

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ

Bộ môn: Công nghệ Thông tin



BÀI TẬP KẾT THÚC MÔN HỌC
LẬP TRÌNH PYTHON

NGÀNH : KỸ THUẬT MÁY TÍNH
HỆ : ĐẠI HỌC CHÍNH QUY

ĐỀ BÀI: TẠO ỨNG DỤNG GUI CHO PHÉP CHỌN THƯ MỤC,
LIỆT KÊ FILE THEO TỪNG LOẠI (.TXT, .PY, .JPG),
VÀ CHO PHÉP MỞ FILE.

Thái Nguyên – 2025

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

Bộ môn : Công nghệ Thông tin



BÀI TẬP KẾT THÚC MÔN HỌC LẬP TRÌNH PYTHON

Giáo viên giảng dạy

: Nguyễn Văn Huy

Sinh viên

: Phương Thị Ánh Nguyệt

Lớp

: K58KTP

https://github.com/phuongthianhnguyet/AnhNguyet_Python.git

Thái Nguyên - 2025

TRƯỜNG ĐHKTCN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN TỬ

Độc lập – Tự do – Hạnh phúc

BÀI TẬP KẾT THÚC MÔN HỌC LẬP TRÌNH PYTHON

Giáo viên hướng dẫn: Nguyễn Văn Huy

Sinh viên : Phương Thị Ánh Nguyệt

Mssv : K225480106098

Lớp : K58KTP

Ngành : Công nghệ thông tin

Ngày giao đề: 20/05/2025

Ngày hoàn thành: 10/06/2025.

Tên đề tài: Tạo ứng dụng GUI cho phép chọn thư mục, liệt kê file theo từng loại (.txt, .py, .jpg), và cho phép mở file.

Yêu cầu:

- Sử dụng os để scan folder.
- Bắt lỗi không tìm thấy đường dẫn.
- GUI với Treeview (tkinter.ttk).
- Mở file bằng chương trình mặc định.

Thái Nguyên, ngày tháng năm 2025

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

TRƯỜNG ĐHKTCN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN TỬ

Độc lập – Tự do – Hạnh phúc

PHIẾU GHI ĐIỂM

HƯỚNG DẪN BÀI TẬP KẾT THÚC MÔN HỌC

GVHD : Nguyễn Văn Huy
Sinh viên : Phương Thị Ánh Nguyệt
Mssv : K225480106098
Lớp : K58KTP

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....
.....
.....
.....

Xếp loại : Điểm :

Thái Nguyên, ngày tháng năm 2025

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

MỤC LỤC

LỜI CAM ĐOAN	2
LỜI MỞ ĐẦU	3
DANH MỤC HÌNH ẢNH.....	4
CHƯƠNG 1. GIỚI THIỆU ĐẦU BÀI	5
1.1. Lý do chọn đề tài	5
1.2. Mục tiêu của chương trình	5
1.3. Tính năng chính của chương trình.....	5
1.4. Thách thức khi thực hiện đề tài	5
1.5. Kiến thức đã vận dụng	6
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	7
2.1. Cấu trúc dữ liệu trong Python.	7
2.2. Tổng kết.....	8
CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH.....	9
3.1. Sơ đồ khối hệ thống.	9
3.1.1. Mô tả các module chính trong chương trình.	9
3.1.2. Biểu đồ phân cấp chức năng.	11
3.2. Sơ đồ khối các thuật toán chính.	11
3.2.1. Sơ đồ khối các thuật toán.....	11
3.2.2. Phân tích các khối thuật toán chính.	14
3.2.3. Môi quan hệ đầu ra – đầu vào giữa các thuật toán	15
3.3. Cấu trúc dữ liệu.....	16
3.3.1. Cấu trúc dữ liệu đầu vào.....	16
3.4. Chương trình.....	17
3.4.1. Các hàm chính.	17
CHƯƠNG 4. THỰC NGHIỆM VÀ KẾT LUẬN	20
4.1. Thực nghiệm.	20
4.2. Kết luận.	21
TỔNG KẾT.....	23
TÀI LIỆU THAM KHẢO	24

LỜI CAM ĐOAN

Em xin cam đoan rằng bài tập kết thúc môn “Lập trình Python” là kết quả nghiên cứu và thực hiện của em, dưới sự hướng dẫn vô cùng tận tình của thầy giáo Nguyễn Văn Huy.

Trong quá trình thực hiện bài tập kết thúc môn học, em đã tìm hiểu, phân tích, thiết kế dựa trên kiến thức đã học và các tài liệu tham khảo có trích dẫn rõ ràng. Em hoàn toàn chịu trách nhiệm về tính trung thực và nội dung bài làm của mình.

Em cam kết mọi thông tin, số liệu và kết quả trong bài tập là chính xác và trung thực. Em xin chịu hoàn toàn trách nhiệm với thầy cô, khoa, nhà trường nếu vi phạm và không trung thực.

LỜI MỞ ĐẦU

Trong thời đại công nghệ số hiện nay, việc tổ chức và quản lý dữ liệu một cách trực quan, tiện lợi đóng vai trò vô cùng quan trọng. Python với ưu điểm là ngôn ngữ lập trình đơn giản, mạnh mẽ và dễ tiếp cận – đã trở thành một trong những công cụ phổ biến được sử dụng để xây dựng các ứng dụng hỗ trợ xử lý dữ liệu hiệu quả.

Thông qua môn học “Lập trình Python”, em đã được trang bị kiến thức cơ bản về cú pháp ngôn ngữ, tư duy lập trình hướng đối tượng, cũng như khả năng xây dựng các ứng dụng có giao diện đồ họa (GUI). Để vận dụng các kiến thức đã học, em thực hiện bài tập lớn với đề tài: *“Tạo ứng dụng GUI cho phép chọn thư mục, liệt kê file theo từng loại (.txt, .py, .jpg), và cho phép mở file”*.

Đề tài không chỉ giúp em củng cố kỹ năng lập trình với thư viện tkinter, mà còn rèn luyện khả năng xử lý file hệ thống, tạo giao diện người dùng thân thiện và xây dựng ứng dụng thực tế phục vụ nhu cầu cá nhân cũng như học thuật. Qua đó, em hiểu rõ hơn về quy trình thiết kế phần mềm theo từng bước cụ thể: từ phân tích yêu cầu, thiết kế giao diện đến kiểm thử và hoàn thiện ứng dụng.

Em xin chân thành cảm ơn thầy Nguyễn Văn Huy đã tận tình giảng dạy và tạo điều kiện để em có cơ hội thực hiện đề tài này để hoàn thành môn học trong suốt kì học vừa qua!

DANH MỤC HÌNH ẢNH

Hình 3.1. Sơ đồ khối hệ thống trình quản lý thư mục GUI

Hình 3.2. Biểu đồ phân cấp chức năng của trình quản lý thư mục GUI

Hình 3.3. Thuật toán chon_thu_muc()

Hình 3.4. Thuật toán hien_thi_file()

Hình 3.5. Thuật toán mo_file()

Bảng 3.6. Thuật toán và chức năng chính.

Hình 4.1. Giao diện thư mục GUI khi đã chọn ra các file có đuôi .py, .txt

Hình 4.2. Click vào 1 file bất kì đã chọn lọc và hiển thị nội dung ra notepad

Bảng 4.3. Hướng cải tiến trong tương lai.

CHƯƠNG 1. GIỚI THIỆU ĐẦU BÀI

1.1. Lý do chọn đề tài

Trong thời đại công nghệ phát triển nhanh chóng, việc quản lý tệp và thư mục trên máy tính trở thành một nhu cầu thiết yếu. Tuy nhiên, không phải người dùng nào cũng có kỹ năng để thao tác hiệu quả với các thư mục chứa nhiều định dạng file khác nhau. Chính vì vậy, việc xây dựng một ứng dụng đơn giản, trực quan, cho phép người dùng chọn thư mục, liệt kê và mở các tệp phổ biến như .txt, .py, .jpg là rất thiết thực và hữu ích.

Việc lựa chọn đề tài này giúp em có cơ hội vận dụng kiến thức đã học để giải quyết bài toán thực tế, đồng thời tiếp cận với quy trình phát triển phần mềm có giao diện người dùng (GUI) – một kỹ năng quan trọng đối với lập trình viên.

1.2. Mục tiêu của chương trình

Ứng dụng được xây dựng nhằm đạt các mục tiêu sau:

- Tạo giao diện đồ họa thân thiện để người dùng chọn thư mục.
- Liệt kê tất cả các file trong thư mục, phân loại theo yêu cầu của đề bài với định dạng (.txt, .py, .jpg).
- Hiển thị danh sách file bằng thành phần Treeview (thư viện tkinter.ttk).
- Cho phép người dùng mở file bằng chương trình mặc định của hệ điều hành (Notepad, Python IDE, trình xem ảnh, v.v).

1.3. Tính năng chính của chương trình

- Chọn thư mục bằng hộp thoại đồ họa (sử dụng askdirectory()).
- Liệt kê các file có trong thư mục, hiển thị dạng cây bằng Treeview.
- Lọc và phân loại file theo định dạng phổ biến: .txt, .py, .jpg.
- Xử lý sự kiện nhấn nút hoặc double-click để mở file.
- Báo lỗi nếu thư mục không tồn tại hoặc không có file phù hợp.

1.4. Thách thức khi thực hiện đề tài

- Thiết kế giao diện đơn giản nhưng trực quan, dễ sử dụng.
- Hiển thị file đúng định dạng, xử lý lỗi nếu định dạng không đúng yêu cầu của đầu bài.
- Quản lý sự kiện GUI như double-click, nút mở file.
- Tương thích với hệ điều hành khi mở file bằng chương trình mặc định.
- Làm quen với Treeview, os module và thư viện tkinter của Python.

1.5. Kiến thức đã vận dụng

Khi thực hiện đề tài này, em đã áp dụng và củng cố các kiến thức đã học trong môn Lập trình Python, bao gồm:

- Xử lý tệp và thư mục: sử dụng module os để duyệt và phân tích đường dẫn, file.
- Lập trình giao diện GUI với tkinter: tạo Frame, Button, Treeview.
- Xử lý sự kiện người dùng: bind sự kiện click, chọn file, hiển thị nội dung.
- Tư duy phân tích, thiết kế chương trình: chia nhỏ vấn đề thành từng hàm, từng chức năng rõ ràng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Cấu trúc dữ liệu trong Python.

a) Danh sách (List) trong Python

Trong chương trình này, danh sách (list) được sử dụng linh hoạt và hiệu quả cho nhiều mục đích khác nhau. Hàm `os.listdir()` trả về danh sách tất cả file trong thư mục, sau đó được lọc theo đuôi file mong muốn bằng phương thức `endswith()` với một tuple chứa các đuôi file (`.txt`, `.py`, `.jpg`). `Treeview` sử dụng tuple columns để định nghĩa các cột hiển thị, trong khi `tree.getchildren()` trả về danh sách các item hiện có để xóa dữ liệu cũ. Mỗi bản ghi trong `Treeview` được thêm vào bằng một tuple giá trị tương ứng với các cột. Cách sử dụng list/tuple này giúp chương trình xử lý dữ liệu file một cách có hệ thống, dễ bảo trì và mở rộng khi cần thêm định dạng file hoặc cột thông tin mới.

b) Tkinter

Thư viện Tkinter là thư viện đồ họa (GUI) chuẩn của Python, dùng để tạo ứng dụng desktop với giao diện trực quan. Đây là công cụ quan trọng nhất trong đoạn code của bạn, đóng vai trò:

- Tạo cửa sổ chính.
- Xây dựng các thành phần GUI:
 - Nút bấm (Button) để chọn thư mục.
 - Bảng hiển thị (Treeview) để liệt kê file.
 - Hộp thoại (Filedialog) để duyệt thư mục.
 - Thông báo lỗi (Messagebox).
- Là thư viện tiêu chuẩn để tạo GUI trong Python.
- Được sử dụng để tạo cửa sổ ứng dụng, nút bấm, hộp thoại chọn thư mục.

c) Treeview (tkinter.ttk)

`Treeview (tkinter.ttk)` là một widget mạnh mẽ trong Tkinter, được dùng để hiển thị dữ liệu dạng phân cấp (cây) hoặc dạng bảng. Trong code của bạn, nó đóng vai trò là bảng hiển thị danh sách file. Tính năng:

- Tương tác với file: Bắt sự kiện double-click để mở file:
- Thành phần giao diện để hiển thị dữ liệu dạng cây.
- Dùng để trình bày danh sách file gọn gàng, phân nhóm dễ quan sát.

d) Các thành phần GUI khác

- Button: nút bấm “Chọn thư mục”, “Mở file”.
- Label: hiển thị thông báo lỗi hoặc đường dẫn thư mục.
- Askdirectory(): hộp thoại cho người dùng chọn thư mục.

e) Phân loại và lọc file theo định dạng

- Sau khi quét thư mục, chương trình cần lọc các tệp theo đuôi .txt, .py, .jpg.
- Dùng phương pháp duyệt file và kiểm tra file.endswith('.txt'), v.v.

f) Quản lý lỗi

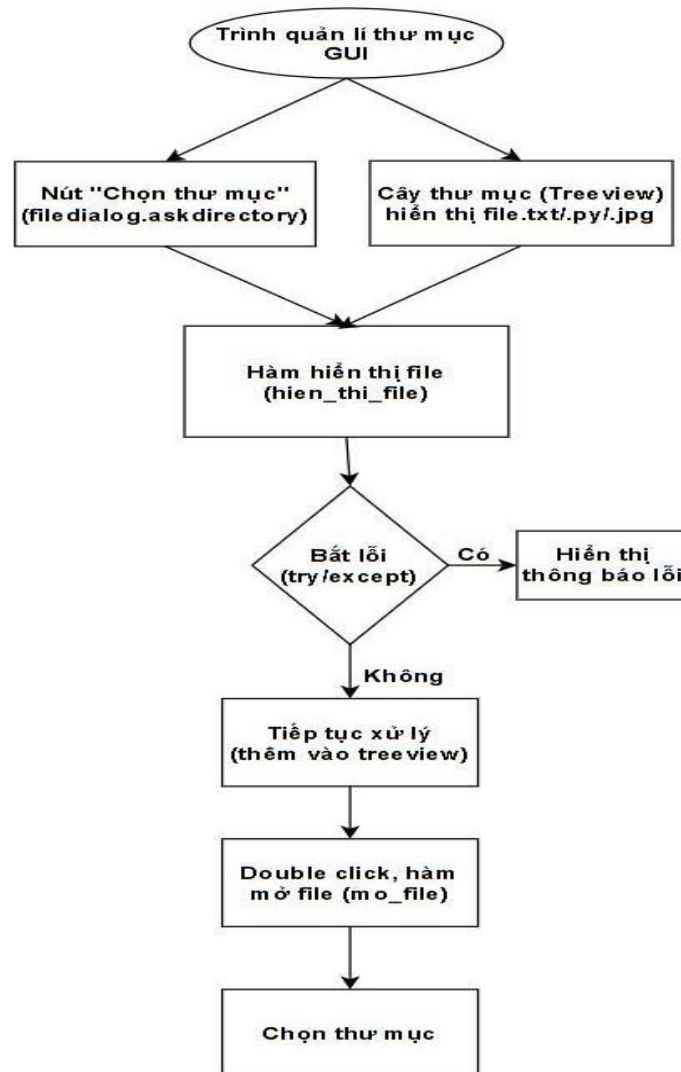
Khi người dùng chọn thư mục không tồn tại hoặc không chứa file phù hợp, chương trình cần xử lý bằng câu lệnh try...except, hoặc hiển thị cảnh báo thông qua messagebox.

2.2. Tổng kết

Toàn bộ chương trình được xây dựng dựa trên kiến thức cơ bản nhưng thiết thực của Python: xử lý file, quản lý thư mục, lập trình GUI với tkinter, cấu trúc dữ liệu (list, dict), xử lý sự kiện và lỗi. Đây là nền tảng vững chắc để sinh viên phát triển các ứng dụng có giao diện đồ họa trong thực tế.

CHƯƠNG 3. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

3.1. Sơ đồ khối hệ thống.



Hình 3.1. Sơ đồ khối hệ thống trình quản lí thư mục GUI

3.1.1 Mô tả các module chính trong chương trình.

a) GUI (Giao diện người dùng)

Chức năng:

- Tạo cửa sổ ứng dụng chính.
- Hiển thị nút "Chọn thư mục".
- Hiển thị bảng danh sách file bằng Treeview (tk.Treeview).
- Bố trí giao diện trực quan để người dùng thao tác dễ dàng.

Cách triển khai:

- Sử dụng tkinter.Tk() để tạo cửa sổ.
- Sử dụng tk.Button() và tk.Treeview() để tạo các thành phần.

- Giao diện phản hồi với người dùng qua các thao tác click, double-click.

b) File Scanner (Trình quét file)

Chức năng:

- Quét thư mục người dùng đã chọn.
- Lọc file theo định dạng .txt, .py, .jpg.
- Thêm thông tin file vào bảng Treeview.

Cách triển khai:

- Sử dụng `os.listdir()` để lấy danh sách file.
- Dùng `os.path.splitext()` để xác định định dạng file.
- Duyệt qua các file hợp lệ và `tree.insert()` để hiển thị.

c) Error Handler (Xử lý lỗi)

Chức năng:

- Xử lý lỗi khi thư mục không tồn tại hoặc không thể truy cập.
- Xử lý lỗi khi không mở được file.

Cách triển khai:

- Sử dụng `try-except` trong hàm `hien_thi_file()` và `mo_file()`.
- Thông báo lỗi cho người dùng bằng `messagebox.showerror()`.

d) File Opener (Mở file)

Chức năng:

- Mở file đã chọn bằng chương trình mặc định của hệ điều hành (Notepad, Photos...).

Cách triển khai:

- Lấy đường dẫn từ Treeview bằng `tree.item()['values']`[2].
- Gọi `os.startfile()` để mở file.
- Bắt lỗi nếu không thể mở và thông báo cho người dùng.

e) Event Binding (Gán sự kiện)

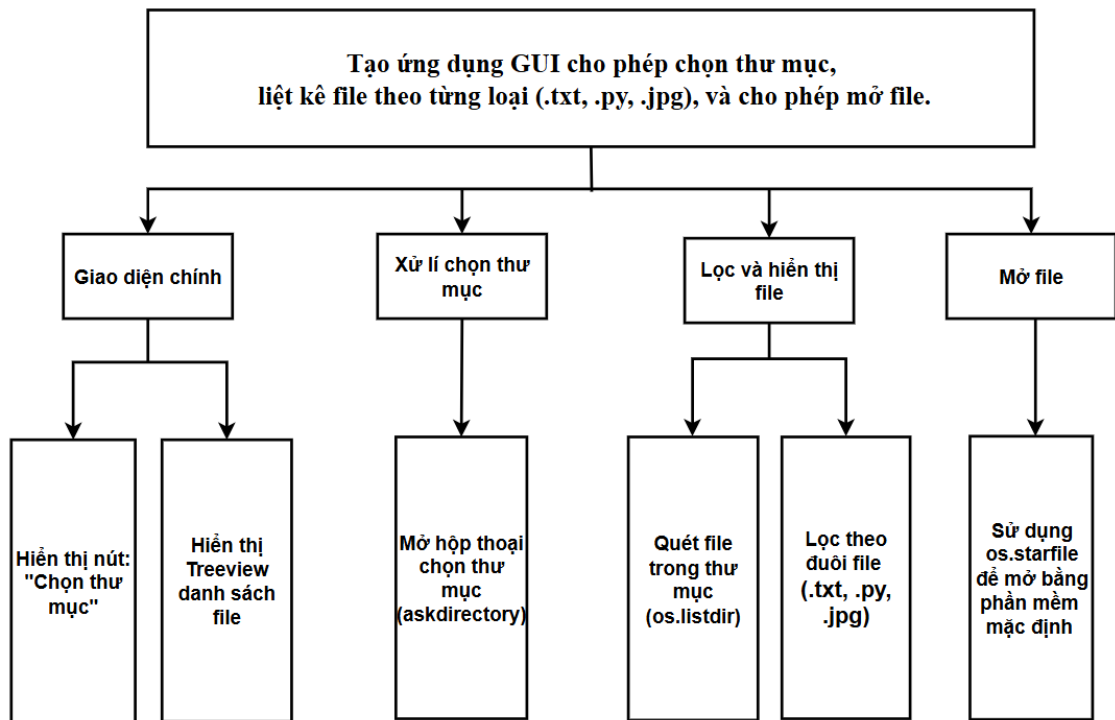
Chức năng:

- Gán sự kiện double-click để mở file.
- Gán sự kiện nút "Chọn thư mục" để chọn đường dẫn mới.

Cách triển khai:

- Dùng `tree.bind("<Double-1>", mo_file)` để xử lý double-click.
- Gán `command=chon_thu_muc` cho nút chọn thư mục.

3.1.2 Biểu đồ phân cấp chức năng.



Hình 3.2. Biểu đồ phân cấp chức năng của trình quản lý thư mục GUI

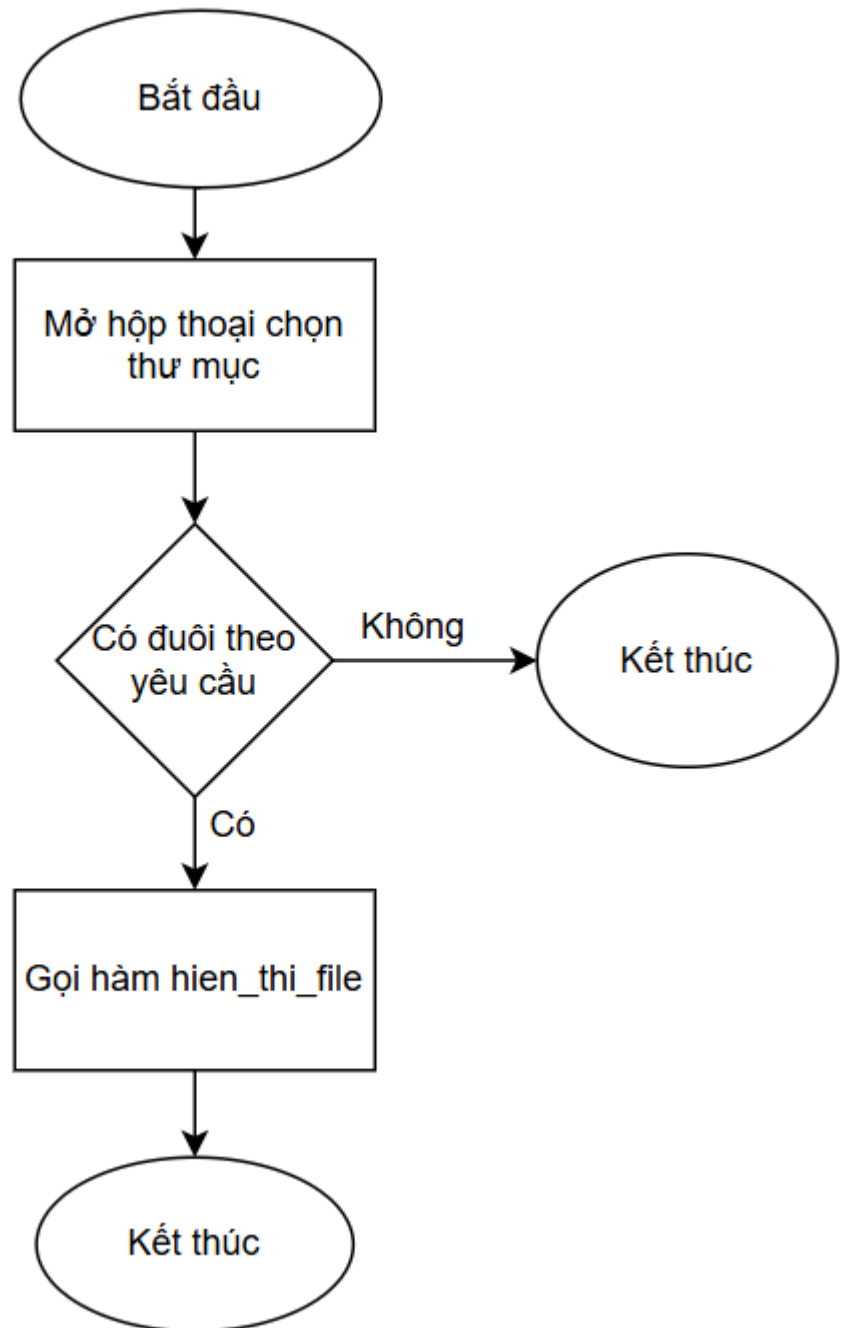
3.2. Sơ đồ khối các thuật toán chính.

3.2.1 Sơ đồ khối các thuật toán

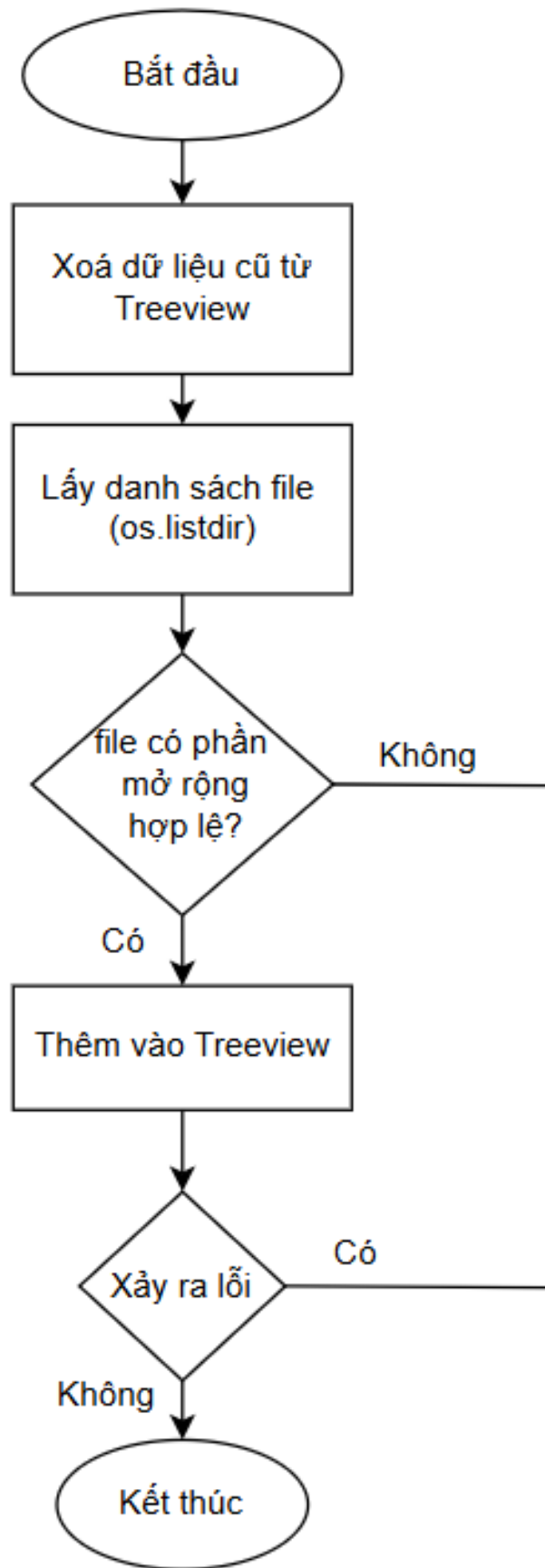
Ở đề bài này có 3 thuật toán chính:

Mô tả:

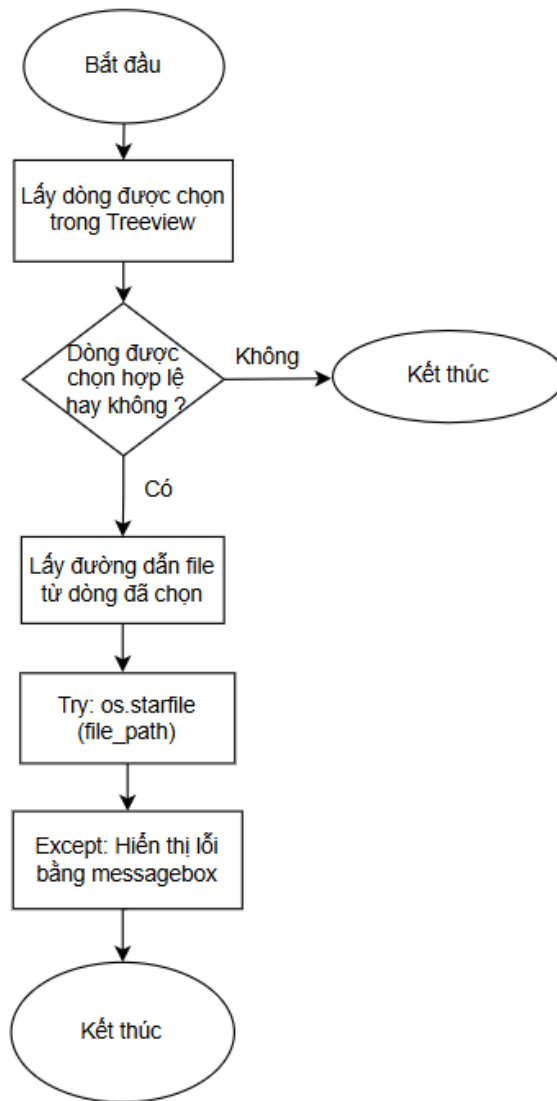
- *Chon_thu_muc()*=>*chọn thư mục*: Mở hộp thoại chọn thư mục, truyền đường dẫn sang *hien_thi_file()*
- *Hien_thi_file()*=>*hiển thị danh sách file*: Xoá dữ liệu cũ, quét thư mục, lọc file, chèn vào Treeview, xử lý lỗi
- *Mo_file()*=>*mở file*: Mở file khi double-click, xử lý lỗi nếu không mở được



Hình 3.3. Thuật toán `chon_thu_muc()`



Hình 3.4. Thuật toán hien_thi_file()



Hình 3.5. Thuật toán *mo_file()*

3.2.2 Phân tích các khối thuật toán chính.

Thuật toán	Chức năng chính
chon_thu_muc()	Lấy đường dẫn thư mục từ người dùng.
hien_thi_file()	Quét thư mục, lọc file, đẩy dữ liệu vào Treeview, xử lý lỗi.
mo_file()	Lấy dòng được chọn, mở file tương ứng bằng chương trình mặc định.

Bảng 3.6. Thuật toán và chức năng chính.

3.2.3 Mối quan hệ đầu ra – đầu vào giữa các thuật toán

a) Khối: `chon_thu_muc()`

- Đầu vào: Hành động người dùng: Nhấn nút “Chọn thư mục” trên giao diện.
- Xử lý: Gọi `filedialog.askdirectory()` để hiện hộp thoại chọn thư mục
- Đầu ra: Đường dẫn thư mục (kiểu str), nếu người dùng chọn xong
- Chuyển tiếp tới: Gọi hàm `hien_thi_file(folder)` và truyền đường dẫn vừa chọn

b) Khối: `hien_thi_file(folder)`

- Đầu vào: Đường dẫn thư mục (từ `chon_thu_muc()`)
- Xử lý:
 - Xóa dữ liệu cũ trong Treeview
 - Duyệt file trong thư mục (`os.listdir`)
 - Lọc các file theo đuôi `.txt`, `.py`, `.jpg`
 - Thêm file vào Treeview (sử dụng `tree.insert`)
 - Bắt lỗi nếu không truy cập được thư mục |
- Đầu ra: Treeview được cập nhật – chứa các hàng tương ứng với tên file, loại file và đường dẫn đầy đủ
- Chuyển tiếp tới: Cho phép người dùng tương tác bằng double-click, được xử lý trong `mo_file()`

c) Khối: `mo_file(event)`

- Đầu vào: Hành động người dùng: Double-click vào một dòng trong Treeview
- Xử lý:
 - Lấy dòng được chọn (`tree.focus()`)
 - Lấy đường dẫn file từ dòng đó
 - Gọi `os.startfile()` để mở file
 - Bắt lỗi nếu không mở được
- Đầu ra: Mở file bằng chương trình mặc định của hệ thống hoặc hiển thị thông báo lỗi

3.3 Cấu trúc dữ liệu

Cấu trúc dữ liệu chính: Treeview

Trong chương trình, dữ liệu được hiển thị và lưu trữ tạm thời trong một widget ttk.Treeview. Đây là thành phần dùng để:

- Lưu trữ danh sách các file trong thư mục được chọn.
- Hiển thị lên giao diện người dùng dưới dạng bảng 3 cột.

3.3.1 Cấu trúc dữ liệu đầu vào.

a) Loại đầu vào chính.

Chương trình nhận đầu vào từ người dùng (chọn thư mục), sau đó xử lý để lấy thông tin các file trong thư mục đó.

- Đầu vào: folder
- Kiểu dữ liệu: str
- Nguồn từ `filedialog.askdirectory()`
- Mục đích Lấy đường dẫn thư mục do người dùng chọn

Ví dụ: `folder = "C:/Users/User/Documents/MyFolder"`

b) Dữ liệu đầu vào tiếp theo từ hệ thống (nội dung thư mục)

Sau khi chọn thư mục, chương trình dùng `os.listdir(folder)` để lấy danh sách tên file trong thư mục.

- Tên biến: file
- Kiểu dữ liệu: str
- Nguồn: `os.listdir(folder)`

Lấy tên file trong thư mục.

c) Kiểm tra lọc file đầu vào

Chương trình chỉ lấy các file có phần mở rộng hợp lệ:

→ Điều kiện lọc đầu vào: chỉ chọn file có định dạng `.txt`, `.py`, hoặc `.jpg`.

Code: `if file.endswith((''.txt', '.py', '.jpg')):`

→ Và mỗi bản ghi sẽ được đưa vào Treeview. Kết quả thu được là các file có đuôi theo như yêu cầu của đầu bài.

3.4. Chương trình

3.4.1 Các hàm chính.

a) Hàm chon_thu_muc()

```
def chon_thu_muc():  
    folder = filedialog.askdirectory()  
    if not folder:  
        return  
    hien_thi_file(folder)
```

Chức năng:

- Gọi hộp thoại để người dùng chọn thư mục.
- Gọi tiếp hàm hien_thi_file(folder) để hiển thị danh sách file trong thư mục đã chọn.

Vai trò:

- Đầu vào: Người dùng chọn thư mục từ hộp thoại.
- Đầu ra: Gọi hàm hiển thị file từ thư mục đó.

b) Hàm hien_thi_file(folder)

```
def hien_thi_file(folder):  
    for i in tree.get_children():  
        tree.delete(i)  
    try:  
        for file in os.listdir(folder):  
            if file.endswith((' .txt', '.py', '.jpg')):  
                file_path = os.path.join(folder, file)  
                file_type = os.path.splitext(file)[1]  
                tree.insert("", "end", values=(file, file_type, file_path))  
    except Exception as e:  
        messagebox.showerror("Lỗi", f"Không thể đọc thư mục: {e}")
```

Chức năng:

- Xóa dữ liệu cũ trong Treeview.
- Quét thư mục và lọc các file có đuôi .txt, .py, .jpg.
- Hiển thị file hợp lệ lên Treeview (bảng hiển thị).

Vai trò:

- Đầu vào: Đường dẫn thư mục (từ `chon_thu_muc`).
- Xử lý: Quét thư mục, lọc file, xử lý lỗi.
- Đầu ra: Treeview hiển thị danh sách file hợp lệ.

c) Hàm `mo_file(event)`

```
def mo_file(event):
    selected = tree.focus()
    if selected:
        file_path = tree.item(selected)['values'][2]
        try:
            os.startfile(file_path)
        except Exception as e:
            messagebox.showerror("Lỗi", f"Không thể mở file: {e}")
```

Chức năng:

- Khi người dùng double-click một file trong Treeview → lấy đường dẫn và mở file bằng chương trình mặc định.

Vai trò:

- Đầu vào: Sự kiện double-click vào item.
- Xử lý: Lấy đường dẫn file từ Treeview, gọi `os.startfile()` để mở.
- Đầu ra: Mở file; nếu lỗi → hiện thông báo.

d) Phần code chính (Main program)

Chức năng: Thiết lập giao diện chính của ứng dụng

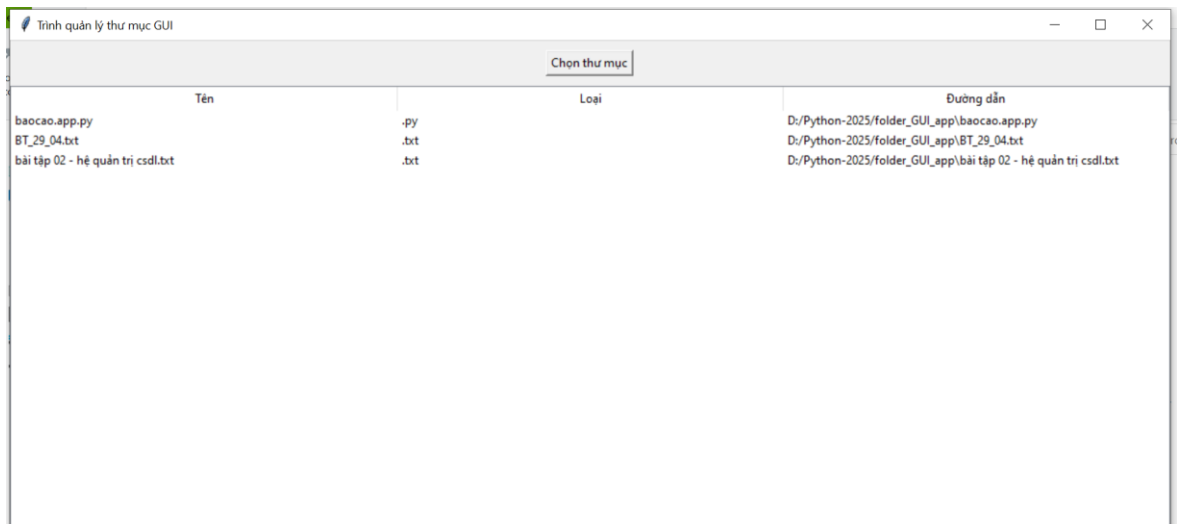
```
# Tạo cửa sổ chính
root = tk.Tk()
root.title("Trình quản lý thư mục GUI")
root.geometry("700x400")
# Tạo nút chọn thư mục
btn_select = tk.Button(root, text="Chọn thư mục",
command=chon_thu_muc)
btn_select.pack(pady=10)
# Tạo Treeview để hiển thị file
columns = ("Tên", "Loại", "Đường dẫn")
tree = ttk.Treeview(root, columns=columns, show="headings")
# Thiết lập tiêu đề và độ rộng cho các cột
```

```
for col in columns:
    tree.heading(col, text=col)
    tree.column(col, width=200)
tree.pack(fill=tk.BOTH, expand=True) # Đặt Treeview vào cửa sổ
# Gán sự kiện double click cho Treeview
tree.bind("<Double-1>", mo_file)
# Chạy vòng lặp chính của ứng dụng
root.mainloop()
```

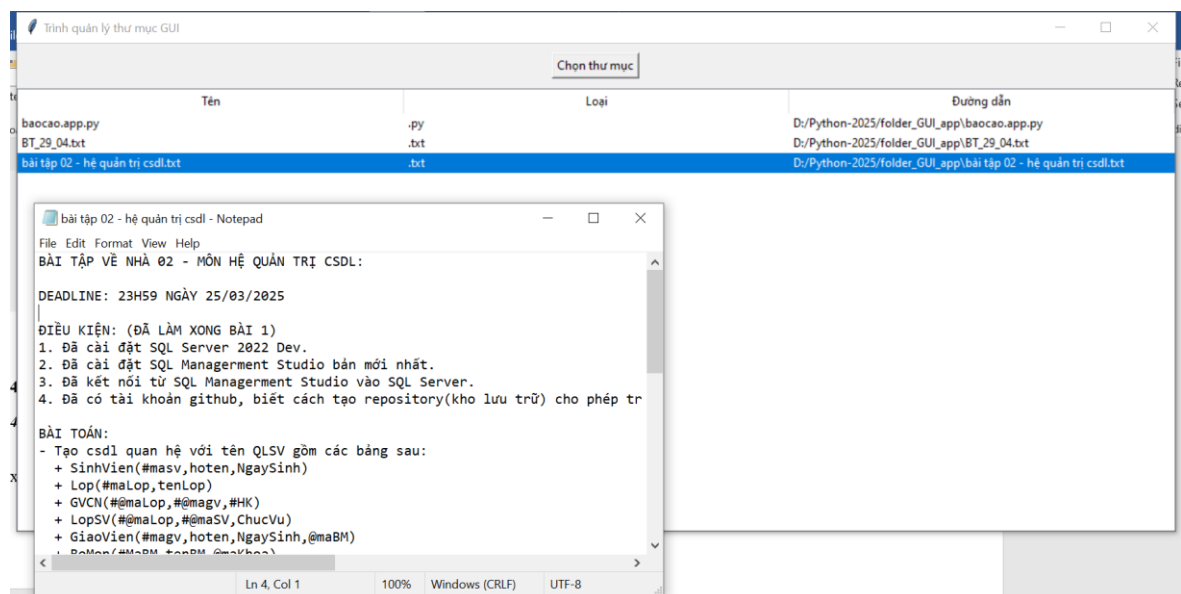
CHƯƠNG 4. THỰC NGHIỆM VÀ KẾT LUẬN

4.1. Thực nghiệm.

Trong quá trình thực nghiệm chương trình "Trình quản lý thư mục GUI", hệ thống đã hoạt động đúng như yêu cầu đặt ra. Sau khi người dùng nhấn nút "Chọn thư mục" và chọn thư mục chứa các tệp dữ liệu, giao diện đã hiển thị đầy đủ danh sách các tệp có phần mở rộng phù hợp (.txt, .py). đây là kết quả khi em chạy chương trình:



Hình 4.1. Giao diện thư mục GUI khi đã chọn ra các file có đuôi .py, .txt



Hình 4.2. Click vào 1 file bất kì đã chọn lọc và hiển thị nội dung ra notepad

4.2. Kết luận.

a) Sản phẩm đã làm được những gì?

Giao diện người dùng được thiết kế đơn giản với các thành phần rõ ràng:

- Nút "Chọn thư mục"
- Bảng Treeview hiển thị danh sách file theo cột: Tên, Loại, Đường dẫn.
- Chức năng chọn thư mục hoạt động chính xác
- Lọc và hiển thị đúng các tệp yêu cầu (.txt, .py, .jpg)
- Tự động làm mới dữ liệu khi chọn thư mục mới
- Mở file bằng ứng dụng mặc định thành công
- Bắt lỗi và hiển thị thông báo hợp lý

Sản phẩm đã đáp ứng đầy đủ các yêu cầu chức năng đề ra trong đầu bài. Giao diện thân thiện, xử lý chính xác và có khả năng phản hồi lỗi tốt. Đây là một ứng dụng GUI đơn giản nhưng hiệu quả, có thể mở rộng thêm các chức năng nâng cao trong tương lai như tìm kiếm, lọc nâng cao, hoặc hiển thị thông tin chi tiết của từng file.

b) Học được gì?

Kỹ năng làm việc với thư viện tkinter:

- Biết cách tạo giao diện người dùng (GUI) đơn giản bằng tkinter.
- Sử dụng các widget như Button, Treeview, filedialog, messagebox một cách hiệu quả.

Làm việc với hệ thống tệp (os module):

- Cách duyệt danh sách file trong thư mục với `os.listdir()` và `os.path`.
- Cách ghép đường dẫn, lấy phần mở rộng file.

Xử lý sự kiện và dữ liệu giao diện:

- Biết cách xử lý sự kiện click chuột trong Treeview.
- Cập nhật nội dung bảng động mỗi lần người dùng chọn thư mục mới.

Kỹ thuật xử lý lỗi (Exception Handling):

- Biết bao bọc các đoạn code dễ sinh lỗi bằng try/except.
- Biết cách đưa ra thông báo lỗi hợp lý cho người dùng cuối.

Mở file bằng chương trình mặc định:

- Sử dụng `os.startfile()` để mở tệp theo cấu hình hệ điều hành.

c) Hướng cải tiến là gì?

Ý tưởng cải tiến	Mô tả cụ thể
Thêm chức năng tìm kiếm	Cho phép người dùng gõ tên file hoặc từ khóa để lọc file trong danh sách.
Hỗ trợ nhiều định dạng hơn	Hiện tại chỉ có .txt, .py, .jpg. Có thể mở rộng ra các loại như .pdf, .docx, .png, .csv, v.v.
Hiển thị thư mục con	Hiển thị luôn cả thư mục con (dạng cây – Treeview lồng nhau).
Thêm cột thông tin file	Bổ sung các trường như: kích thước file, ngày sửa đổi gần nhất.
Thông báo mở file thành công	Sau khi mở file, hiển thị thông báo “Đã mở thành công” hoặc biểu tượng xác nhận.
Giao diện nâng cao (Dark mode)	Tùy chọn giao diện tối cho người dùng thích trải nghiệm hiện đại.
Cập nhật tự động khi thư mục thay đổi	Theo dõi thư mục và tự động cập nhật nếu có file mới.
Đóng gói thành phần mềm chạy độc lập (.exe)	Sử dụng pyinstaller để tạo ứng dụng độc lập cho người dùng cuối.

Bảng 4.3. Hướng cải tiến trong tương lai.

TỔNG KẾT

Sau quá trình nghiên cứu và triển khai, đề tài “Trình quản lý thư mục GUI” đã hoàn thành với kết quả đáp ứng đúng yêu cầu đặt ra. Ứng dụng được xây dựng bằng ngôn ngữ Python với thư viện tkinter để tạo giao diện đồ họa thân thiện, giúp người dùng có thể chọn thư mục bất kỳ trên hệ thống, liệt kê các tệp tin thuộc các định dạng .txt, .py, và .jpg, đồng thời cho phép mở tệp bằng chương trình mặc định chỉ với một thao tác click đúp. Chương trình bao gồm ba hàm chính: `chon_thu_muc()` để xử lý chọn thư mục, `hien_thi_file()` để quét và hiển thị danh sách file, và `mo_file()` để xử lý sự kiện mở file. Giao diện được thiết kế tối giản với một nút chọn thư mục và bảng Treeview hiển thị đầy đủ thông tin về tên file, loại file và đường dẫn tuyệt đối.

Qua quá trình thực hiện, ứng dụng đã được kiểm thử thực tế với các thư mục chứa file .txt, .py, và .jpg và cho kết quả chính xác: hiển thị đúng số lượng file, đúng định dạng và mở được tệp thành công. Nhờ đó, người thực hiện đề tài đã rèn luyện được nhiều kỹ năng như: sử dụng tkinter để thiết kế GUI, làm việc với hệ thống file qua os, và xử lý sự kiện người dùng trong giao diện.

Tuy chương trình đã đáp ứng mục tiêu ban đầu, nhưng vẫn còn nhiều hướng cải tiến có thể thực hiện trong tương lai như: mở rộng hỗ trợ nhiều định dạng tệp hơn, bổ sung chức năng tìm kiếm, hiển thị thêm các thông tin về kích thước và thời gian chỉnh sửa file, cải tiến giao diện (ví dụ: thêm chế độ nền tối), hoặc đóng gói ứng dụng thành file .exe để thuận tiện khi sử dụng. Tổng thể, đây là một bài tập thực hành hiệu quả, giúp nâng cao tư duy thiết kế phần mềm, khả năng lập trình hướng sự kiện và tương tác hệ thống của người học.

TÀI LIỆU THAM KHẢO

https://r.search.yahoo.com/_ylt=AwrO5rpdMkdoHbgTje5XNyoA;_ylu=Y29sbwNncTEEcG9zAzMEdnRpZAMEc2VjA3Ny/RV=2/RE=1750706013/RO=10/RU=https%3a%2f%2fdocs.python.org%2f3%2ftutorial%2findex.html/RK=2/RS=ycQ63mXpGTllfmgEJjyOXDAEbB8-

Giáo trình: Python® Programming for the Absolute Beginner, Third Edition

https://r.search.yahoo.com/_ylt=AwrjZWTdMkdod.4QEytXNyoA;_ylu=Y29sbwNncTEEcG9zAzEEdnRpZAMEc2VjA3Ny/RV=2/RE=1750706141/RO=10/RU=https%3a%2f%2ftoidicode.com%2fpython-co-ban/RK=2/RS=BNawzc7EoSsbKI3dJKuD8fuQei0-

https://r.search.yahoo.com/_ylt=AwrjZWTdMkdod.4QLStXNyoA;_ylu=Y29sbwNncTEEcG9zAzcEdnRpZAMEc2VjA3Ny/RV=2/RE=1750706141/RO=10/RU=https%3a%2f%2fhoc.hocpython.org%2fcourse%2fpython-can-ban/RK=2/RS=yji.0Gh59X9zwG.tnXgb0uI3CFU-