswer:  
 self.comboAssignClassToProfessor.addItems(item)  
  
 database.close()  
  
 def assignProfessor(self):  
 professor = self.comboAssignProfessor.currentText()  
 classData = str(self.comboAssignClassToProfessor.currentText())  
 classQuery = **"SELECT classID from class Where '"** + classData + **"' = className"** print(**"ClassQuery = "** + classQuery)  
 professorQuery = **"SELECT accountID from accounts Where userName = '"** + professor + **"' and accountType = 'professor'"** print(**"Professor Query = "** + professorQuery)  
  
 *#print("Creating Professor, " + username + ", " + password)  
 #query = "INSERT INTO accounts (userName, Password, accountType) VALUES ('" + username + "', '" + password + "', 'professor')"* database = DatabaseHelper()  
 classID = database.select(classQuery)  
 professorID = database.select(professorQuery)  
 *#print(type(studentID[0][0]))  
 # print(studentID[0][0])* assignProfessorQuery = **"UPDATE class SET professorID = "** + str(professorID[0][0]) + **" "** \  
 **"WHERE classID = "** + str(classID[0][0])  
  
 *# assignProfessorQuery = "INSERT INTO class (professorID) VALUES (" + str(professorID[0][0]) + ") " \  
 # "Where classID = " + str(classID[0][0])* print(**"Assign professor query = "** + assignProfessorQuery)  
  
 database.edit(assignProfessorQuery)  
 database.close()  
  
*#done*class Ui\_FormAssignStudent(object):  
 def setupUi(self, FormAssignStudent):  
 FormAssignStudent.setObjectName(**"FormAssignStudent"**)  
 FormAssignStudent.resize(400, 300)  
 self.comboAssignClassToStudent = QtWidgets.QComboBox(FormAssignStudent)  
 self.comboAssignClassToStudent.setGeometry(QtCore.QRect(140, 130, 161, 22))  
 self.comboAssignClassToStudent.setObjectName(**"comboAssignClassToStudent"**)  
 self.buttonAssignStudent = QtWidgets.QPushButton(FormAssignStudent)  
 self.buttonAssignStudent.setGeometry(QtCore.QRect(170, 160, 101, 31))  
 self.buttonAssignStudent.setObjectName(**"buttonAssignStudent"**)  
 self.labelAssignStudent = QtWidgets.QLabel(FormAssignStudent)  
 self.labelAssignStudent.setGeometry(QtCore.QRect(80, 20, 241, 91))  
 font = QtGui.QFont()  
 font.setPointSize(24)  
 self.labelAssignStudent.setFont(font)  
 self.labelAssignStudent.setObjectName(**"labelAssignStudent"**)  
 self.comboAssignStudent = QtWidgets.QComboBox(FormAssignStudent)  
 self.comboAssignStudent.setGeometry(QtCore.QRect(140, 100, 161, 22))  
 self.comboAssignStudent.setObjectName(**"comboAssignStudent"**)  
 self.labelAssignClassToStudent = QtWidgets.QLabel(FormAssignStudent)  
 self.labelAssignClassToStudent.setGeometry(QtCore.QRect(60, 130, 51, 20))  
 self.labelAssignClassToStudent.setObjectName(**"labelAssignClassToStudent"**)  
 self.labelAssignStudentToClass = QtWidgets.QLabel(FormAssignStudent)  
 self.labelAssignStudentToClass.setGeometry(QtCore.QRect(60, 100, 47, 14))  
 self.labelAssignStudentToClass.setObjectName(**"labelAssignStudentToClass"**)  
  
 self.retranslateUi(FormAssignStudent)  
 self.buttonAssignStudent.clicked.connect(lambda: self.assignStudent())  
  
 QtCore.QMetaObject.connectSlotsByName(FormAssignStudent)  
  
 def retranslateUi(self, FormAssignStudent):  
 \_translate = QtCore.QCoreApplication.translate  
 FormAssignStudent.setWindowTitle(\_translate(**"FormAssignStudent"**, **"Form"**))  
 self.buttonAssignStudent.setText(\_translate(**"FormAssignStudent"**, **"Assign"**))  
 self.labelAssignStudent.setText(\_translate(**"FormAssignStudent"**, **"Assign Student"**))  
 self.labelAssignClassToStudent.setText(\_translate(**"FormAssignStudent"**, **"Class"**))  
 self.labelAssignStudentToClass.setText(\_translate(**"FormAssignStudent"**, **"Student"**))  
 database = DatabaseHelper()  
 query = **"SELECT userName from accounts where accountType = 'student'"** answer = database.select(query)  
 for item in answer:  
 self.comboAssignStudent.addItems(item)  
  
 classQuery = **"SELECT className from class"** classAnswer = database.select(classQuery)  
 for item in classAnswer:  
 self.comboAssignClassToStudent.addItems(item)  
  
 database.close()  
  
 def assignStudent(self):  
 student = self.comboAssignStudent.currentText()  
 classData = str(self.comboAssignClassToStudent.currentText())  
 classQuery = **"SELECT classID from class Where '"** + classData + **"' = className"** print(**"ClassQuery = "** + classQuery)  
 studentQuery = **"SELECT accountID from accounts Where userName = '"** + student + **"' and accountType = 'student'"** print(**"Student Query = "** + studentQuery)  
  
 *#print("Creating Professor, " + username + ", " + password)  
 #query = "INSERT INTO accounts (userName, Password, accountType) VALUES ('" + username + "', '" + password + "', 'professor')"* database = DatabaseHelper()  
 classID = database.select(classQuery)  
 studentID = database.select(studentQuery)  
 *#print(type(studentID[0][0]))  
 # print(studentID[0][0])* assignStudentQuery = **"INSERT INTO classToStudent (classID, studentID) "** \  
 **"VALUES ("** + str(classID[0][0]) + **", "** + str(studentID[0][0]) + **")"** print(**"Assign student query = "** + assignStudentQuery)  
  
 database.edit(assignStudentQuery)  
 database.close()  
  
*#done*class Ui\_FormCreateCourse(object):  
 def setupUi(self, FormCreateCourse):  
 FormCreateCourse.setObjectName(**"FormCreateCourse"**)  
 FormCreateCourse.resize(400, 300)  
 self.labelCreateCourse = QtWidgets.QLabel(FormCreateCourse)  
 self.labelCreateCourse.setGeometry(QtCore.QRect(110, 10, 241, 91))  
 font = QtGui.QFont()  
 font.setPointSize(24)  
 self.labelCreateCourse.setFont(font)  
 self.labelCreateCourse.setObjectName(**"labelCreateCourse"**)  
 self.textCourseName = QtWidgets.QTextEdit(FormCreateCourse)  
 self.textCourseName.setGeometry(QtCore.QRect(124, 106, 181, 31))  
 self.textCourseName.setObjectName(**"textCourseName"**)  
 self.buttonCreateCourse = QtWidgets.QPushButton(FormCreateCourse)  
 self.buttonCreateCourse.setGeometry(QtCore.QRect(170, 140, 75, 23))  
 self.buttonCreateCourse.setObjectName(**"buttonCreateCourse"**)  
 self.labelCourseName = QtWidgets.QLabel(FormCreateCourse)  
 self.labelCourseName.setGeometry(QtCore.QRect(30, 110, 71, 20))  
 self.labelCourseName.setObjectName(**"labelCourseName"**)  
  
 self.buttonCreateCourse.clicked.connect(lambda: self.createCourse())  
  
 self.retranslateUi(FormCreateCourse)  
 QtCore.QMetaObject.connectSlotsByName(FormCreateCourse)  
  
 def retranslateUi(self, FormCreateCourse):  
 \_translate = QtCore.QCoreApplication.translate  
 FormCreateCourse.setWindowTitle(\_translate(**"FormCreateCourse"**, **"Form"**))  
 self.labelCreateCourse.setText(\_translate(**"FormCreateCourse"**, **"Create Course"**))  
 self.buttonCreateCourse.setText(\_translate(**"FormCreateCourse"**, **"Create"**))  
 self.labelCourseName.setText(\_translate(**"FormCreateCourse"**, **"Course Name"**))  
  
 def createCourse(self):  
 *#print(self.textStudentUserName)* courseName = self.textCourseName.toPlainText()  
 print(**"Creating Course, "** + courseName)  
 query = **"INSERT INTO class (className) VALUES ('"** + courseName + **"')"** *#query = "INSERT INTO class (className, professorID) VALUES ('test3', 1)"* print(query)  
 database = DatabaseHelper()  
 database.edit(query)  
 database.close()  
  
*#done*class Ui\_FormCreateProfessor(object):  
 def setupUi(self, FormCreateProfessor):  
 FormCreateProfessor.setObjectName(**"FormCreateProfessor"**)  
 FormCreateProfessor.resize(400, 298)  
 self.textProfessorPassword = QtWidgets.QTextEdit(FormCreateProfessor)  
 self.textProfessorPassword.setGeometry(QtCore.QRect(120, 160, 181, 31))  
 self.textProfessorPassword.setObjectName(**"textProfessorPassword"**)  
 self.textProfessorUserName = QtWidgets.QTextEdit(FormCreateProfessor)  
 self.textProfessorUserName.setGeometry(QtCore.QRect(120, 120, 181, 31))  
 self.textProfessorUserName.setObjectName(**"textProfessorUserName"**)  
 self.labelProfessorUserName = QtWidgets.QLabel(FormCreateProfessor)  
 self.labelProfessorUserName.setGeometry(QtCore.QRect(36, 124, 61, 20))  
 self.labelProfessorUserName.setObjectName(**"labelProfessorUserName"**)  
 self.labelProfessorPassword = QtWidgets.QLabel(FormCreateProfessor)  
 self.labelProfessorPassword.setGeometry(QtCore.QRect(40, 160, 61, 20))  
 self.labelProfessorPassword.setObjectName(**"labelProfessorPassword"**)  
 self.buttonCreateProfessor = QtWidgets.QPushButton(FormCreateProfessor)  
 self.buttonCreateProfessor.setGeometry(QtCore.QRect(170, 200, 75, 23))  
 self.buttonCreateProfessor.setObjectName(**"buttonCreateProfessor"**)  
 self.labelCreateProfessor = QtWidgets.QLabel(FormCreateProfessor)  
 self.labelCreateProfessor.setGeometry(QtCore.QRect(80, 30, 241, 91))  
 font = QtGui.QFont()  
 font.setPointSize(24)  
 self.labelCreateProfessor.setFont(font)  
 self.labelCreateProfessor.setObjectName(**"labelCreateProfessor"**)  
  
 self.retranslateUi(FormCreateProfessor)  
 self.buttonCreateProfessor.clicked.connect(lambda: self.createProfessor())  
 QtCore.QMetaObject.connectSlotsByName(FormCreateProfessor)  
  
 def retranslateUi(self, FormCreateProfessor):  
 \_translate = QtCore.QCoreApplication.translate  
 FormCreateProfessor.setWindowTitle(\_translate(**"FormCreateProfessor"**, **"Form"**))  
 self.labelProfessorUserName.setText(\_translate(**"FormCreateProfessor"**, **"User Name"**))  
 self.labelProfessorPassword.setText(\_translate(**"FormCreateProfessor"**, **"Password"**))  
 self.buttonCreateProfessor.setText(\_translate(**"FormCreateProfessor"**, **"Create"**))  
 self.labelCreateProfessor.setText(\_translate(**"FormCreateProfessor"**, **"Create Professor"**))  
  
 def createProfessor(self):  
 *#print(self.textStudentUserName)* username = self.textProfessorUserName.toPlainText()  
 password = self.textProfessorPassword.toPlainText()  
 print(**"Creating Professor, "** + username + **", "** + password)  
 query = **"INSERT INTO accounts (userName, Password, accountType) VALUES ('"** + username + **"', '"** + password + **"', 'professor')"** database = DatabaseHelper()  
 database.edit(query)  
 database.close()  
  
*#done*class Ui\_FormCreateStudent(object):  
 def setupUi(self, FormCreateStudent):  
 FormCreateStudent.setObjectName(**"FormCreateStudent"**)  
 FormCreateStudent.resize(402, 295)  
 self.labelCreateStudent = QtWidgets.QLabel(FormCreateStudent)  
 self.labelCreateStudent.setGeometry(QtCore.QRect(90, 10, 221, 91))  
 font = QtGui.QFont()  
 font.setPointSize(24)  
 self.labelCreateStudent.setFont(font)  
 self.labelCreateStudent.setObjectName(**"labelCreateStudent"**)  
 self.textStudentUserName = QtWidgets.QTextEdit(FormCreateStudent)  
 self.textStudentUserName.setGeometry(QtCore.QRect(110, 100, 181, 31))  
 self.textStudentUserName.setObjectName(**"textStudentUserName"**)  
 self.textStudentPassword = QtWidgets.QTextEdit(FormCreateStudent)  
 self.textStudentPassword.setGeometry(QtCore.QRect(110, 140, 181, 31))  
 self.textStudentPassword.setObjectName(**"textStudentPassword"**)  
 self.buttonCreateStudent = QtWidgets.QPushButton(FormCreateStudent)  
 self.buttonCreateStudent.setGeometry(QtCore.QRect(160, 180, 75, 23))  
 self.buttonCreateStudent.setObjectName(**"buttonCreateStudent"**)  
 self.labelStudentUserName = QtWidgets.QLabel(FormCreateStudent)  
 self.labelStudentUserName.setGeometry(QtCore.QRect(26, 104, 61, 20))  
 self.labelStudentUserName.setObjectName(**"labelStudentUserName"**)  
 self.labelStudentPassword = QtWidgets.QLabel(FormCreateStudent)  
 self.labelStudentPassword.setGeometry(QtCore.QRect(30, 140, 61, 20))  
 self.labelStudentPassword.setObjectName(**"labelStudentPassword"**)  
  
 self.retranslateUi(FormCreateStudent)  
 self.buttonCreateStudent.clicked.connect(lambda: self.createStudent())  
 QtCore.QMetaObject.connectSlotsByName(FormCreateStudent)  
  
 def retranslateUi(self, FormCreateStudent):  
 \_translate = QtCore.QCoreApplication.translate  
 FormCreateStudent.setWindowTitle(\_translate(**"FormCreateStudent"**, **"Form"**))  
 self.labelCreateStudent.setText(\_translate(**"FormCreateStudent"**, **"Create Student"**))  
 self.buttonCreateStudent.setText(\_translate(**"FormCreateStudent"**, **"Create"**))  
 self.labelStudentUserName.setText(\_translate(**"FormCreateStudent"**, **"User Name"**))  
 self.labelStudentPassword.setText(\_translate(**"FormCreateStudent"**, **"Password"**))  
  
 def createStudent(self):  
 *#print(self.textStudentUserName)* username = self.textStudentUserName.toPlainText()  
 password = self.textStudentPassword.toPlainText()  
 print(**"Creating Student, "** + username + **", "** + password)  
 query = **"INSERT INTO accounts (userName, Password, accountType) VALUES ('"** + username + **"', '"** + password + **"', 'student')"** database = DatabaseHelper()  
 database.edit(query)  
 database.close()  
  
class Ui\_FormProfessorView(object):  
 def setupUi(self, FormProfessorView, userName):  
 self.userName = userName  
 FormProfessorView.setObjectName(**"FormProfessorView"**)  
 FormProfessorView.resize(726, 522)  
  
 self.viewTableWidget = QtWidgets.QTableWidget(FormProfessorView)  
 self.viewTableWidget.setGeometry(QtCore.QRect(5, 230, 201, 281))  
 self.viewTableWidget.setObjectName(**"viewTableWidget"**)  
 self.viewTableWidget.setColumnCount(1)  
 self.viewTableWidget.setRowCount(0)  
 item = QtWidgets.QTableWidgetItem()  
 self.viewTableWidget.setHorizontalHeaderItem(0, item)  
  
 self.buttonUploadFiles = QtWidgets.QPushButton(FormProfessorView)  
 self.buttonUploadFiles.setGeometry(QtCore.QRect(10, 10, 111, 31))  
 self.buttonUploadFiles.setObjectName(**"buttonUploadFiles"**)  
 self.buttonDownloadFiles = QtWidgets.QPushButton(FormProfessorView)  
 self.buttonDownloadFiles.setGeometry(QtCore.QRect(10, 40, 111, 31))  
 self.buttonDownloadFiles.setObjectName(**"buttonDownloadFiles"**)  
 self.labelCourseDashboard = QtWidgets.QLabel(FormProfessorView)  
 self.labelCourseDashboard.setGeometry(QtCore.QRect(270, 200, 241, 16))  
 font = QtGui.QFont()  
 font.setPointSize(16)  
 self.labelCourseDashboard.setFont(font)  
 self.labelCourseDashboard.setObjectName(**"labelCourseDashboard"**)  
 self.buttonAddStudent = QtWidgets.QPushButton(FormProfessorView)  
 self.buttonAddStudent.setGeometry(QtCore.QRect(10, 70, 111, 31))  
 self.buttonAddStudent.setObjectName(**"buttonAddStudent"**)  
 self.buttonAddAssignment = QtWidgets.QPushButton(FormProfessorView)  
 self.buttonAddAssignment.setGeometry(QtCore.QRect(10, 100, 111, 31))  
 self.buttonAddAssignment.setObjectName(**"buttonAddAssignment"**)  
 self.buttonPostGrades = QtWidgets.QPushButton(FormProfessorView)  
 self.buttonPostGrades.setGeometry(QtCore.QRect(10, 130, 111, 31))  
 self.buttonPostGrades.setObjectName(**"buttonPostGrades"**)  
 self.buttonViewAnnouncements = QtWidgets.QPushButton(FormProfessorView)  
 self.buttonViewAnnouncements.setGeometry(QtCore.QRect(10, 160, 111, 31))  
 self.buttonViewAnnouncements.setObjectName(**"buttonViewAnnouncements"**)  
 self.buttonPostAnnouncement = QtWidgets.QPushButton(FormProfessorView)  
 self.buttonPostAnnouncement.setGeometry(QtCore.QRect(10, 190, 111, 31))  
 self.buttonPostAnnouncement.setObjectName(**"buttonPostAnnouncement"**)  
  
 self.buttonUploadFiles.clicked.connect(lambda : self.FileDialog(forOpen=False) )  
 self.buttonDownloadFiles.clicked.connect(lambda : self.FileDialog())  
 self.buttonAddStudent.clicked.connect(lambda : self.showAssignStudent())  
 self.buttonAddAssignment.clicked.connect(lambda: self.showAssignments())  
 self.buttonPostGrades.clicked.connect(lambda: self.showGrades())  
 self.buttonPostAnnouncement.clicked.connect(lambda : self.showAddAnnouncements())  
 self.buttonViewAnnouncements.clicked.connect(lambda : self.showViewAnnouncements())  
  
 self.retranslateUi(FormProfessorView)  
 QtCore.QMetaObject.connectSlotsByName(FormProfessorView)  
  
 def retranslateUi(self, FormProfessorView):  
 \_translate = QtCore.QCoreApplication.translate  
 FormProfessorView.setWindowTitle(\_translate(**"FormProfessorView"**, **"Form"**))  
 item = self.viewTableWidget.horizontalHeaderItem(0)  
 item.setText(\_translate(**"FormProfessorView"**, **"Classes"**))  
 self.buttonUploadFiles.setText(\_translate(**"FormProfessorView"**, **"Upload Files"**))  
 self.buttonDownloadFiles.setText(\_translate(**"FormProfessorView"**, **"Download Files"**))  
 self.labelCourseDashboard.setText(\_translate(**"FormProfessorView"**, **"Course Dashboard"**))  
 self.buttonAddStudent.setText(\_translate(**"FormProfessorView"**, **"Add Student"**))  
 self.buttonAddAssignment.setText(\_translate(**"FormProfessorView"**, **"Add Assignment"**))  
 self.buttonPostGrades.setText(\_translate(**"FormProfessorView"**, **"Post Grades"**))  
 self.buttonViewAnnouncements.setText(\_translate(**"FormProfessorView"**, **"Announcements"**))  
 self.buttonPostAnnouncement.setText(\_translate(**"FormProfessorView"**, **"Post Announcements"**))  
 self.loadData()  
  
 def loadData(self):  
 database = DatabaseHelper()  
  
 query = **"SELECT class.className from class, accounts "** \  
 **"where accounts.userName = '"** + self.userName + **"' and class.professorID = accounts.accountID"** answer = database.select(query)  
  
 for rowCount, answer in enumerate(answer):  
 self.viewTableWidget.insertRow(rowCount)  
 for columnNumber, data in enumerate(answer):  
 cell = QtWidgets.QTableWidgetItem(str(data))  
 self.viewTableWidget.setItem(rowCount, columnNumber, cell)  
 database.close()  
  
 def openDialog(self):  
 print(**"Testing"**)  
 dialog = QFileDialog()  
 dialog.setFileMode(QFileDialog.AnyFile)  
 dialog.show()  
 *# #dialog.setNameFilter(tr("Images (\*.png \*.xpm \*.jpg)"))  
 # dialog.setViewMode(QFileDialog.Detail)  
 # if dialog.exec\_():  
 # fileNames = dialog.selectedFiles()  
 # fname = QFileDialog.getOpenFileName(self, 'Open file',  
 # 'c:\\', "Image files (\*.jpg \*.gif)")  
 # self.le = QLabel('Hello')  
 # self.le.setPixmap(QPixmap(fname))  
 # dir = QFileDialog.getExistingDirectory(self, tr("Open Directory"),  
 # "/home",  
 # QFileDialog.ShowDirsOnly  
 # | QFileDialog.DontResolveSymlinks)* def FileDialog(directory=**''**, forOpen=True, fmt=**''**, isFolder=False):  
 options = QFileDialog.Options()  
 options |= QFileDialog.DontUseNativeDialog  
 options |= QFileDialog.DontUseCustomDirectoryIcons  
 dialog = QFileDialog()  
 dialog.setOptions(options)  
  
 dialog.setFilter(dialog.filter() | QtCore.QDir.Hidden)  
  
 *# ARE WE TALKING ABOUT FILES OR FOLDERS* if isFolder:  
 dialog.setFileMode(QFileDialog.DirectoryOnly)  
 else:  
 dialog.setFileMode(QFileDialog.AnyFile)  
 *# OPENING OR SAVING* dialog.setAcceptMode(QFileDialog.AcceptOpen) if forOpen else dialog.setAcceptMode(QFileDialog.AcceptSave)  
  
 *# SET FORMAT, IF SPECIFIED* if fmt != **''** and isFolder is False:  
 dialog.setDefaultSuffix(fmt)  
 dialog.setNameFilters([**f'**{fmt} **(\*.**{fmt}**)'**])  
  
 *# SET THE STARTING DIRECTORY* if directory != **''**:  
 dialog.setDirectory(str(directory))  
 else:  
 print(**"I'm stuck"**)  
 *#dialog.setDirectory(str(ROOT\_DIR))* if dialog.exec\_() == QDialog.Accepted:  
 path = dialog.selectedFiles()[0] *# returns a list* return path  
 else:  
 return **''** def showAssignStudent(self):  
 *#print("clicked")* self.FormAssignStudent = QtWidgets.QMainWindow()  
 self.assignStudentUI = Ui\_FormAssignStudent()  
 self.assignStudentUI.setupUi(self.FormAssignStudent)  
 self.FormAssignStudent.show()  
  
 def showAssignments(self):  
 self.FormAssignments = QtWidgets.QWidget()  
 self.AssignementsUi = Ui\_FormAssignments()  
 self.AssignementsUi.setupUi(self.FormAssignments, self.userName)  
 self.FormAssignments.show()  
  
 def showGrades(self):  
 self.FormGrades = QtWidgets.QWidget()  
 self.Gradesui = Ui\_FormGrades()  
 self.Gradesui.setupUi(self.FormGrades)  
 self.FormGrades.show()  
  
 def showAddAnnouncements(self):  
 self.FormAnnouncement = QtWidgets.QWidget()  
 self.announcmentUI = Ui\_FormAnnouncement()  
 self.announcmentUI.setupUi(self.FormAnnouncement)  
 self.FormAnnouncement.show()  
  
 def showViewAnnouncements(self):  
 self.FormAnnouncementView = QtWidgets.QWidget()  
 self.ViewAnnouncementui = Ui\_FormAnnouncementView()  
 self.ViewAnnouncementui.setupUi(self.FormAnnouncementView)  
 self.FormAnnouncementView.show()  
  
*#done*class Ui\_FormAssignments(object):  
 def setupUi(self, FormAssignments, userName):  
 self.userName = userName  
 FormAssignments.setObjectName(**"FormAssignments"**)  
 FormAssignments.resize(499, 300)  
 self.labelAssignments = QtWidgets.QLabel(FormAssignments)  
 self.labelAssignments.setGeometry(QtCore.QRect(190, 20, 131, 61))  
 font = QtGui.QFont()  
 font.setPointSize(16)  
 self.labelAssignments.setFont(font)  
 self.labelAssignments.setObjectName(**"labelAssignments"**)  
 self.comboClasses = QtWidgets.QComboBox(FormAssignments)  
 self.comboClasses.setGeometry(QtCore.QRect(170, 90, 151, 22))  
 self.comboClasses.setObjectName(**"comboClasses"**)  
 self.textAssignmentDescription = QtWidgets.QTextEdit(FormAssignments)  
 self.textAssignmentDescription.setGeometry(QtCore.QRect(170, 120, 161, 31))  
 self.textAssignmentDescription.setObjectName(**"textAssignmentDescription"**)  
 self.labelClasses = QtWidgets.QLabel(FormAssignments)  
 self.labelClasses.setGeometry(QtCore.QRect(40, 90, 71, 21))  
 font = QtGui.QFont()  
 font.setPointSize(8)  
 self.labelClasses.setFont(font)  
 self.labelClasses.setObjectName(**"labelClasses"**)  
 self.labelAssignmentDescription = QtWidgets.QLabel(FormAssignments)  
 self.labelAssignmentDescription.setGeometry(QtCore.QRect(40, 120, 121, 21))  
 font = QtGui.QFont()  
 font.setPointSize(8)  
 self.labelAssignmentDescription.setFont(font)  
 self.labelAssignmentDescription.setObjectName(**"labelAssignmentDescription"**)  
 self.labelDueDate = QtWidgets.QLabel(FormAssignments)  
 self.labelDueDate.setGeometry(QtCore.QRect(40, 160, 121, 21))  
 font = QtGui.QFont()  
 font.setPointSize(8)  
 self.labelDueDate.setFont(font)  
 self.labelDueDate.setObjectName(**"labelDueDate"**)  
 self.textDueDate = QtWidgets.QTextEdit(FormAssignments)  
 self.textDueDate.setGeometry(QtCore.QRect(170, 160, 161, 31))  
 self.textDueDate.setObjectName(**"textDueDate"**)  
 self.buttonCreateAssignment = QtWidgets.QPushButton(FormAssignments)  
 self.buttonCreateAssignment.setGeometry(QtCore.QRect(190, 200, 111, 41))  
 self.buttonCreateAssignment.setObjectName(**"buttonCreateAssignment"**)  
  
 self.buttonCreateAssignment.clicked.connect(lambda : self.createAssignment())  
  
 self.retranslateUi(FormAssignments)  
 QtCore.QMetaObject.connectSlotsByName(FormAssignments)  
  
 def retranslateUi(self, FormAssignments):  
 \_translate = QtCore.QCoreApplication.translate  
 FormAssignments.setWindowTitle(\_translate(**"FormAssignments"**, **"Form"**))  
 self.labelAssignments.setText(\_translate(**"FormAssignments"**, **"Assignments"**))  
 self.labelClasses.setText(\_translate(**"FormAssignments"**, **"Classes"**))  
 self.labelAssignmentDescription.setText(\_translate(**"FormAssignments"**, **"Assignment Description"**))  
 self.labelDueDate.setText(\_translate(**"FormAssignments"**, **"Due Date"**))  
 self.buttonCreateAssignment.setText(\_translate(**"FormAssignments"**, **"Create Assignment"**))  
  
 database = DatabaseHelper()  
  
 query = database.queryClassesForThisProfessor(self.userName)  
  
 answer = database.select(query)  
 for item in answer:  
 self.comboClasses.addItems(item)  
  
 database.close()  
  
 def createAssignment(self):  
 classData = self.comboClasses.currentText()  
 classQuery = **"SELECT classID from class Where '"** + classData + **"' = className"** print(**"ClassQuery = "** + classQuery)  
 database = DatabaseHelper()  
 classID = database.select(classQuery)  
 assignmentDescription = self.textAssignmentDescription.toPlainText()  
 dueDate = self.textDueDate.toPlainText()  
  
 queryAddAssignment = **"INSERT INTO assignments (classID, assignmentDescription, dueDate) "** \  
 **"VALUES ("** + str(classID[0][0]) + **", '"** + assignmentDescription + **"', '"** + dueDate + **"')"** print(**"Add assignment query = "** + queryAddAssignment)  
  
 database.edit(queryAddAssignment)  
 database.close()  
  
*#done*class Ui\_FormGrades(object):  
 def setupUi(self, FormGrades):  
 FormGrades.setObjectName(**"FormGrades"**)  
 FormGrades.resize(499, 300)  
 self.comboClasses = QtWidgets.QComboBox(FormGrades)  
 self.comboClasses.setGeometry(QtCore.QRect(190, 90, 151, 22))  
 self.comboClasses.setObjectName(**"comboClasses"**)  
 self.textDueDate = QtWidgets.QTextEdit(FormGrades)  
 self.textDueDate.setGeometry(QtCore.QRect(190, 180, 161, 31))  
 self.textDueDate.setObjectName(**"textDueDate"**)  
 self.buttonPostGrade = QtWidgets.QPushButton(FormGrades)  
 self.buttonPostGrade.setGeometry(QtCore.QRect(200, 230, 111, 41))  
 self.buttonPostGrade.setObjectName(**"buttonPostGrade"**)  
 self.labelGrades = QtWidgets.QLabel(FormGrades)  
 self.labelGrades.setGeometry(QtCore.QRect(210, 20, 131, 61))  
 font = QtGui.QFont()  
 font.setPointSize(16)  
 self.labelGrades.setFont(font)  
 self.labelGrades.setObjectName(**"labelGrades"**)  
 self.labelClasses = QtWidgets.QLabel(FormGrades)  
 self.labelClasses.setGeometry(QtCore.QRect(60, 90, 71, 21))  
 font = QtGui.QFont()  
 font.setPointSize(8)  
 self.labelClasses.setFont(font)  
 self.labelClasses.setObjectName(**"labelClasses"**)  
 self.labelPostGrade = QtWidgets.QLabel(FormGrades)  
 self.labelPostGrade.setGeometry(QtCore.QRect(60, 180, 121, 21))  
 font = QtGui.QFont()  
 font.setPointSize(8)  
 self.labelPostGrade.setFont(font)  
 self.labelPostGrade.setObjectName(**"labelPostGrade"**)  
 self.labelAssignments = QtWidgets.QLabel(FormGrades)  
 self.labelAssignments.setGeometry(QtCore.QRect(60, 120, 71, 21))  
 font = QtGui.QFont()  
 font.setPointSize(8)  
 self.labelAssignments.setFont(font)  
 self.labelAssignments.setObjectName(**"labelAssignments"**)  
 self.comboAssignments = QtWidgets.QComboBox(FormGrades)  
 self.comboAssignments.setGeometry(QtCore.QRect(190, 120, 151, 22))  
 self.comboAssignments.setObjectName(**"comboAssignments"**)  
 self.labelStudents = QtWidgets.QLabel(FormGrades)  
 self.labelStudents.setGeometry(QtCore.QRect(60, 150, 71, 21))  
 font = QtGui.QFont()  
 font.setPointSize(8)  
 self.labelStudents.setFont(font)  
 self.labelStudents.setObjectName(**"labelStudents"**)  
 self.comboStudents = QtWidgets.QComboBox(FormGrades)  
 self.comboStudents.setGeometry(QtCore.QRect(190, 150, 151, 22))  
 self.comboStudents.setObjectName(**"comboStudents"**)  
  
 self.comboClasses.currentIndexChanged.connect(lambda : self.comboBoxClassChanged())  
 self.buttonPostGrade.clicked.connect(lambda : self.postGrade())  
  
 self.retranslateUi(FormGrades)  
 QtCore.QMetaObject.connectSlotsByName(FormGrades)  
  
 def retranslateUi(self, FormGrades):  
 \_translate = QtCore.QCoreApplication.translate  
 FormGrades.setWindowTitle(\_translate(**"FormGrades"**, **"Form"**))  
 self.buttonPostGrade.setText(\_translate(**"FormGrades"**, **"Post Grade"**))  
 self.labelGrades.setText(\_translate(**"FormGrades"**, **"Grades"**))  
 self.labelClasses.setText(\_translate(**"FormGrades"**, **"Classes"**))  
 self.labelPostGrade.setText(\_translate(**"FormGrades"**, **"Grade"**))  
 self.labelAssignments.setText(\_translate(**"FormGrades"**, **"Assignments"**))  
 self.labelStudents.setText(\_translate(**"FormGrades"**, **"Students"**))  
  
 database = DatabaseHelper()  
  
 query = database.queryClassesForThisProfessor(userName)  
  
 answer = database.select(query)  
 for item in answer:  
 self.comboClasses.addItems(item)  
  
 database.close()  
  
 def comboBoxClassChanged(self):  
 self.comboStudents.clear()  
 self.comboAssignments.clear()  
  
 database = DatabaseHelper()  
  
 classID = self.comboClasses.currentText()  
 query = **"SELECT assignmentDescription from assignments, class "** \  
 **"WHERE class.className = '"** + classID + **"' and assignments.classID = class.classID"** answer = database.select(query)  
 for item in answer:  
 self.comboAssignments.addItems(item)  
  
 studentQuery = **"SELECT userName from accounts, classToStudent, class "** \  
 **"Where class.className = '"** + classID + **"' and "** \  
 **"accounts.accountID = classToStudent.studentID "** \  
 **"and classToStudent.classID = class.classID"** studentAnswer = database.select(studentQuery)  
 for student in studentAnswer:  
 self.comboStudents.addItems(student)  
  
 database.close()  
  
 def postGrade(self):  
 database = DatabaseHelper()  
  
 studentUserName = self.comboStudents.currentText()  
 studentIDQuery = **"SELECT accountID from accounts where userName = '"** + studentUserName + **"'"** studentID = database.select(studentIDQuery)  
  
 assignmentName = self.comboAssignments.currentText()  
 assignmentIDQuery = **"SELECT assignmentID from assignments where assignmentDescription = '"** + assignmentName + **"'"** assignmentID = database.select(assignmentIDQuery)  
  
 grade = self.textDueDate.toPlainText()  
  
 queryAddGrade = **"INSERT INTO studentsToAssignments (studentID, assignmentID,grade) "** \  
 **"VALUES ("** + str(studentID[0][0]) + **", "** + str(assignmentID[0][0]) + **", "** + grade + **")"** print(**"Add assignment query = "** + queryAddGrade)  
  
 database.edit(queryAddGrade)  
 database.close()  
  
class Ui\_FormStudentView(object):  
 def setupUi(self, FormStudentView):  
 FormStudentView.setObjectName(**"FormStudentView"**)  
 FormStudentView.resize(499, 425)  
 self.viewTableWidget = QtWidgets.QTableWidget(FormStudentView)  
 self.viewTableWidget.setGeometry(QtCore.QRect(60, 110, 401, 281))  
 self.viewTableWidget.setObjectName(**"viewTableWidget"**)  
 self.viewTableWidget.setColumnCount(3)  
 self.viewTableWidget.setRowCount(0)  
 item = QtWidgets.QTableWidgetItem()  
 self.viewTableWidget.setHorizontalHeaderItem(0, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.viewTableWidget.setHorizontalHeaderItem(1, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.viewTableWidget.setHorizontalHeaderItem(2, item)  
 self.buttonDownloadFiles = QtWidgets.QPushButton(FormStudentView)  
 self.buttonDownloadFiles.setGeometry(QtCore.QRect(130, 10, 111, 31))  
 self.buttonDownloadFiles.setObjectName(**"buttonDownloadFiles"**)  
 self.labelCourseDashboard = QtWidgets.QLabel(FormStudentView)  
 self.labelCourseDashboard.setGeometry(QtCore.QRect(160, 60, 241, 21))  
 font = QtGui.QFont()  
 font.setPointSize(16)  
 self.labelCourseDashboard.setFont(font)  
 self.labelCourseDashboard.setObjectName(**"labelCourseDashboard"**)  
 self.buttonViewAnnouncements = QtWidgets.QPushButton(FormStudentView)  
 self.buttonViewAnnouncements.setGeometry(QtCore.QRect(250, 10, 111, 31))  
 self.buttonViewAnnouncements.setObjectName(**"buttonViewAnnouncements"**)  
  
 self.retranslateUi(FormStudentView)  
  
 self.buttonDownloadFiles.clicked.connect(lambda : self.FileDialog())  
 self.buttonViewAnnouncements.clicked.connect(lambda : self.showViewAnnouncements())  
  
 QtCore.QMetaObject.connectSlotsByName(FormStudentView)  
  
 def retranslateUi(self, FormStudentView):  
 \_translate = QtCore.QCoreApplication.translate  
 FormStudentView.setWindowTitle(\_translate(**"FormStudentView"**, **"Form"**))  
 item = self.viewTableWidget.horizontalHeaderItem(0)  
 item.setText(\_translate(**"FormStudentView"**, **"Classes"**))  
 item = self.viewTableWidget.horizontalHeaderItem(1)  
 item.setText(\_translate(**"FormStudentView"**, **"Assignments"**))  
 item = self.viewTableWidget.horizontalHeaderItem(2)  
 item.setText(\_translate(**"FormStudentView"**, **"Grades"**))  
 self.buttonDownloadFiles.setText(\_translate(**"FormStudentView"**, **"Download Files"**))  
 self.labelCourseDashboard.setText(\_translate(**"FormStudentView"**, **"Course Dashboard"**))  
 self.buttonViewAnnouncements.setText(\_translate(**"FormStudentView"**, **"View Announcements"**))  
 self.loadData()  
  
 def loadData(self):  
 database = DatabaseHelper()  
 *#print("trying first query")* query = **"Select accountID from accounts where userName = '"** + userName + **"'"** userID = database.select(query)  
  
 *#print("trying second query")* queryPopulateTable = **"SELECT class.className, assignments.assignmentDescription, studentsToAssignments.grade "** \  
 **"FROM accounts "** \  
 **"LEFT join classToStudent on accounts.accountID = classToStudent.studentID "** \  
 **"LEFT join class on class.classID = classToStudent.classID "** \  
 **"LEFT join assignments on class.classID = assignments.classID "** \  
 **"LEFT join studentsToAssignments on assignments.assignmentID = studentsToAssignments.assignmentID "** \  
 **"Where accounts.accountID = "** + str(userID[0][0])  
  
 *#print(queryPopulateTable)* answerPopulateTable = database.select(queryPopulateTable)  
  
 *#print("trying to populate data")* for rowCount, answerPopulateTable in enumerate(answerPopulateTable):  
 self.viewTableWidget.insertRow(rowCount)  
 for columnNumber, data in enumerate(answerPopulateTable):  
 cell = QtWidgets.QTableWidgetItem(str(data))  
 self.viewTableWidget.setItem(rowCount, columnNumber, cell)  
 database.close()  
  
 def FileDialog(directory=**''**, forOpen=True, fmt=**''**, isFolder=False):  
 options = QFileDialog.Options()  
 options |= QFileDialog.DontUseNativeDialog  
 options |= QFileDialog.DontUseCustomDirectoryIcons  
 dialog = QFileDialog()  
 dialog.setOptions(options)  
  
 dialog.setFilter(dialog.filter() | QtCore.QDir.Hidden)  
  
 *# ARE WE TALKING ABOUT FILES OR FOLDERS* if isFolder:  
 dialog.setFileMode(QFileDialog.DirectoryOnly)  
 else:  
 dialog.setFileMode(QFileDialog.AnyFile)  
 *# OPENING OR SAVING* dialog.setAcceptMode(QFileDialog.AcceptOpen) if forOpen else dialog.setAcceptMode(QFileDialog.AcceptSave)  
  
 *# SET FORMAT, IF SPECIFIED* if fmt != **''** and isFolder is False:  
 dialog.setDefaultSuffix(fmt)  
 dialog.setNameFilters([**f'**{fmt} **(\*.**{fmt}**)'**])  
  
 *# SET THE STARTING DIRECTORY* if directory != **''**:  
 dialog.setDirectory(str(directory))  
 else:  
 print(**"I'm stuck"**)  
 *#dialog.setDirectory(str(ROOT\_DIR))* if dialog.exec\_() == QDialog.Accepted:  
 path = dialog.selectedFiles()[0] *# returns a list* return path  
 else:  
 return **''** def showViewAnnouncements(self):  
 self.FormAnnouncementView = QtWidgets.QWidget()  
 self.ViewAnnouncementui = Ui\_FormAnnouncementView()  
 self.ViewAnnouncementui.setupUi(self.FormAnnouncementView)  
 self.FormAnnouncementView.show()  
  
  
class Ui\_FormAnnouncement(object):  
 def setupUi(self, FormAnnouncement):  
 FormAnnouncement.setObjectName(**"FormAnnouncement"**)  
 FormAnnouncement.resize(384, 228)  
 self.textAnnouncment = QtWidgets.QTextEdit(FormAnnouncement)  
 self.textAnnouncment.setGeometry(QtCore.QRect(160, 120, 161, 51))  
 self.textAnnouncment.setObjectName(**"textAnnouncment"**)  
 self.buttonPostAnnouncement = QtWidgets.QPushButton(FormAnnouncement)  
 self.buttonPostAnnouncement.setGeometry(QtCore.QRect(180, 180, 111, 41))  
 self.buttonPostAnnouncement.setObjectName(**"buttonPostAnnouncement"**)  
 self.labelAnnouncement = QtWidgets.QLabel(FormAnnouncement)  
 self.labelAnnouncement.setGeometry(QtCore.QRect(30, 120, 121, 21))  
 font = QtGui.QFont()  
 font.setPointSize(8)  
 self.labelAnnouncement.setFont(font)  
 self.labelAnnouncement.setObjectName(**"labelAnnouncement"**)  
 self.labelTitle = QtWidgets.QLabel(FormAnnouncement)  
 self.labelTitle.setGeometry(QtCore.QRect(150, 10, 161, 61))  
 font = QtGui.QFont()  
 font.setPointSize(16)  
 self.labelTitle.setFont(font)  
 self.labelTitle.setObjectName(**"labelTitle"**)  
 self.comboClasses = QtWidgets.QComboBox(FormAnnouncement)  
 self.comboClasses.setGeometry(QtCore.QRect(160, 80, 151, 22))  
 self.comboClasses.setObjectName(**"comboClasses"**)  
 self.labelClasses = QtWidgets.QLabel(FormAnnouncement)  
 self.labelClasses.setGeometry(QtCore.QRect(30, 80, 71, 21))  
 font = QtGui.QFont()  
 font.setPointSize(8)  
 self.labelClasses.setFont(font)  
 self.labelClasses.setObjectName(**"labelClasses"**)  
  
 self.retranslateUi(FormAnnouncement)  
  
 self.buttonPostAnnouncement.clicked.connect(lambda : self.addAnnouncement())  
  
 QtCore.QMetaObject.connectSlotsByName(FormAnnouncement)  
  
 def retranslateUi(self, FormAnnouncement):  
 \_translate = QtCore.QCoreApplication.translate  
 FormAnnouncement.setWindowTitle(\_translate(**"FormAnnouncement"**, **"Form"**))  
 self.buttonPostAnnouncement.setText(\_translate(**"FormAnnouncement"**, **"Make Announcement"**))  
 self.labelAnnouncement.setText(\_translate(**"FormAnnouncement"**, **"Announcement"**))  
 self.labelTitle.setText(\_translate(**"FormAnnouncement"**, **"Announcements"**))  
 self.labelClasses.setText(\_translate(**"FormAnnouncement"**, **"Classes"**))  
  
 database = DatabaseHelper()  
  
 query = database.queryClassesForThisProfessor(userName)  
  
 answer = database.select(query)  
 for item in answer:  
 self.comboClasses.addItems(item)  
  
 database.close()  
  
 def addAnnouncement(self):  
 database = DatabaseHelper()  
  
 className = self.comboClasses.currentText()  
 classIDQuery = **"SELECT classID from class where className = '"** + className + **"'"** classID = database.select(classIDQuery)  
  
 announcement = self.textAnnouncment.toPlainText()  
  
 queryAddAnnouncement = **"INSERT INTO announcements (announcementDetails, classID) "** \  
 **"VALUES ('"** + announcement + **"', "** + str(classID[0][0]) + **")"** print(**"Add announcement query = "** + queryAddAnnouncement)  
  
 database.edit(queryAddAnnouncement)  
 database.close()  
  
class Ui\_FormAnnouncementView(object):  
 def setupUi(self, FormAnnouncementView):  
 FormAnnouncementView.setObjectName(**"FormAnnouncementView"**)  
 FormAnnouncementView.resize(529, 501)  
 self.viewTableWidget = QtWidgets.QTableWidget(FormAnnouncementView)  
 self.viewTableWidget.setGeometry(QtCore.QRect(40, 180, 451, 281))  
 self.viewTableWidget.setObjectName(**"viewTableWidget"**)  
 self.viewTableWidget.setColumnCount(2)  
 self.viewTableWidget.setRowCount(0)  
 item = QtWidgets.QTableWidgetItem()  
 self.viewTableWidget.setHorizontalHeaderItem(0, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.viewTableWidget.setHorizontalHeaderItem(1, item)  
 self.labelTitle = QtWidgets.QLabel(FormAnnouncementView)  
 self.labelTitle.setGeometry(QtCore.QRect(150, 120, 241, 16))  
 font = QtGui.QFont()  
 font.setPointSize(16)  
 self.labelTitle.setFont(font)  
 self.labelTitle.setObjectName(**"labelTitle"**)  
  
 self.retranslateUi(FormAnnouncementView)  
 QtCore.QMetaObject.connectSlotsByName(FormAnnouncementView)  
  
 def retranslateUi(self, FormAnnouncementView):  
 \_translate = QtCore.QCoreApplication.translate  
 FormAnnouncementView.setWindowTitle(\_translate(**"FormAnnouncementView"**, **"Form"**))  
 item = self.viewTableWidget.horizontalHeaderItem(0)  
 item.setText(\_translate(**"FormAnnouncementView"**, **"Classes"**))  
 item = self.viewTableWidget.horizontalHeaderItem(1)  
 item.setText(\_translate(**"FormAnnouncementView"**, **"Announcements"**))  
 self.labelTitle.setText(\_translate(**"FormAnnouncementView"**, **"Announcments"**))  
 print(**"Your account type is: "** + accountType[0][0])  
  
 if(accountType[0][0] == **'professor'**):  
 self.loadProfessorData()  
 elif(accountType[0][0] == **'student'**):  
 self.loadStudentData()  
  
 def loadProfessorData(self):  
 *#print("Loading data")* database = DatabaseHelper()  
 *# #print("trying first query")  
 # query = "Select accountID from accounts where userName = '" + userName + "'"  
 # userID = database.select(query)  
  
 #print("trying second query")* queryPopulateTable = **"Select class.className, announcements.announcementDetails "** \  
 **"from accounts "** \  
 **"left join class on accounts.accountID = class.professorID "** \  
 **"left join announcements on class.classID = announcements.classID "** \  
 **"Where accounts.userName = '"** + userName + **"'"** *#print(queryPopulateTable)* answerPopulateTable = database.select(queryPopulateTable)  
  
 *#print("trying to populate data")* for rowCount, answerPopulateTable in enumerate(answerPopulateTable):  
 self.viewTableWidget.insertRow(rowCount)  
 for columnNumber, data in enumerate(answerPopulateTable):  
 cell = QtWidgets.QTableWidgetItem(str(data))  
 self.viewTableWidget.setItem(rowCount, columnNumber, cell)  
 database.close()  
  
 def loadStudentData(self):  
 *#print("Loading data")* database = DatabaseHelper()  
 *# #print("trying first query")  
 # query = "Select accountID from accounts where userName = '" + userName + "'"  
 # userID = database.select(query)  
  
 #print("trying second query")* queryPopulateTable = **"Select class.className, announcements.announcementDetails "** \  
 **"from accounts "** \  
 **"left join classToStudent on accounts.accountID = classToStudent.studentID "** \  
 **"left join class on classToStudent.classID = class.classID "** \  
 **"left join announcements on class.classID = announcements.classID "** \  
 **"Where accounts.userName = '"** + userName + **"'"** *#print(queryPopulateTable)* answerPopulateTable = database.select(queryPopulateTable)  
  
 *#print("trying to populate data")* for rowCount, answerPopulateTable in enumerate(answerPopulateTable):  
 self.viewTableWidget.insertRow(rowCount)  
 for columnNumber, data in enumerate(answerPopulateTable):  
 cell = QtWidgets.QTableWidgetItem(str(data))  
 self.viewTableWidget.setItem(rowCount, columnNumber, cell)  
 database.close()  
  
class Database():  
 def \_\_init\_\_(self):  
 self.connect = sqlite3.connect(**'student.db'**)  
 self.cursor = self.connect.cursor()  
 try:  
 self.cursor.execute(**"""CREATE TABLE student (  
 studentID INTEGER NOT NULL PRIMARY KEY,  
 userName text,  
 Password text)"""**)  
 self.connect.commit()  
 except:  
 **"do nothing"** try:  
 self.cursor.execute(**"""CREATE TABLE professor (  
 professorID INTEGER NOT NULL PRIMARY KEY,  
 userName text,  
 Password text)"""**)  
 self.connect.commit()  
 except:  
 **"do nothing"** try:  
 self.cursor.execute(**"""CREATE TABLE admin (  
 adminID INTEGER NOT NULL PRIMARY KEY,  
 userName text,  
 Password text)"""**)  
 self.connect.commit()  
 except:  
 **"do nothing"** try:  
 self.cursor.execute(**"""CREATE TABLE class (  
 classID INTEGER NOT NULL PRIMARY KEY,  
 className text,  
 professorID INTERGER)"""**)  
 self.connect.commit()  
 except:  
 **"do nothing"** try:  
 self.cursor.execute(**"""CREATE TABLE classToStudent (  
 classID INTEGER NOT NULL PRIMARY KEY,  
 studentID INTERGER)"""**)  
 self.connect.commit()  
 except:  
 **"do nothing"** self.fetchAll()  
 self.connect.close()  
  
 def deleteItem(self):  
 *#print("Trying to delete baby")* self.selectedItem = self.listbox.get(**'active'**)  
 *#print(self.selectedItem)  
 #print(self.selectedItem[0])* self.cursor.execute(**"DELETE FROM student WHERE first = ? AND last = ?"**,  
 (self.selectedItem[0], self.selectedItem[1]))  
 self.connect.commit()  
 self.listbox.delete(**'active'**)  
 self.fetchAll()  
  
 def fetchAll(self):  
 self.listbox.delete(0, **'end'**)  
 self.cursor.execute(**"SELECT \* FROM STUDENT"**)  
 self.currentSearch = self.cursor.fetchall()  
 for item in self.currentSearch:  
 self.listbox.insert(**'end'**, item)  
  
 def submit(self):  
 self.student = Student(self.firstNameEntry.get(), self.lastNameEntry.get(), self.streetEntry.get(), self.cityEntry.get(),  
 self.stateEntry.get(), self.emailEntry.get(), self.telephoneEntry.get())  
  
 self.cursor.execute(**"INSERT INTO student VALUES (?,?,?,?,?,?,?)"**, (self.firstNameEntry.get(),  
 self.lastNameEntry.get(),  
 self.streetEntry.get(),  
 self.cityEntry.get(),  
 self.stateEntry.get(),  
 self.emailEntry.get(),  
 self.telephoneEntry.get()))  
  
 **"""  
 self.cursor.execute("INSERT INTO student VALUES ('Bobby', 'Fisher', 'Bayberry lane', 'Bridgeport', 'CT', "  
 "'bobbyFisher@bridgeport.edu', '8675309')")"""** self.connect.commit()  
 self.fetchAll()  
  
class DatabaseHelper():  
 def \_\_init\_\_(self, name=**'projectDatabase.db'**):  
 self.connect = None  
 self.cursor = None  
  
 if name:  
 self.open(name)  
  
 def open(self, name=**'projectDatabase.db'**):  
 try:  
 self.connect = sqlite3.connect(name)  
 self.cursor = self.connect.cursor()  
 print(sqlite3.version)  
 except sqlite3.Error as e:  
 print(**"Failed to connect to database"**)  
  
 def CreateTables(self):  
  
 try:  
 self.cursor.execute(**"""CREATE TABLE accounts (  
 accountID INTEGER NOT NULL PRIMARY KEY,  
 userName text,  
 Password text,  
 accountType text)"""**)  
 self.connect.commit()  
  
 self.cursor.execute(**"INSERT INTO accounts (userName, Password, accountType) VALUES ('admin', 'password', 'admin')"**)  
 self.connect.commit()  
  
 except:  
 **"do nothing"** *# try:  
 # self.cursor.execute("""CREATE TABLE student (  
 # studentID INTEGER NOT NULL PRIMARY KEY,  
 # userName text,  
 # Password text)""")  
 # self.connect.commit()  
 # except:  
 # "do nothing"  
 # try:  
 # self.cursor.execute("""CREATE TABLE professor (  
 # professorID INTEGER NOT NULL PRIMARY KEY,  
 # userName text,  
 # Password text)""")  
 # self.connect.commit()  
 # except:  
 # "do nothing"  
 #  
 # try:  
 # self.cursor.execute("""CREATE TABLE admin (  
 # adminID INTEGER NOT NULL PRIMARY KEY,  
 # userName text,  
 # Password text)""")  
 # self.connect.commit()  
 # except:  
 # "do nothing"* try:  
 self.cursor.execute(**"""CREATE TABLE class (  
 classID INTEGER NOT NULL PRIMARY KEY,  
 className text,  
 professorID INTERGER)"""**)  
 self.connect.commit()  
 except:  
 **"do nothing"** try:  
 self.cursor.execute(**"""CREATE TABLE classToStudent (  
 classID INTEGER NOT NULL PRIMARY KEY,  
 studentID INTERGER)"""**)  
 self.connect.commit()  
 except:  
 **"do nothing"** *# try:  
 # self.cursor.execute("""CREATE TABLE accountType (  
 # accountTypeID INTEGER NOT NULL PRIMARY KEY,  
 # accountType text)""")  
 # self.connect.commit()  
 # except:  
 # "do nothing"* def edit(self, query): *#insert & update* try:  
 c = self.cursor  
 c.execute(query)  
 self.connect.commit()  
 except :  
 print(**"something is wrong with the query"**)  
 print(query)  
  
 def select(self, query):*#select* try:  
 c = self.cursor  
 c.execute(query)  
 return c.fetchall()  
 except:  
 print(query + **" is a bad query"**)  
  
 def queryClassesForThisProfessor(self, userName):  
 return **"SELECT class.className from class, accounts "** \  
 **"where accounts.userName = '"** + userName + **"' and class.professorID = accounts.accountID"** def close(self):  
 self.connect.close()  
  
if \_\_name\_\_ == **"\_\_main\_\_"**:  
 import sys  
 app = QtWidgets.QApplication(sys.argv)  
 MainWindow = QtWidgets.QMainWindow()  
 ui = Ui\_MainWindow()  
 ui.setupUi(MainWindow)  
 MainWindow.show()  
 sys.exit(app.exec\_())