ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ

**Государственное бюджетное профессиональное образовательное учреждение города Москвы**

**«Московский государственный колледж электромеханики и**

**информационных технологий»**

**(ГБПОУ МГКЭИТ)**

**ОТЧЕТ ПО ПРАКТИКЕ**

**Специальность** 09.02.07 Информационные системы и программирование

**Вид практики** Производственная по ПМ.02 (ПП.02)

**Сроки прохождения практики** с 15.12.2021 г. по 28.12.2021 г.

**Место прохождения практики**: АО НИПЦ ГНТ

*(название предприятия)*

Выполнил студент **3** курса группы **3ИП11-19**

**\_\_\_\_\_\_Петряев Игорь Николаевич\_\_\_\_\_\_\_\_\_\_\_\_**

*(Ф.И.О. студента)*

Москва, 2021

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc91451886)

[ОСНОВНАЯ ЧАСТЬ 4](#_Toc91451887)

[1 Анализ предметной области 4](#_Toc91451888)

[2 Построение концептуальной модели 5](#_Toc91451889)

[3 Требования к ПО 6](#_Toc91451890)

[4 Генерация кода проектируемого ПО 6](#_Toc91451891)

[5 Программный продукт 7](#_Toc91451892)

[ЗАКЛЮЧЕНИЕ 9](#_Toc91451893)

[СПИСОК ЛИТЕРАТУРЫ 10](#_Toc91451894)

[ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ОБЪЕКТОВ 11](#_Toc91451895)

ВВЕДЕНИЕ

Я проходил учебную практику в Учебном Центре «Академия супервайзинга бурения и нефтегазодобычи» АО «НИПЦ ГНТ» с 15 декабря 2021 г. по 28 декабря 2021 г. Руководителем практики от предприятия была назначена Иванченко Алёна Игоревна*.*

Цель производственной практики — комплексное освоение всех видов профессиональной деятельности по специальности 09.02.07 «Информационные системы и программирование», формирование общих и профессиональных компетенций, а также приобретение необходимых умений и опыта практической работы по специальности.

Задачи производственной практики (освоение общих и профессиональных компетенций):

1) выбирать способы решения задач профессиональной деятельности,

2) осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности,

3) планировать и реализовывать собственное профессиональное и личностное развитие,

4) работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами,

5) осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста,

6) использовать информационные технологии в профессиональной деятельности,

7) приобрести навыки разработки модулей программного обеспечения для компьютерных систем.

На период практики от руководителя практики от ГБПОУ МГКЭИТ было получено следующее задание: разработка БД и приложения планирования строительства скважин на шельфе.

ОСНОВНАЯ ЧАСТЬ

**1 Анализ предметной области**

АО «НИПЦ ГНТ» является научно-исследовательским и проектным центром газонефтяных технологий, созданный в 2001 году Российским государственным университетом нефти и газа им. И.М. Губкина для решения инновационных задач нефтегазовой отрасли и нефтегазового высшего образования.

В настоящее время организация занимается технико-технологическим контролем (супервайзингом) за качеством строительства и восстановления скважин, а также ведет разработку проектно-сметной документации на строительство скважин на месторождениях ОOО «Лукойл-КОМИ», ОOО «Лукойл-Западная Сибирь», ОOО «Лукойл-Пермь», ОАО «РИТЭК», ОАО «НГК «Славнефть», ОАО «Газпром нефть», ОАО «НК «Роснефть», ОАО «Татнефть», ОАО «ННК».

Конечной целью АО «НИПЦ ГНТ» является повешение эффективности инвестиций в строительство и восстановление скважин путем высокой организации труда и качественном контроле производственных процессов.

Структура подразделения в отделы организации рисунки 1 и 2:



Рисунок 1 — cтруктура организации

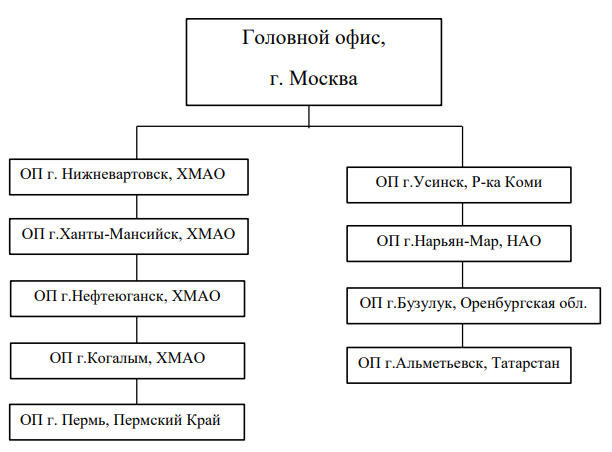


Рисунок 2 — cтруктура организации

Согласно теме исследования основными понятиями являются:

1) Скважина — горная выработка круглого сечения, пробуренная с поверхности земли или с подземной выработки без доступа человека к забою под любым углом к горизонту, диаметр которой много меньше ее глубины.

2) Шельф — относительно мелководные (до нескольких сотен метров) участки дна океанов, окраинных и внутренних морей, окаймляющие континенты и острова.

2 Построение концептуальной модели

Основные сущности и их описания отображены на рисунке 3.

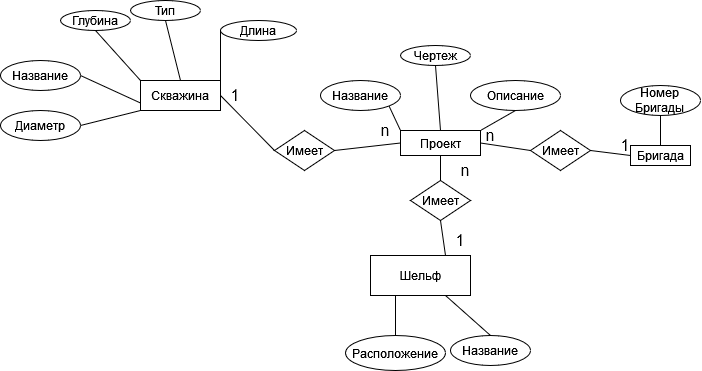


Рисунок 3 — концептуальная модель

3 Требования к ПО

Основные требования для работы с программным продуктом являются:

— процессор с тактовой частотой 1200 MHz или более мощный,

— оперативная память 256 Мб или больше,

— свободное место на жёстком диске от 540 Мб,

— архитектура с разрядностью 32 бит или 64 бит (x86 или x64),

— операционная система Windows XP, Windows Vista, Windows 7, Windows 8.

4 Генерация кода проектируемого ПО

Для данного программного продукта была выбрана СУБД SQlite. А для создания интерфейса был использован PyQt5.

Примеры генерации кода отображены на рисунке 4 и на рисунке 5. Основной код отображен в приложении А в таблице А3, А4.

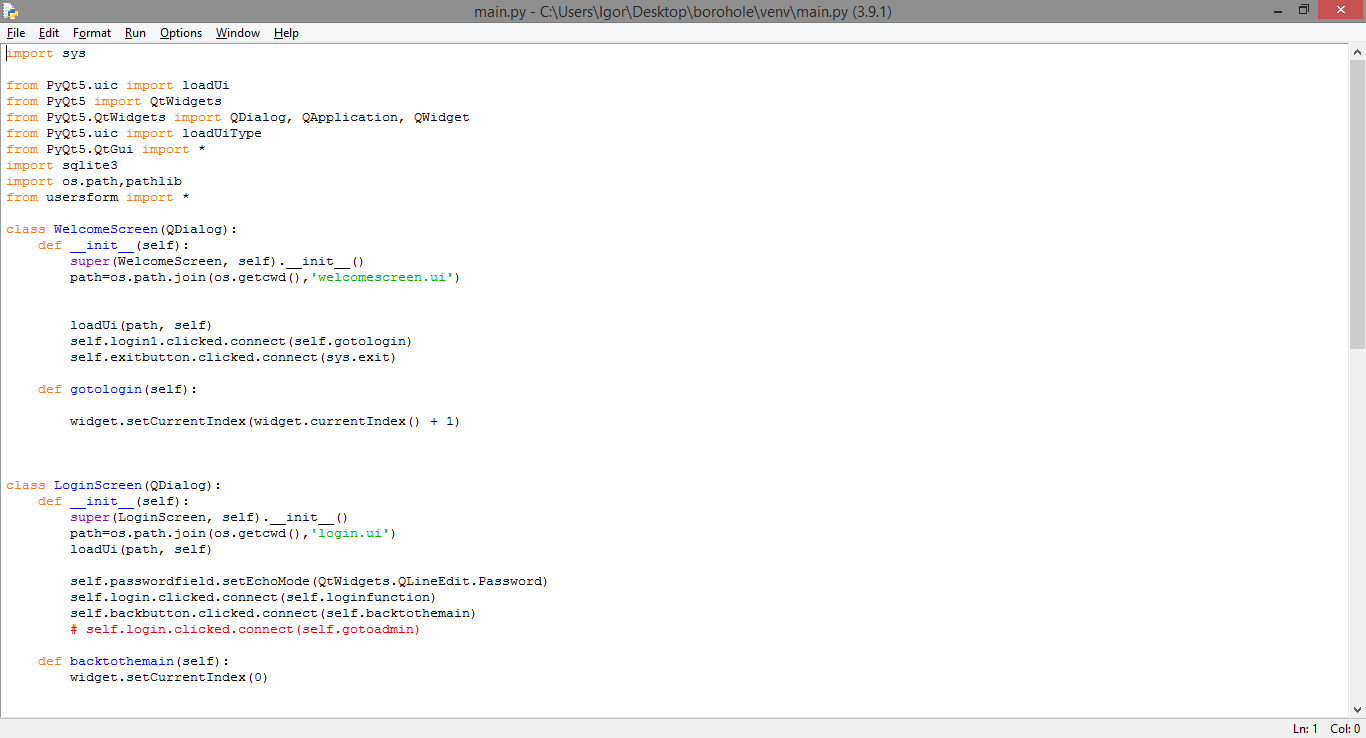


Рисунок 4 — пример генерации кода.

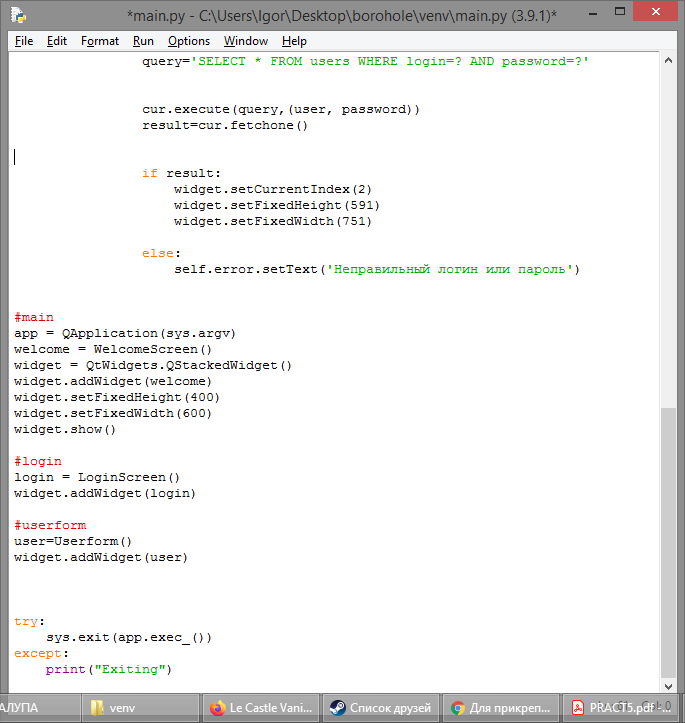


Рисунок 5 — пример генерации кода.

5 Программный продукт

Разрабатываемый программный продукт будет содержать окно авторизации и взаимодействие с базой данной. Пример продукта отображен на рисунке 6, 7.

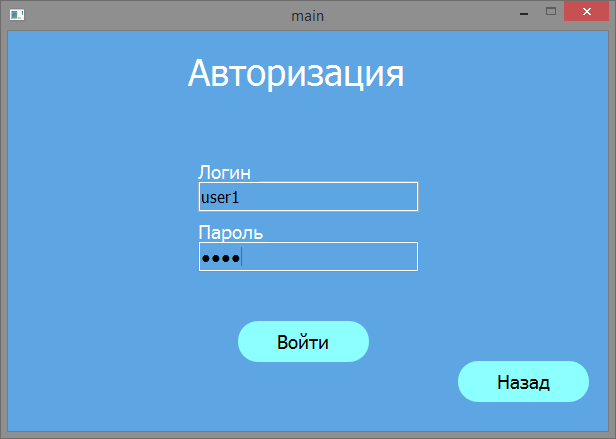


Рисунок 6 — окно авторизации

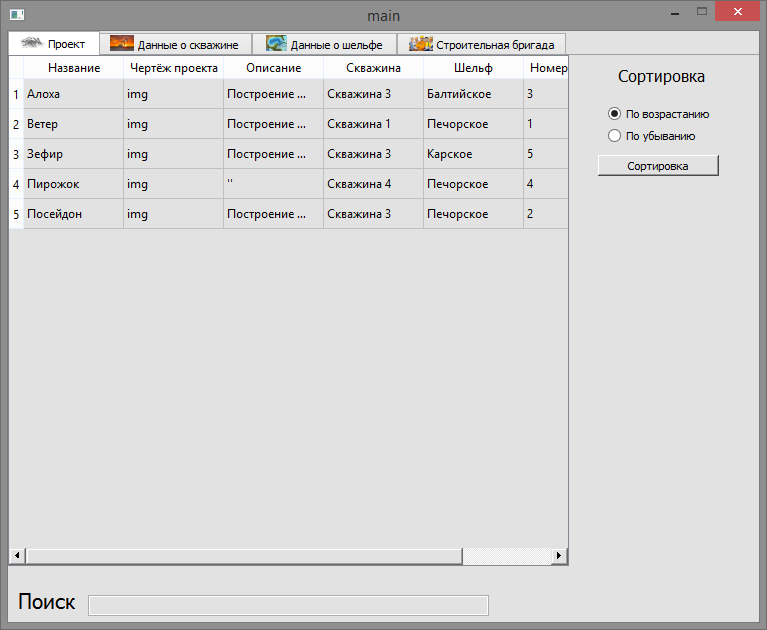


Рисунок 7 — взаимодействие с базой данной

ЗАКЛЮЧЕНИЕ

В результате прохождения учебной практики в «Научно-исследовательский и проектный центр газонефтяных технологий» АО «НИПЦ ГНТ»я приобрел практические навыки по специальности «Информационные системы и программирование». Были выполнены все поставленные задачи.

СПИСОК ЛИТЕРАТУРЫ

1) Научно – исследовательский и проектный центр газонефтяных технологий [Электронный ресурс]. — URL: <https://gasoilcenter.ru/> (дата обращения: 17.12.2021).

2) Нефтяная скважина, виды, устройство, строительство и этапы разработки [Электронный ресурс]. — URL: <https://втораяиндустриализация.рф/neftyanaya-skvazhina-vidyi-ustroystvo-stroitelstvo-i-etapyi-razrabotki/> (дата обращения: 18.12.2021).

3) Шельф [Электронный ресурс]. — URL: <https://gkd.ru/444675a-shelf---eto-ponyatie-opredelenie-granitsa-shelfa-glubina-struktura-i-razrabotki> (дата обращения: 21.12.2021).

4) Draw io. [Электронный ресурс]. – URL: <https://app.diagrams.net/> (Дата обращения: 20.12.2021).

5) Qt for Python [Электронный ресурс]. — URL: <https://doc.qt.io/qtforpython/> (дата обращения: 25.12.2021).

6) SQlite documentation [Электронный ресурс]. — URL: <https://www.sqlite.org/docs.html> (дата обращения: 25.12.2021).

ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ОБЪЕКТОВ

Таблица А.1 – Исходный код создания таблиц

|  |
| --- |
| **borehole.sql** |
| CREATE TABLE "borehole" (  "id" INTEGER NOT NULL,  "name" TEXT NOT NULL,  "type\_borehole" INTEGER NOT NULL,  "diameter" REAL NOT NULL,  "lenght" REAL NOT NULL,  "depth" REAL NOT NULL,  PRIMARY KEY("id" AUTOINCREMENT)  );  CREATE TABLE "construction\_team" (  "id" INTEGER NOT NULL,  "number\_of\_team" TEXT NOT NULL,  PRIMARY KEY("id" AUTOINCREMENT)  );  CREATE TABLE "project" (  "id" INTEGER NOT NULL,  "name\_project" TEXT NOT NULL,  "plan\_project" TEXT,  "description" TEXT,  "id\_team" INTEGER NOT NULL,  "id\_borehole" INTEGER NOT NULL,  "id\_shelf" INTEGER NOT NULL,  PRIMARY KEY("id" AUTOINCREMENT),  FOREIGN KEY("id\_team") REFERENCES "construction\_team"("id"),  FOREIGN KEY("id\_borehole") REFERENCES "borehole"("id")  );  CREATE TABLE "shelf" (  "id" INTEGER NOT NULL,  "name" TEXT NOT NULL,  "location" TEXT NOT NULL,  PRIMARY KEY("id" AUTOINCREMENT)  );  CREATE TABLE "users" (  "login" TEXT,  "password" TEXT  ); |

Таблица А.2 – Исходный код создания представления

|  |
| --- |
| **borehole.sql** |
| CREATE VIEW 'BoreholeView' AS  SELECT name, type\_borehole, diameter, lenght, depth  FROM borehole;  CREATE VIEW 'ProjectView' AS  SELECT p.name\_project, p.plan\_project, p.description, b.name AS borehole,  sh.name AS shelf, ct.number\_of\_team FROM project AS p  JOIN borehole AS b ON b.id=p.id\_borehole  JOIN shelf AS sh ON sh.id=p.id\_shelf  JOIN construction\_team AS ct ON p.id\_team=ct.id;  CREATE VIEW 'ShelfView' AS  SELECT name, location FROM shelf;  CREATE VIEW 'TeamView' AS  SELECT number\_of\_team FROM construction\_team; |

Таблица А.3 – Исходный код создания окна авторизации

|  |
| --- |
| **main.py** |
| import sys  from PyQt5.uic import loadUi  from PyQt5 import QtWidgets  from PyQt5.QtWidgets import QDialog, QApplication, QWidget  from PyQt5.uic import loadUiType  from PyQt5.QtGui import \*  import sqlite3  import os.path,pathlib  from usersform import \*  class WelcomeScreen(QDialog):  def \_\_init\_\_(self):  super(WelcomeScreen, self).\_\_init\_\_()  path=os.path.join(os.getcwd(),'welcomescreen.ui')      loadUi(path, self)  self.login1.clicked.connect(self.gotologin)  self.exitbutton.clicked.connect(sys.exit)  def gotologin(self):  widget.setCurrentIndex(widget.currentIndex() + 1)  class LoginScreen(QDialog):  def \_\_init\_\_(self):  super(LoginScreen, self).\_\_init\_\_()  path=os.path.join(os.getcwd(),'login.ui')  loadUi(path, self)  self.passwordfield.setEchoMode(QtWidgets.QLineEdit.Password)  self.login.clicked.connect(self.loginfunction)  self.backbutton.clicked.connect(self.backtothemain)  # self.login.clicked.connect(self.gotoadmin)  def backtothemain(self):  widget.setCurrentIndex(0)  def loginfunction(self):  user = self.emailfield.text()  password = self.passwordfield.text()  if len(user) == 0 or len(password) == 0:  self.error.setText("Пожалуйста заполните все поля")  else:  path=os.path.join(os.getcwd(),'borehole.db')  with sqlite3.connect(path) as conn:  cur = conn.cursor()  query='SELECT \* FROM users WHERE login=? AND password=?'    cur.execute(query,(user, password))  result=cur.fetchone()    if result:  widget.setCurrentIndex(2)  widget.setFixedHeight(591)  widget.setFixedWidth(751)  else:  self.error.setText('Неправильный логин или пароль')  #main  app = QApplication(sys.argv)  welcome = WelcomeScreen()  widget = QtWidgets.QStackedWidget()  widget.addWidget(welcome)  widget.setFixedHeight(400)  widget.setFixedWidth(600)  widget.show()  #login  login = LoginScreen()  widget.addWidget(login)  #userform  user=Userform()  widget.addWidget(user)  try:  sys.exit(app.exec\_())  except:  print("Exiting") |

Таблица А.4 – Исходный код пользовательского окна

|  |
| --- |
| **userform.py** |
| import sys  from PyQt5.uic import loadUi  from PyQt5 import QtWidgets  from PyQt5.QtWidgets import QDialog, QApplication, QWidget  from PyQt5.uic import loadUiType  import sqlite3  import os.path  class Userform(QDialog):  def \_\_init\_\_(self):  super(Userform, self).\_\_init\_\_()  path=os.path.join(os.getcwd(),'usersform.ui')  loadUi(path,self)  self.openBD()  self.load\_project()  self.load\_team()  self. load\_borehole()  self.load\_shelf()  def buttons(self):  self.pushButton\_9.clicked.connect(self.printB)  def printB(self):  self.temp=True  def openBD(self):  path=os.path.join(os.getcwd(),'borehole.db')  with sqlite3.connect(path) as self.connection:  self.cur=self.connection.cursor()  def load\_project(self):  if self.radioButton\_18.isChecked():  self.sqlquery='SELECT \* FROM ProjectView ORDER BY name\_project'  elif radioButton\_17.isChecked():  self.sqlquery = 'SELECT \* FROM ProjectView ORDER BY name\_project DESC'  self.tableWidget\_project.setRowCount(5)  #print(self.pushButton\_9.clicked.connect())  tablerow = 0  for row in self.cur.execute(self.sqlquery):  self.tableWidget\_project.setItem(tablerow, 0, QtWidgets.QTableWidgetItem(str(row[0])))  self.tableWidget\_project.setItem(tablerow, 1, QtWidgets.QTableWidgetItem(str(row[1])))  self.tableWidget\_project.setItem(tablerow, 2, QtWidgets.QTableWidgetItem(str(row[2])))  self.tableWidget\_project.setItem(tablerow, 3, QtWidgets.QTableWidgetItem(str(row[3])))  self.tableWidget\_project.setItem(tablerow, 4, QtWidgets.QTableWidgetItem(str(row[4])))  self.tableWidget\_project.setItem(tablerow, 5, QtWidgets.QTableWidgetItem(str(row[5])))  tablerow+=1  def load\_team(self):  self.sqlquery='SELECT \* FROM TeamView'  self.tableWidget\_team.setRowCount(5)  tablerow=0  for row in self.cur.execute(self.sqlquery):  self.tableWidget\_team.setItem(tablerow, 0, QtWidgets.QTableWidgetItem(str(row[0])))  tablerow+=1  def load\_borehole(self):  self.sqlquery='SELECT \* FROM BoreholeView'  self.tableWidget\_borehole.setRowCount(4)  tablerow=0  for row in self.cur.execute(self.sqlquery):  self.tableWidget\_borehole.setItem(tablerow, 0, QtWidgets.QTableWidgetItem(str(row[0])))  self.tableWidget\_borehole.setItem(tablerow, 1, QtWidgets.QTableWidgetItem(str(row[1])))  self.tableWidget\_borehole.setItem(tablerow, 2, QtWidgets.QTableWidgetItem(str(row[2])))  self.tableWidget\_borehole.setItem(tablerow, 3, QtWidgets.QTableWidgetItem(str(row[3])))  self.tableWidget\_borehole.setItem(tablerow, 4, QtWidgets.QTableWidgetItem(str(row[4])))  tablerow+=1  def load\_shelf(self):  self.sqlquery='SELECT \* FROM ShelfView'  self.tableWidget\_shelf.setRowCount(2)  tablerow=0  for row in self.cur.execute(self.sqlquery):  self.tableWidget\_shelf.setItem(tablerow, 0, QtWidgets.QTableWidgetItem(str(row[0])))  self.tableWidget\_shelf.setItem(tablerow, 1, QtWidgets.QTableWidgetItem(str(row[1])))  tablerow+=1  #userform  if \_\_name\_\_=='\_\_main\_\_':  app = QApplication(sys.argv)  user=Userform()  widget = QtWidgets.QStackedWidget()  widget.addWidget(user)  widget.setCurrentIndex(2)  widget.setFixedHeight(591)  widget.setFixedWidth(751)  widget.show()  try:  sys.exit(app.exec\_())  except:  print("Exiting") |