**Report Mở Rộng Xử Lý Số Tín Hiệu 23/11/2020**

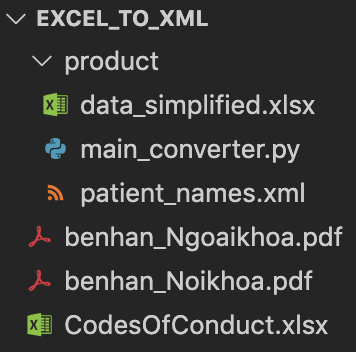
**Giáo viên hướng dẫn:** Thầy Nguyễn Thanh Tuấn

**Sinh viên thực hiện:** Thái Quang Nguyên

**MSSV:** 1813294

***Chủ đề: Chuyển đổi định dạng .xlsx thành định dạng .xml dùng ngôn ngữ Python.***

1. **Cấu tạo thư mục & thư viện cần có:**
2. **Cấu tạo thư mục:**

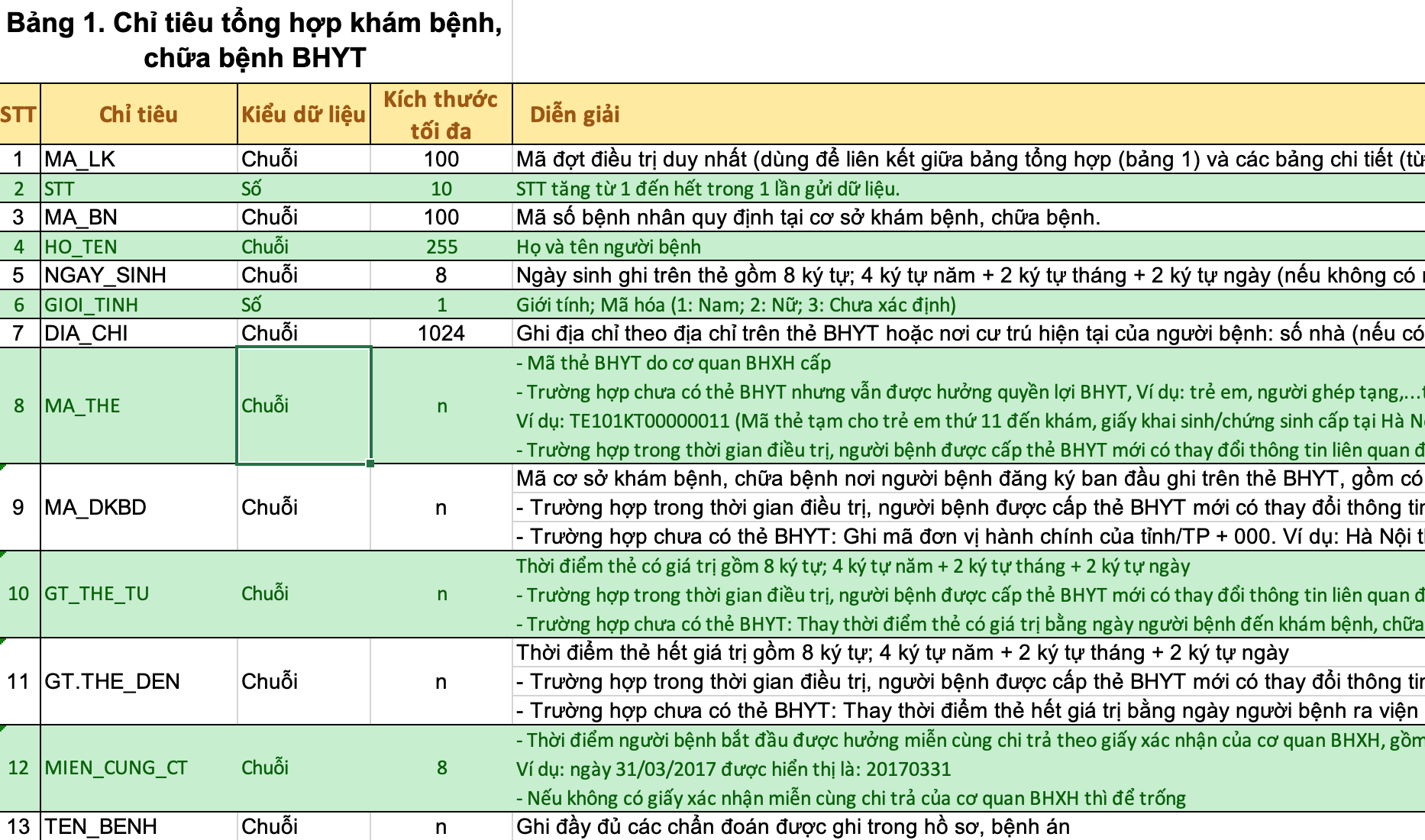
****Truy cập repo của project tại [link này](https://github.com/thaiquangnguyen/201_XLSTH_MR).

Dựa trên [trang web](https://thuvienphapluat.vn/van-ban/bao-hiem/Quyet-dinh-4210-QD-BYT-2017-du-lieu-dau-ra-trong-thanh-toan-chi-phi-kham-chua-benh-y-te-361955.aspx) chứa quy định chuẩn và định dạng dữ liệu trong quản lý y tế của *thuvienphapluat.vn*, em đã soạn lại một số tiêu chuẩn trong file excel *CodesOfConduct.xlsx* ở thư mục ngoài *EXCEL\_TO\_XML.* File này chứa những tiêu chuẩn của các trường của dữ liệu, chia làm 4 sheets, minh hoạ như hình 1.2.

Ở những file trong thư mục con *product*, chứa file *data\_simplified*.*xlsx* là file excel để trích xuất dữ liệu.

File *main\_converter.py* là file code python để đọc và xuất dữ liệu, data đó cuối cùng được lưu trong file *patient\_name.xml*.

*Hình 1.1. Cấu trúc thư mục*

**

*Hình 1.2. Nội dung minh hoạ của file CodeOfConduct.xlsx*

1. **Các thư viện cần có:**

* [openpyxl](https://openpyxl.readthedocs.io/en/stable/): thư viện dùng để đọc/ghi file Excel theo những định dạng xlsx/xlsm/xltx/xltm.
* [yattag](https://www.yattag.org/): thư viện dùng để tạo ra file HTML hoặc XML bằng code Python.
* [datetime](https://docs.python.org/3/library/datetime.html): thư viện để đọc và tính toán ngày tháng trong Python.

1. **Diễn giải code:**
2. **Cách thức đọc file excel:**

Trước tiên ta import hàm *load\_workbook()* từ thư viện *openpyxl* và gọi hàm với đối số là tên của file excel chứa dữ liệu:

from openpyxl import load\_workbook

wb = load\_workbook("data\_simplified.xlsx")

ws = wb.worksheets[0]

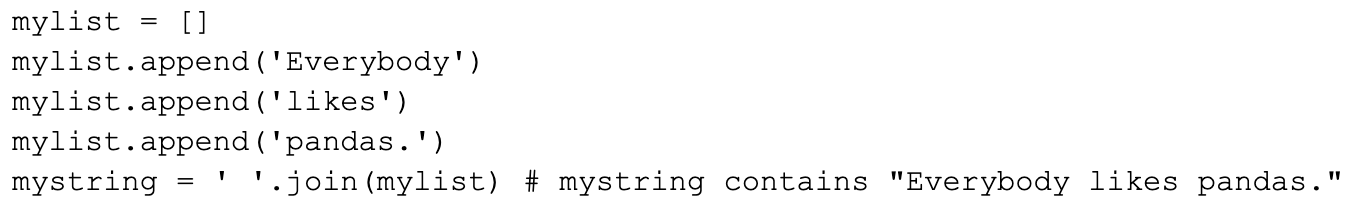
Tiếp đến ta import thư viện yattag để xuất định dạng xml:

from yattag import Doc, indent

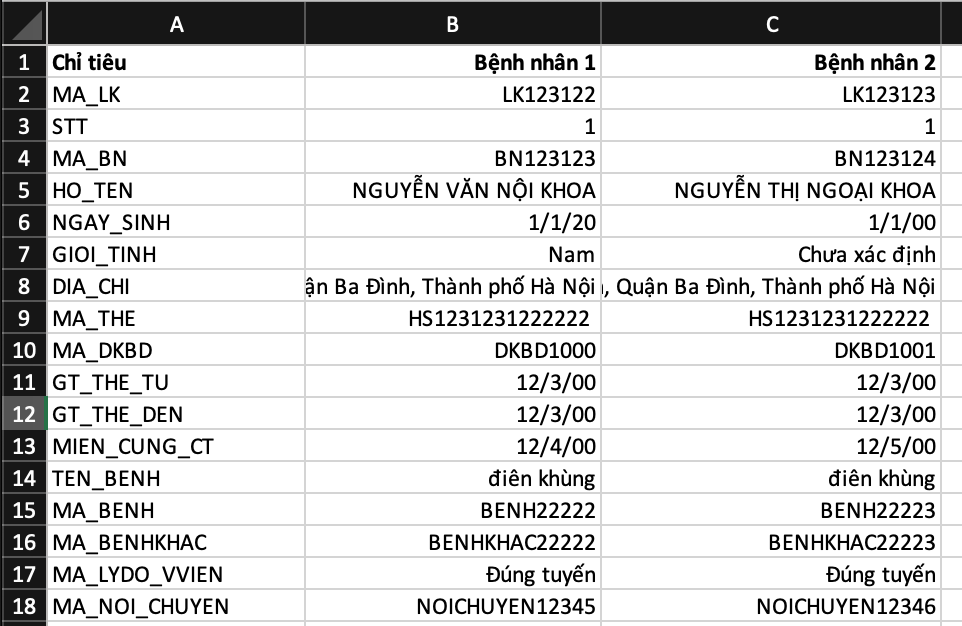
# Create Yattag doc, tag and text objects

doc, tag, text = Doc().tagtext()

Class yattag.Doc hoạt động như cách ta liên kết các chuỗi lại với nhau, ví dụ đơn giản như hình dưới:



*Hình 2.1. Cách thức hoạt động của class yattag.Doc*



Tạo mẫu file *data\_simplified.xlsx* chứa nội dung là các trường (cột A) và các thông số tương ứng của từng bệnh nhân (từ cột B trở đi).

Bây giờ ta sẽ đọc lần lượt từng bệnh nhân (từng cột B, C), trong mỗi bệnh nhân ta đọc từng hàng chính là thông số của các trường liên quan đến bệnh nhân. Các dòng code dưới giúp ta làm việc này:

*Hình 2.2. Nội dung file data\_simplified.xlsx*

1. **Cách thức tạo và validate các trường dữ liệu:**

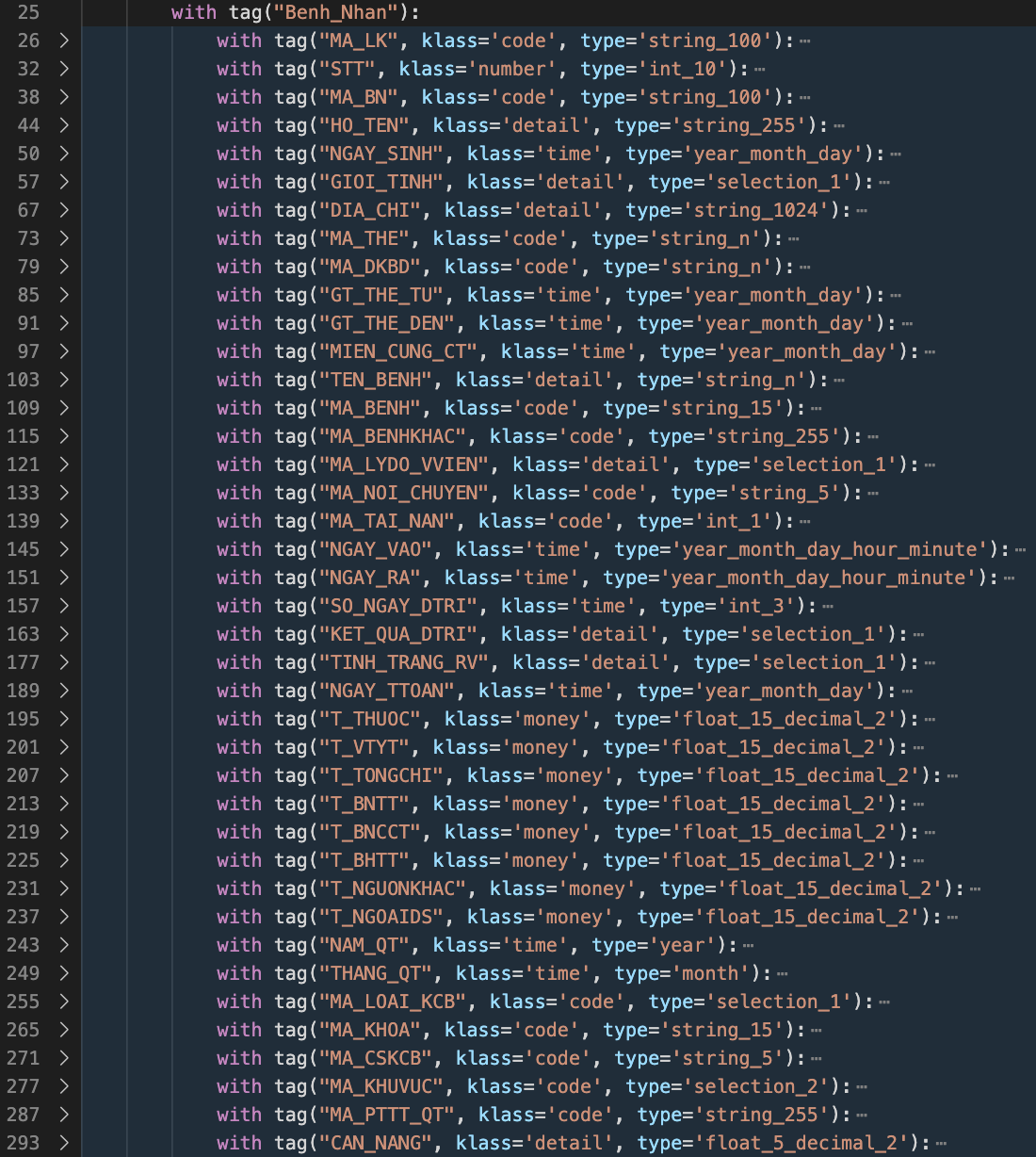
with tag('Cac\_Benh\_Nhan'):

# Use ws.max\_row for all rows

for col in ws.iter\_cols(min\_col=2, max\_col=3, min\_row=2, max\_row=41):

col = [cell.value for cell in col]

Tạo một tag *<Cac\_Benh\_Nhan></Cac\_Benh\_Nhan>* để lưu trữ thông tin từ file *data\_simplified.xlsx* trong một lần đọc dữ liệu. Dữ liệu của chúng ta bắt đầu từ hàng 2 cho đến hàng 41, và từ cột B cho đến C (tức là cột 2 và 3), ta khai báo *min\_row*, *max\_row*, *min\_col*, *max\_col* như trên là đối số của hàm *iter\_cols().* Ta lưu tất cả các giá trị của từng cell của một cột vào mảng một chiều *col*, sau này ta có thể truy xuất từng giá trị đó thông qua index của mảng *col*.

Với mỗi một trường dữ liệu, ta sẽ tạo một tag với tên tương ứng, cú pháp như hình bên. Ta có thể chia các trường dữ liệu theo *class* và *type*, ví dụ như class ‘time’ để chỉ thời gian và type *‘year\_month\_day’* để chỉ quy định của nội dung tag đó.

Type ‘int\_5’ là số integer khi chuyển thành chuỗi bằng lệnh str() thì len(str()) sẽ bé hơn hoặc bằng 5, tương ứng với yêu cầu trong file *CodeOfConduct.xlsx*.

*Hình 2.3. Các tag và class, type tương ứng*

Ở mỗi trường giá trị khi đọc vào, ta check xem có giá trị hay không hàm *if col[count] == None*, biến *count* được dùng để đọc lần lượt các hàng trong một cột, khi đọc xong một hàng thì sẽ được cộng lên 1. Điều kiện tiếp theo chính là chiều dài của dữ liệu có thoả mãn yêu cầu không, ta có ví dụ như trường dữ liệu *MA\_LK*:

with tag("MA\_LK", klass='code', type='string\_100'):

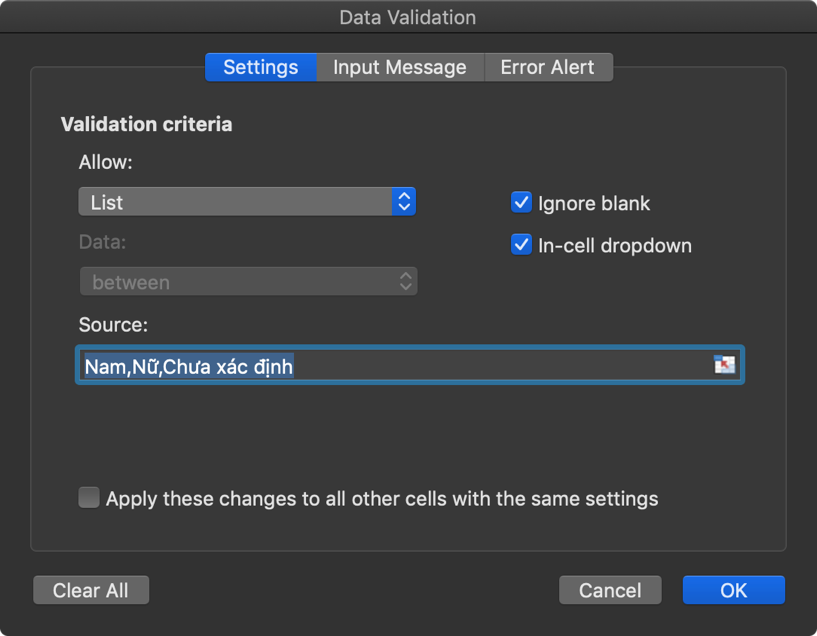
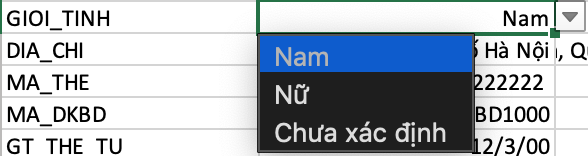
if col[count] == None or len(str(col[count])) > 100:

text("NODATA")

else:

text(col[count])

count += 1

Ở những trường như là ‘GIOI\_TINH’, giá trị nhập vào excel được phân thành những option là ‘Nam’, ‘Nữ’ hoặc ‘Chưa xác định’. Điều này được thực hiện trong file *data\_simplified.xlsx* bằng chức năng Data Validation như sau:

*Hình 2.4. Chức năng Data Validation trong EXCEL để tạo option box*

Nhờ đó, ta có thể rào được khoảng giá trị nhập vào của người nhập. Tiếp đến ta đọc cell này ở code python như sau:

with tag("GIOI\_TINH", klass='detail', type='selection\_1'):

if col[count] == None:

text("NODATA")

elif col[count] == "Nam":

text('1')

elif col[count] == "Nữ":

text('2')

elif col[count] == "Chưa xác định":

text('3')

count += 1

Tương tự như hàm switch case của C, ta xuất ký tự ‘1’, ‘2’, ‘3’ tương tự với các giá trị đầu vào là ‘Nam’, ‘Nữ’ và ‘Chưa xác định’.

Ở trường dữ liệu là thời gian, nhờ vào thư viện datetime, ta check xem đó có phải là biến datetime hay không bằng hàm *type()*, sau đó xuất ra theo định dạng mong muốn là *‘yyyymmdd’* bằng hàm *strftime(“%Y%m%d”)*.

with tag("NGAY\_SINH", klass='time', type='year\_month\_day'):

if type(col[count]) is datetime.datetime:

temp\_date = col[count]

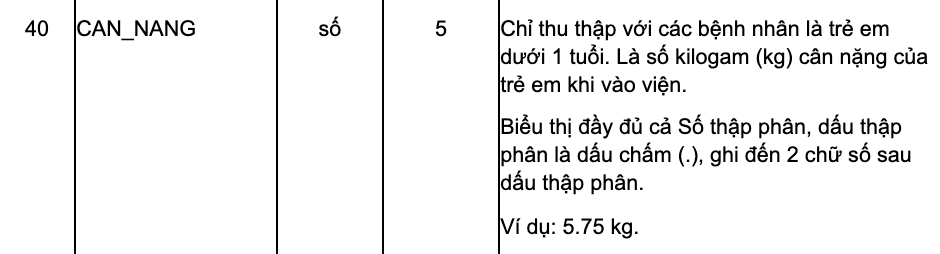
text(col[count].strftime("%Y%m%d"))

else:

text("NODATA")

count += 1

Ở đây, vì ở trường dữ liệu CAN\_NANG ở cuối, ta chỉ thu thập dữ liệu đối với các trẻ em dưới 1 tuổi, nên ta sẽ dùng thư viện datetime để lấy giá trị của thời điểm hiện tại, trừ đi cho giá trị ngày sinh nhập vào xem có bé hơn 365 ngày không (ở đây chưa xét đến yếu tố năm nhuận hay múi giờ).



*Hình 2.5. Quy định của trường dữ liệu CAN\_NANG*

current\_date = datetime.datetime.now()

temp\_date = datetime.datetime(2020, 11, 22)

#...

with tag("CAN\_NANG", klass='detail', type='float\_5\_decimal\_2'):

if (current\_date - temp\_date).days <= 365:

if type(col[count]) == float and len(str(col[count])) <= 5:

text(col[count])

age = (current\_date - temp\_date).days

print(current\_date)

print(temp\_date)

print(age)

else:

text("NODATA")

Biến *current\_date* được gán cho thời điểm đọc file excel, biến *temp\_date* được khởi tạo và về sau được gán bằng giá trị ngày sinh của bệnh nhân. Hiệu của hai giá trị này tính ra số ngày nếu thoả bé hơn hoặc bằng 365 thì thoả mãn, tính được bằng hàm *(current\_date - temp\_date).days*.

Cuối cùng, ta có thể lưu lại các giá trị gồm các tag, text vào file *patient\_names.xml*:

result = indent(

doc.getvalue(),

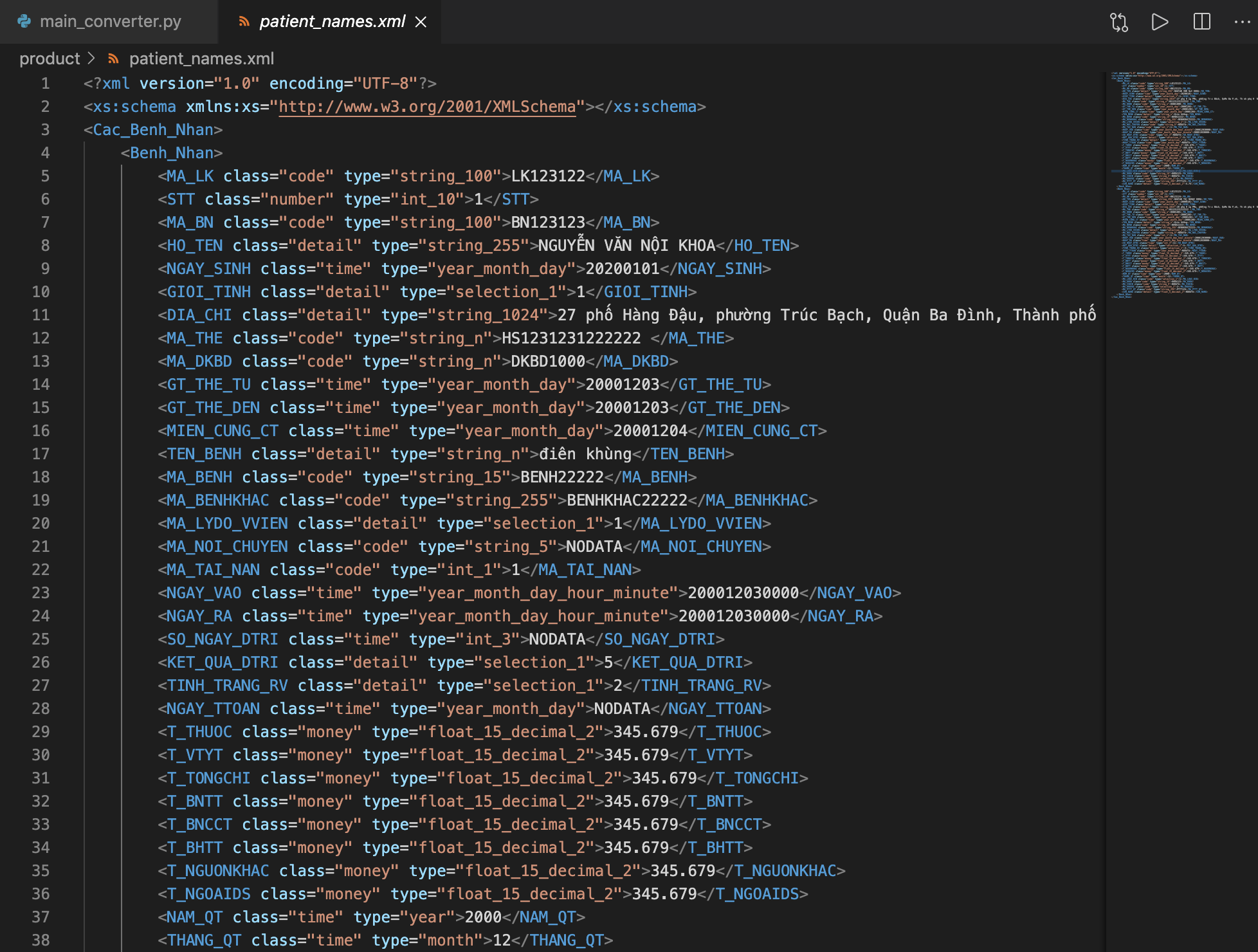
indentation = ' '

)

with open("patient\_names.xml", "w") as f:

f.write(result)

Kết quả thu được như hình dưới:



*Hình 2.6. File patient\_names.xml xuất ra được*