**REPORT WEEK 2**

1. **Nhiệm vụ:**

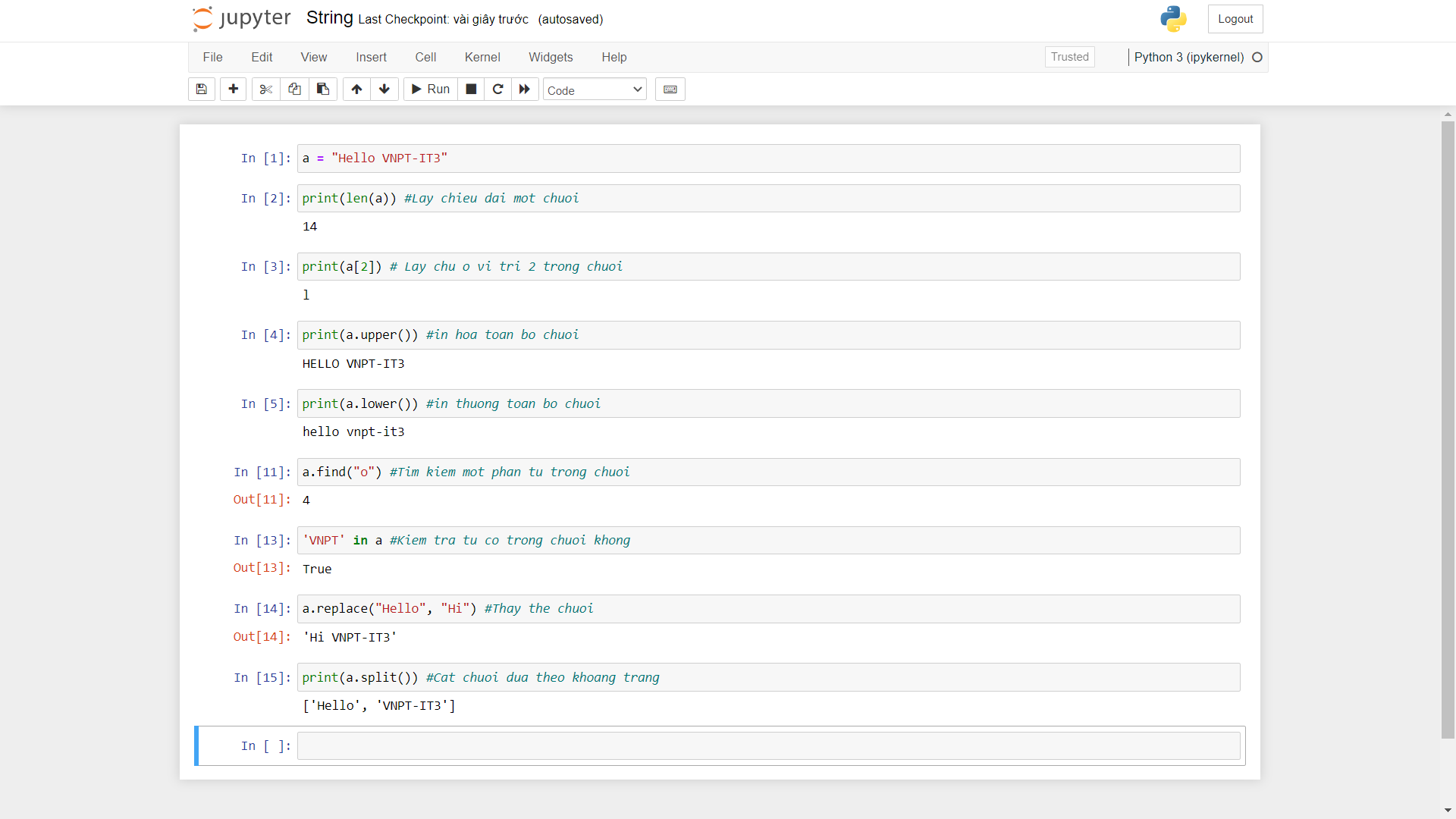
* Tìm hiểu về ngôn ngữ python.
* Tìm hiểu về môi trường ảo pipenv, thao tác, cài thư viện.
* Tìm hiểu và so sánh thao tác giữa python với Sqlite, NoSQL, SQL.
* Tìm hiểu về git.
* Tìm hiểu về telegram BotFather.

1. **Sinh viên thực hiện:**

* Tên: Đặng Ngọc Quốc Bảo
* Trường Đại học Bách Khoa Đà Nẵng
* TelegramID: quocbaodang

1. **Kết quả đạt được:**
2. **Tìm hiểu về ngôn ngữ Python:**

* Biết được các khái niệm cơ bản trong lập trình python.
* Các biểu thức trong lập trình python.
* Đối tượng string trong python. Một số lệnh thường dùng:



* Đối tượng list, tuples, Sets, Dictionary.

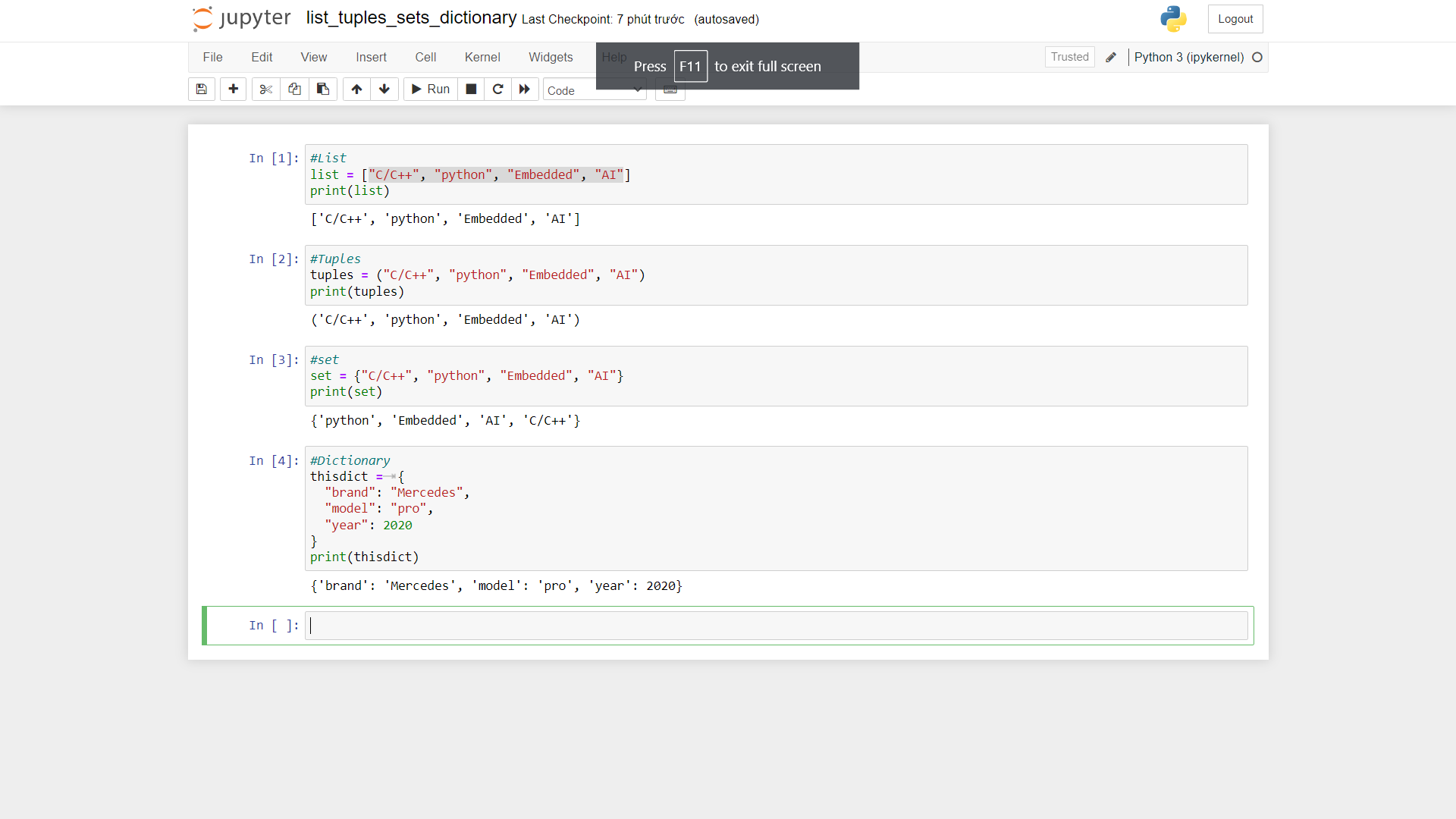
**+ List**: là một collection có thứ tự, có thể thay đổi. Cho phép chứa dữ liệu trùng lặp.

**+ Tuple**: là một collection có thứ tự, không thể thay đổi. Cho phép chứa dữ liệu trùng lặp.

**+ Set**: là một collection không có thứ tự, không có chỉ mục. Không cho phép chứa dữ liệu trùng lặp.

**+ Dictionary**: là một collection không có thứ tự, có thể thay đổi và lập chỉ mục. Không cho phép chứa dữ liệu trùng lặp.

**Các khai báo**:



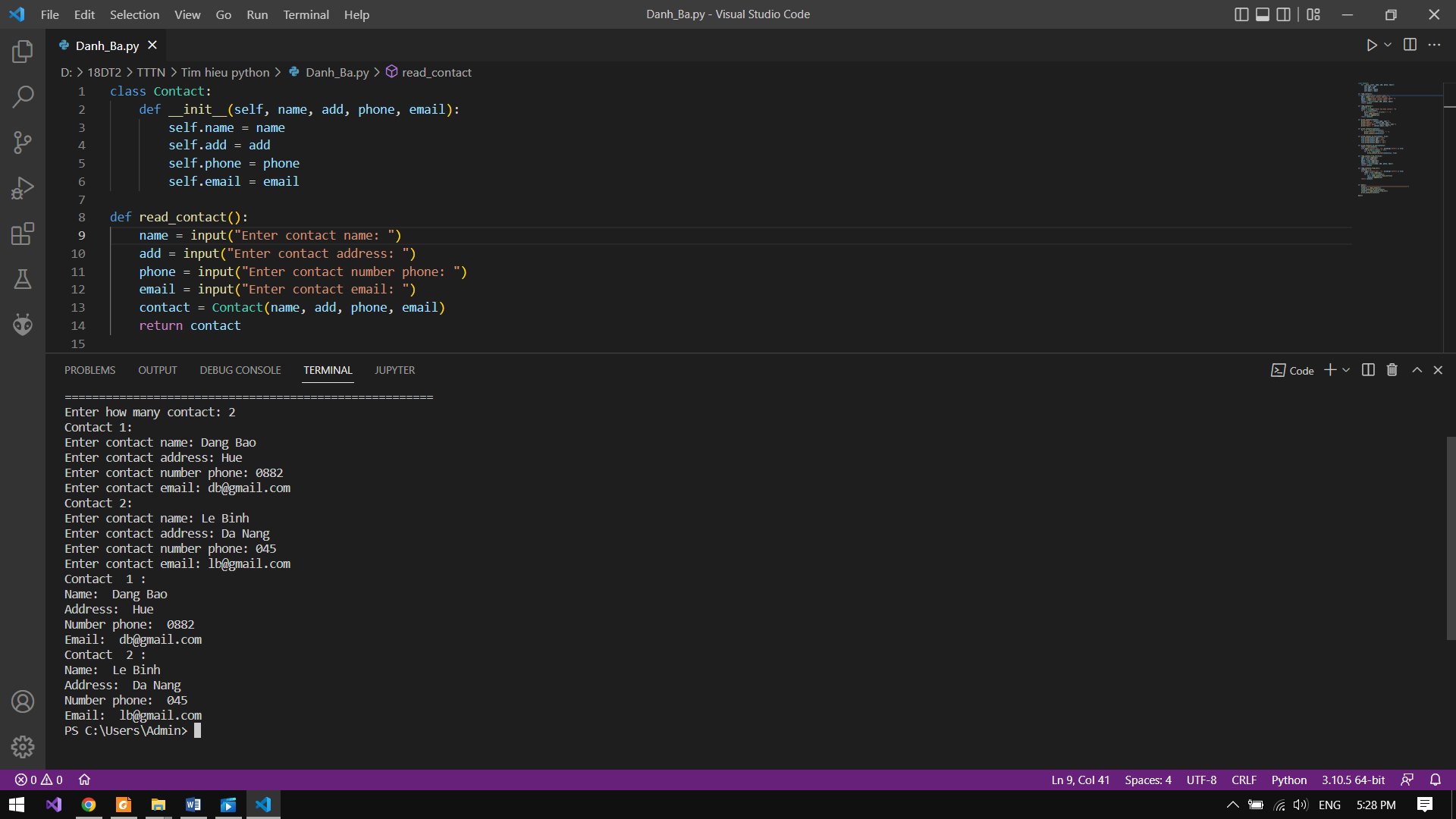
* Thao tác với hàm tự định nghĩa

Bài tập kiểm tra số hoàn thiện (là số có tổng ước bằng chính nó)

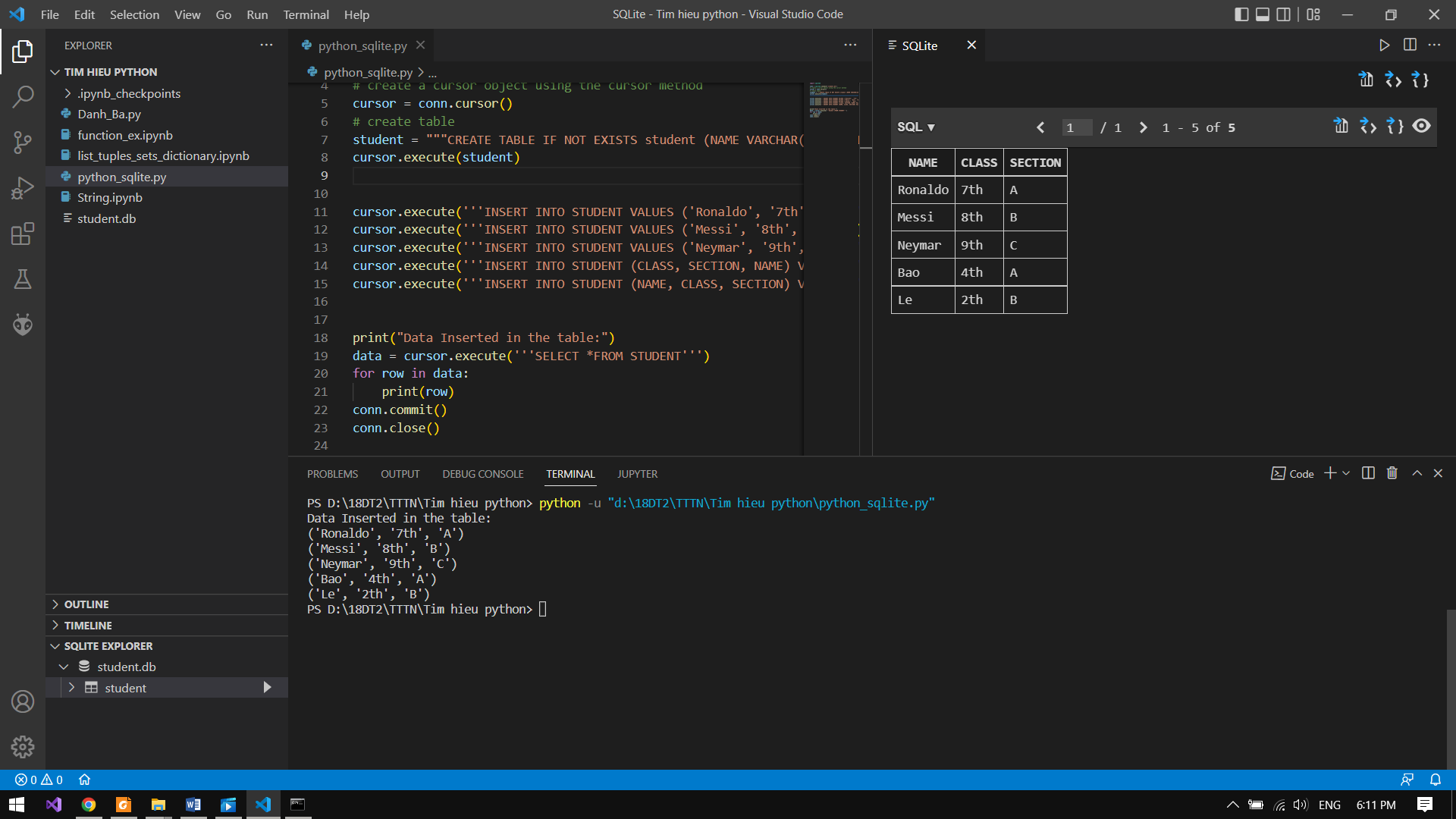


* Thao tác với file trong python.

Bài tập thêm, xem danh sách danh bạ



* Làm việc với SQLite



1. **Tìm hiểu về môi trường ảo pipenv, thao tác, cài thư viện:**

* Môi trường ảo là một công cụ để duy trì không gian riêng biệt cho một dự án với các phụ thuộc và các thư viện của nó ở một nơi. Môi trường này thì riêng biệt cho một dự án cụ thể và không ảnh hưởng đến các phụ thuộc của các dự án khác.
* Pipenv là công cụ quản lý gói. Nó kết hợp chức năng của Pip và Virtualenv, cùng với các tính năng tốt nhất của các công cụ đóng gói từ các ngôn ngữ. Điều này dẫn đến một quy trình công việc đơn giản hóa để cài đặt các gói và quản lý môi trường ảo. Pipenv chính thức được đề nghị để quản lý các phụ thuộc dự án.
* Các vấn đề mà pipenv có thể giải quyết:

+ Quản lý tệp requirements.txt có thể có vấn đề, vì vậy Pipenv sử dụng Pipfile và Pipfile.lock để tách các khai báo phụ thuộc trừu tượng khỏi kết hợp được thử nghiệm cuối cùng.

+ Sử dụng các phiên bản phụ thuộc mới nhất để giảm thiểu rủi ro bảo mật phát sinh từ các thành phần lỗi thời.

* Các thao tác làm việc với pipenv:

+ Cài đặt:

pip install pipenv

+ Tạo một shell trong môi trường ảo:

pipenv shell

* Tạo ra môi trường ảo nếu chưa tồn tại

+ Cài đặt các gói thư viện:

pipenv install flask version==0.12.1

pipenv install numpy

* Hai tệp Pipfile và Pipfile.lock được tạo ra

+ Chạy chương trình trên môi trường ảo:

pipenv run python main.py

+ Giả sử đã hoạt động tốt trong máy của mình, để đưa vào môi trường của máy khác, cần phải khóa môi trường của mình:

pipenv lock

+ Cài đặt môi trường khi nhận được file Pipfile.lock

Pipenv install –ignore-pipfile

* Pipenv sẽ tạo ra môi trường giống hệt môi trường cũ.

1. **Tìm hiểu và so sánh thao tác giữa python với Sqlite, NoSQL, SQL:**

* Python với SQLite:

+ SQLite là một database SQL dựa trên file độc lập. SQLite đi kèm với Python và được dùng trong ứng dụng Python mà không cần cài đặt phần mềm bổ sung nào khác.

+ Thao tác kết nối:

import sqlite3  # thư viện có sẵn trong python 3

conn = sqlite3.connect('student.db') #trả về đối tượng conn dùng để tương tác với database SQLite student.db

cursor = conn.cursor() # trả về đối tượng cursor

# Tạo bảng

student = """CREATE TABLE IF NOT EXISTS student (NAME VARCHAR(255), CLASS VARCHAR(255), SECTION VARCHAR(255));"""

cursor.execute(student) #Gửi các câu lệnh SQL đến database SQL

# Thêm dữ liệu vào bảng

cursor.execute('''INSERT INTO STUDENT VALUES ('Ronaldo', '7th', 'A')''')

cursor.execute('''INSERT INTO STUDENT VALUES ('Messi', '8th', 'B')''')

cursor.execute('''INSERT INTO STUDENT VALUES ('Neymar', '9th', 'C')''')

cursor.execute('''INSERT INTO STUDENT (CLASS, SECTION, NAME) VALUES ('4th', 'A', 'Bao')''')

cursor.execute('''INSERT INTO STUDENT (NAME, CLASS, SECTION) VALUES ('Le', '2th', 'B')''')

print("Data Inserted in the table:")

# Đọc dữ liệu từ bảng

data = cursor.execute('''SELECT \*FROM STUDENT''')

for row in data:

    print(row)

conn.commit()

conn.close()

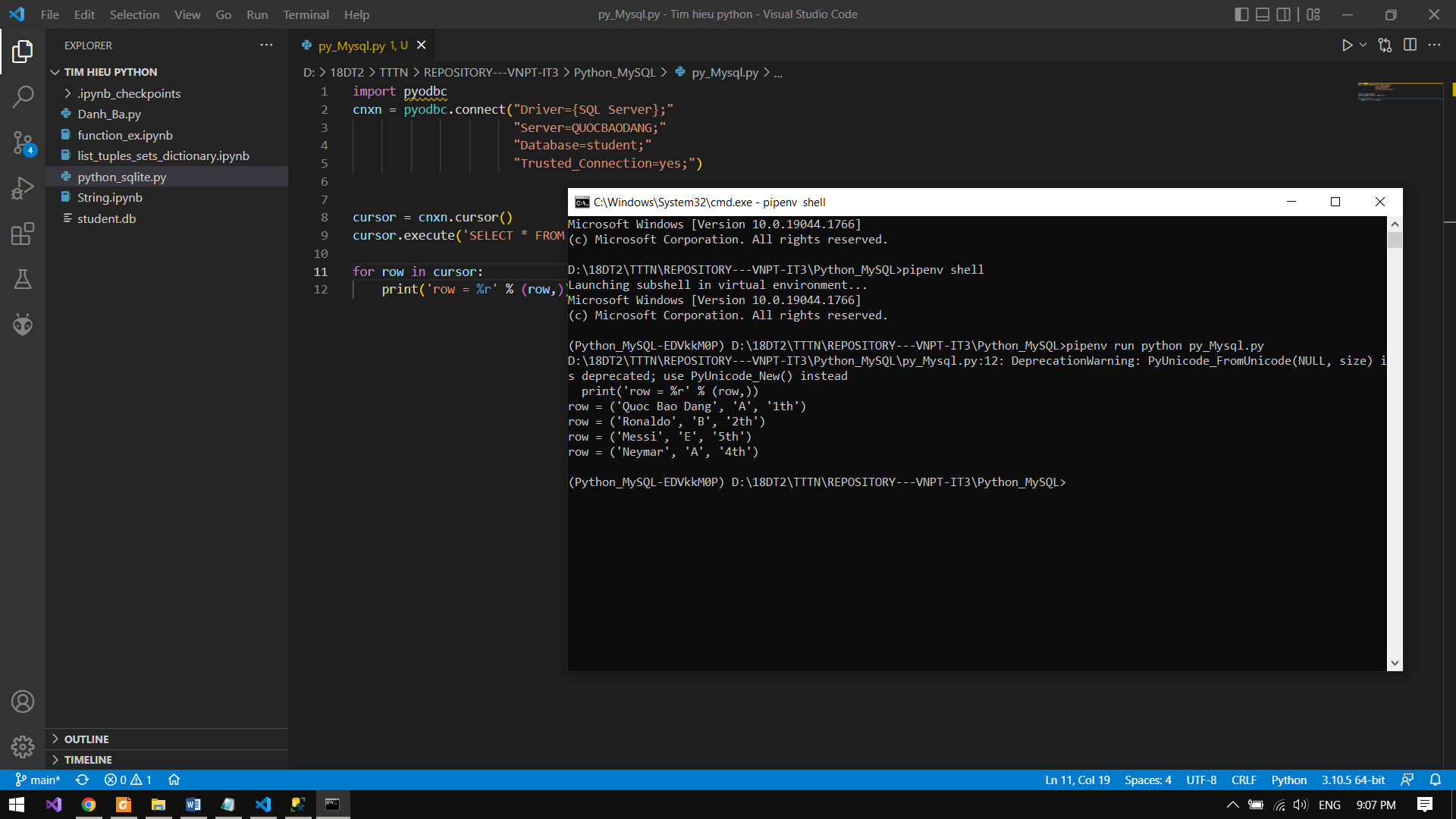
* Python với MySQL:

+ Để kết nối từ Python vào một CSDL cần một Driver, nó là một thư viện dùng để nói chuyện với CSDL. Để kết nối python với SQL server dùng thư viện pyodbc.

+ Thao tác kết nối:

* Cài đặt thư viện:

Pip install pyodbc



* Python với NoSQL:

+ NoSQL cho chúng ta một cách để lưu trữ và truy xuất dữ liệu, có thể lập mô hình ở các dạng khác với dạng quan hệ (bảng). NoSQL DB phần lớn được sử dụng trong các ứng dụng liên quan đến dữ liệu lớn và sử dụng thời gian thực, có thể hỗ trợ các ngôn ngữ truy vấn giống SQL. Có thể sử dụng NoSQL để lưu trữ dữ liệu ở các dạng như khóa-giá trị, tài liệu, cột và đồ thị.

+ Thao tác kết nối:

* Cài thư viên:

Pip install pymongo

from pymongo import MongoClient # khai báo thư viện pymongo

from pprint import pprint # sử dụng thư viện pprint

client=MongoClient() #tạo đối tượng client

db=client.test #kết nối đến database

student=db.student

student\_record={ # tạo record sinh viên

'Name':'Dang Quoc Bao',

'ID':'1654',

'Age':'22'}

result=student.insert\_one(student\_record) # insert record vào database

pprint(student.find\_one({'Age':'22'})) # show ra một sinh viên có tuổi 22

1. **Tìm hiểu về git:**

* Một số lệnh hay sử dụng:

git config –global user.name “[tên người dùng]” // cấu hình tên cho tài khoản git trên máy

git config –global user.email [email] // cấu hình email cho tài khoản git trên máy

git init // tạo repo ở local tại vị trí folder đang sử dụng terminal

git clone [url] // tải source về vị trí folder đang gọi terminal

git add [.] // thêm tất cả / [url] // từ vị trí gọi terminal đến file đó

git commit -m “commit message” // commit

git push origin branch //push du lieu len

….