

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ

Bộ môn: Công nghệ Thông tin.

BÀI TẬP KẾT THÚC MÔN HỌC

MÔN HỌC
LẬP TRÌNH PYTHON

Sinh viên: **VŨ ĐỨC TÚ**

Lớp: **K58.KTP**

Giáo viên **GIẢNG DẠY**: **TS. NGUYỄN VĂN HUY**

Link GitHub:

Link youtube: <https://www.youtube.com/watch?v=xgz11GeKm0o>

Thái Nguyên – 2025

TRƯỜNG ĐHKTCN
KHOA ĐIỆN TỬ

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÀI TẬP KẾT THÚC MÔN HỌC

MÔN HỌC: LẬP TRÌNH PYTHON

BỘ MÔN : CÔNG NGHỆ THÔNG TIN

Sinh viên: VŨ ĐỨC TÚ

Lớp: K58.KTP

Ngành: Kỹ thuật phần mềm

Giáo viên hướng dẫn: TS NGUYỄN VĂN HUY

Ngày giao đề: 20/05/2025

Ngày hoàn thành: 09/06/2025

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 : Đầu vào – đầu ra:

- *Đầu vào: Nút “Chọn thư mục”.*
- *Đầu ra: Treeview hoặc Listbox hiển thị file, nút “Mở”.*

Tính năng 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.*

Kiểm tra & kết quả mẫu:

- *Chọn thư mục có 3 file → hiển thị 3 dòng.*
- *Click “Mở” file .txt → mở Notepad.*

Các bước triển khai:

1. *Frame có nút và Treeview.*
2. *Dialog chọn folder (askdirectory).*
3. *Hàm show_files() scan và fill Treeview.*
4. *Map double-click mở file (os.startfile).*

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, ngày....tháng.....năm 20....

GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

MỤC LỤC

LỜI MỞ ĐẦU.....	3
CHƯƠNG I. GIỚI THIỆU ĐỀ TÀI.....	4
1.1. Giới thiệu đề tài.....	4
1.2. Mục tiêu của đề tài.....	4
1.3. Tính năng và thách thức.....	4
1.4. Các kiến thức vận dụng để làm đề tài.....	5
CHƯƠNG II. CƠ SỞ LÝ THUYẾT.....	7
2.1. Giới thiệu về ngôn ngữ lập trình python.....	7
2.2. Thư viện Tkinter.....	7
2.3. Làm việc với hệ thống file trong python.....	7
2.4. Mở file bằng chương trình mặc định.....	7
2.5. Lập trình hướng đối tượng (oop) trong Python.....	7
2.6. Cấu trúc dữ liệu cơ bản.....	8
2.7. Widget treeview trong Tkinter.....	9
2.8. Xử lý sự kiện và ngoại lệ.....	9
CHƯƠNG III. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH.....	10
3.1. Sơ đồ khối hệ thống.....	10
3.1.1. Các module chính trong chương trình.....	10
3.2. Biểu đồ phân cấp chức năng.....	11
3.3. Cấu trúc dữ liệu.....	12
3.4 Chương Trình.....	13
CHƯƠNG IV. THỰC NGHIỆM VÀ KẾT LUẬN.....	16
4.1. Thực nghiệm.....	16
4.1.1. Kiểm tra chức năng chọn thư mục.....	16
4.1.2. Kiểm tra hiển thị danh sách file.....	17
4.1.3. Kiểm tra chức năng mở file bằng nút “ Mở “.....	17
4.1.4 Kiểm tra mở file bằng double-click.....	19
4.1.5 Kiểm tra báo lỗi.....	19

4.2 Kết luận.....	21
-------------------	----

LỜI MỞ ĐẦU

Trong thời đại công nghệ số hiện nay, việc quản lý tệp tin trên máy tính một cách khoa học và hiệu quả đóng vai trò quan trọng trong học tập, làm việc và phát triển phần mềm. Tuy nhiên, người dùng thường gặp khó khăn trong việc tìm kiếm và truy cập nhanh các tệp tin có định dạng cụ thể trong một thư mục lớn. Xuất phát từ thực tế đó, 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” được xây dựng nhằm hỗ trợ người dùng quản lý tệp tin một cách trực quan, dễ dàng và tiện lợi.

Ứng dụng được thiết kế với giao diện đồ họa (GUI) thân thiện, cho phép người dùng chọn thư mục bất kỳ, hiển thị danh sách các tệp tin theo định dạng được chỉ định, đồng thời hỗ trợ mở trực tiếp tệp từ giao diện. Đây không chỉ là một công cụ hữu ích cho người dùng phổ thông, mà còn giúp sinh viên, lập trình viên tiết kiệm thời gian khi thao tác với nhiều loại tệp khác nhau. Qua quá trình thực hiện đề tài, sinh viên có cơ hội củng cố kiến thức về lập trình Python, xử lý tệp tin, thiết kế giao diện GUI và làm quen với quy trình phát triển phần mềm ứng dụng thực tế.

CHƯƠNG I. GIỚI THIỆU ĐỀ TÀI

1.1. Giới thiệu 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” nhằm xây dựng một chương trình có giao diện đồ họa thân thiện giúp người dùng dễ dàng quản lý và mở các tệp tin trong một thư mục bất kỳ. Ứng dụng được lập trình bằng ngôn ngữ Python, sử dụng thư viện tkinter để tạo giao diện, kết hợp với os, subprocess và filedialog để thao tác với hệ thống tệp. Chương trình cho phép người dùng chọn thư mục, tự động lọc và hiển thị các file có định dạng cụ thể như .txt, .py, .jpg dưới dạng bảng, đồng thời hỗ trợ mở file bằng phần mềm mặc định hoặc bằng IDLE đối với file .py. Trong quá trình thực hiện, thách thức lớn nhất là việc làm chủ cách sử dụng Treeview để hiển thị dữ liệu và xử lý tương tác như double-click. Đề tài giúp vận dụng tổng hợp kiến thức về lập trình Python, xử lý file, và phát triển giao diện người dùng.

1.2. Mục tiêu của đề tài

Mục tiêu của đề tài là xây dựng một ứng dụng giao diện đồ họa (GUI) bằng Python cho phép người dùng chọn thư mục trên máy tính, sau đó tự động quét và liệt kê các tệp tin có định dạng .txt, .py, và .jpg trong thư mục đó. Danh sách các file sẽ được hiển thị rõ ràng trên bảng Treeview. Người dùng có thể mở file bằng cách nhấn nút “Mở” hoặc double-click vào dòng tương ứng. Chương trình cần đảm bảo xử lý lỗi khi không tìm thấy đường dẫn, sử dụng thư viện os để quét thư mục và tkinter để xây dựng giao diện, đáp ứng đúng các yêu cầu về chức năng và tính tương tác với người dùng.

1.3. Tính năng và thách thức

*Tính năng của đề tài:

- Giao diện người dùng (GUI) trực quan được xây dựng bằng thư viện tkinter.
- Cho phép người dùng chọn thư mục thông qua hộp thoại chọn đường dẫn (filedialog.askdirectory()).
- Tự động quét thư mục và lọc các file theo định dạng cụ thể: .txt, .py, .jpg.

Hiển thị danh sách file theo từng cột: tên file, loại file và đường dẫn, sử dụng bảng Treeview.

-Cho phép mở file đã chọn bằng phần mềm mặc định của hệ điều hành hoặc mở file .py bằng IDLE.

-Hỗ trợ thao tác mở file bằng cách click nút hoặc double-click vào dòng trong danh sách.

-Xử lý lỗi khi đường dẫn không hợp lệ hoặc không thể mở file.

*Thách thức của đề tài:

-Làm quen và sử dụng hiệu quả thư viện tkinter và ttk.Treeview để xây dựng giao diện hiển thị dữ liệu có thể tương tác.

-Xử lý logic để lọc đúng các định dạng file theo yêu cầu trong quá trình quét thư mục.

-Đảm bảo chương trình có thể hoạt động linh hoạt trên nhiều hệ điều hành, đặc biệt trong việc mở file bằng phần mềm mặc định (os.startfile, subprocess.Popen).

-Quản lý lỗi người dùng như không chọn thư mục, file bị xóa hoặc không thể mở được, đảm bảo chương trình không bị gián đoạn.

-Thiết kế giao diện đơn giản nhưng đầy đủ chức năng, đảm bảo tính dễ sử dụng cho người dùng cuối.

1.4. Các kiến thức vận dụng để làm đề tài

* Lập trình Python căn bản và nâng cao:

- Sử dụng biến, hàm, cấu trúc điều kiện, vòng lặp.
- Quản lý file và thư mục bằng các thư viện tiêu chuẩn như os.

* Xử lý file và thư mục:

- Duyệt nội dung thư mục bằng os.listdir() và os.path.
- Phân tích và lọc định dạng file bằng os.path.splitext().

*Xây dựng giao diện người dùng với tkinter:

- Tạo cửa sổ chính, khung chứa, nút bấm (Button), bảng (Treeview), và hộp thoại chọn thư mục (filedialog.askdirectory()).
- Tùy chỉnh các thành phần giao diện và xử lý sự kiện người dùng như nhấn nút, double-click.

*Tương tác với hệ thống:

- Mở file bằng phần mềm mặc định qua os.startfile() (Windows) hoặc subprocess.Popen() để chạy file .py bằng IDLE.

* Xử lý lỗi và tương tác người dùng:

- Kiểm tra đường dẫn hợp lệ, bắt lỗi khi không thể mở file hoặc không tìm thấy thư mục.
- Sử dụng messagebox để thông báo lỗi hoặc cảnh báo.

* Tư duy phân tích và thiết kế phần mềm:

- Xác định rõ đầu vào – đầu ra, luồng xử lý dữ liệu.
- Thiết kế giao diện hợp lý, dễ sử dụng và đảm bảo tính tương tác

Tóm tắt chương

Chương này giới thiệu ứng dụng GUI viết bằng Python cho phép người dùng chọn thư mục, liệt kê các file .txt, .py, .jpg và mở file bằng phần mềm mặc định. Ứng dụng sử dụng tkinter để tạo giao diện và os để xử lý thư mục, giúp người học rèn luyện kỹ năng lập trình, xử lý file và thiết kế giao diện đồ họa.

CHƯƠNG II. CƠ SỞ LÝ THUYẾT

2.1. Giới thiệu về ngôn ngữ lập trình python

Python là ngôn ngữ lập trình bậc cao, thông dịch, đơn giản và dễ đọc, rất phù hợp với việc phát triển nhanh các ứng dụng từ đơn giản đến phức tạp. Với thư viện chuẩn phong phú và cú pháp rõ ràng, Python được sử dụng rộng rãi trong nhiều lĩnh vực như xử lý dữ liệu, trí tuệ nhân tạo, và phát triển ứng dụng giao diện đồ họa.

2.2. Thư viện Tkinter

Tkinter là thư viện tiêu chuẩn của Python dùng để xây dựng giao diện người dùng đồ họa (GUI). Tkinter cung cấp các widget cơ bản như nút bấm (Button), khung chứa (Frame), hộp thoại chọn thư mục (filedialog), và các widget nâng cao như bảng dữ liệu (Treeview). Tkinter dễ sử dụng, phù hợp cho các ứng dụng GUI nhỏ và vừa.

2.3. Làm việc với hệ thống file trong python

Thư viện os của Python cung cấp các hàm tương tác với hệ điều hành, giúp truy xuất và xử lý file, thư mục như:

- `os.listdir(path)`: Liệt kê nội dung thư mục.
- `os.path.isfile(path)`: Kiểm tra một đường dẫn có phải là file không.
- `os.path.splitext(filename)`: Tách phần tên và phần mở rộng của file.

Các hàm này hỗ trợ ứng dụng trong việc quét thư mục và lọc file theo loại.

2.4. Mở file bằng chương trình mặc định

Trên Windows, hàm `os.startfile(filepath)` giúp mở file bằng ứng dụng mặc định tương ứng với loại file.

Đối với file Python (.py), có thể sử dụng `subprocess.Popen` để mở file trong môi trường IDLE, giúp thuận tiện chỉnh sửa và chạy file.

2.5. Lập trình hướng đối tượng (oop) trong Python

Lập trình hướng đối tượng giúp tổ chức mã nguồn thành các lớp và đối tượng, tăng tính tái sử dụng và dễ bảo trì. Các khái niệm chính gồm:

- Lớp (Class): Khuôn mẫu định nghĩa thuộc tính và phương thức.
- Đối tượng (Object): Thể hiện cụ thể của lớp.
- Đóng gói (Encapsulation): Ẩn thông tin bên trong đối tượng.
- Kế thừa (Inheritance) và Đa hình (Polymorphism): Các tính năng nâng cao giúp mở rộng chức năng.

Trong đề tài này, ứng dụng GUI được thiết kế theo hướng đối tượng giúp tách biệt các chức năng, dễ dàng mở rộng và sửa đổi.

2.6. Cấu trúc dữ liệu cơ bản

*Danh sách file (List)

- Ứng dụng sử dụng kiểu dữ liệu danh sách (list) để lưu trữ tên các file lấy từ thư mục được chọn.
- Mỗi phần tử trong danh sách là tên file dạng chuỗi (string).
- Danh sách này giúp duyệt và xử lý từng file để lọc theo loại và hiển thị ra giao diện.

*Tuple chứa thông tin file

- Mỗi file được đại diện dưới dạng một tuple chứa các thông tin:
 - Tên file (str)
 - Phần mở rộng file (loại file) (str)
 - Đường dẫn đầy đủ đến file (str)

Ví dụ: ("document.txt", ".txt", "C:/Users/User/Documents/document.txt")

- Tuple này được sử dụng để đưa vào bảng Treeview, giúp người dùng dễ dàng xem và thao tác.

*Widget Treeview (Bảng dữ liệu)

- Widget Treeview sử dụng cấu trúc bảng gồm các cột và hàng để hiển thị danh sách file.
- Mỗi dòng trong Treeview tương ứng với một tuple file, chứa các cột: tên file, loại file, đường dẫn.

- Treeview hỗ trợ các thao tác chọn dòng, double-click mở file.

***Các biến trạng thái**

- Biến lưu giữ thư mục hiện tại được chọn để tiện thao tác.
- Biến lưu trạng thái file được chọn trong Treeview phục vụ thao tác mở file.

2.7. Widget treeview trong Tkinter

Treeview thuộc thư viện tkinter.ttk là widget dạng bảng, cho phép hiển thị dữ liệu có cấu trúc dạng cột, hàng. Ứng dụng sử dụng Treeview để liệt kê danh sách file với các thông tin: tên file, loại file và đường dẫn đầy đủ.

2.8. Xử lý sự kiện và ngoại lệ

Ứng dụng sử dụng xử lý sự kiện người dùng như:

- Bấm nút để chọn thư mục và mở file.
- Double-click trên dòng file để mở nhanh.

Bên cạnh đó, chương trình cũng bắt lỗi phổ biến như:

- Người dùng không chọn thư mục.
- Không thể mở file (file bị khóa, đường dẫn sai).
- Thư mục không tồn tại.

Việc xử lý ngoại lệ giúp ứng dụng hoạt động ổn định, thân thiện với người dùng.

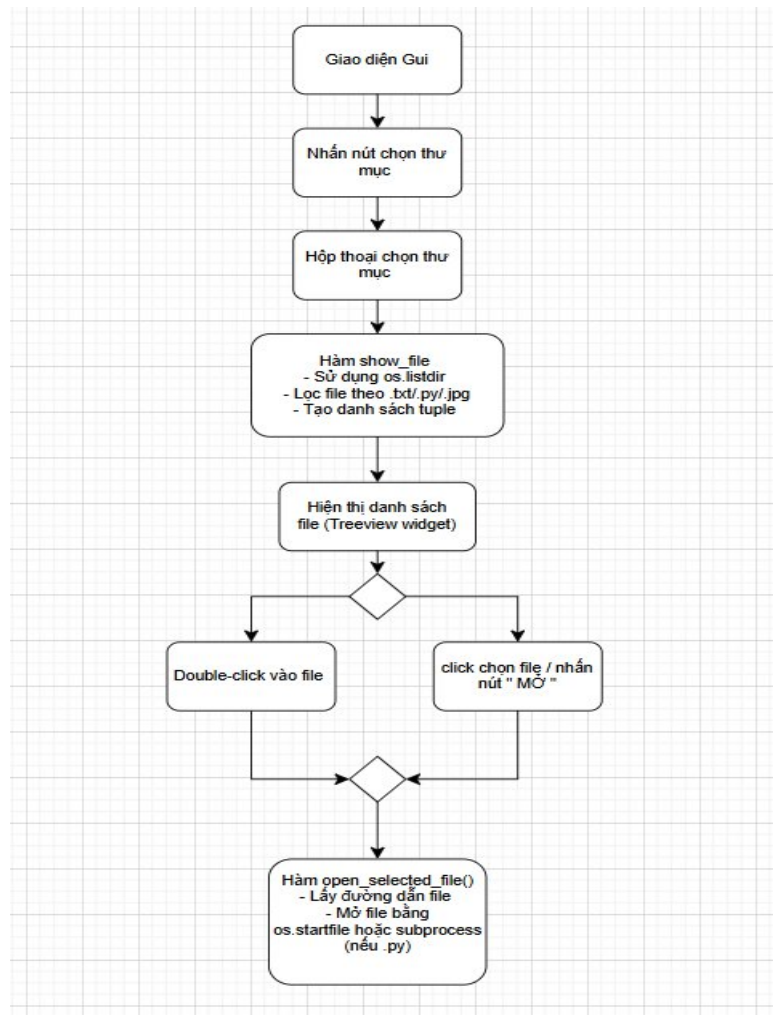
Tóm tắt chương

Chương 2 trình bày các kiến thức nền tảng và cơ sở lý thuyết liên quan đến việc xây dựng ứng dụng GUI chọn thư mục, liệt kê và mở file. Nội dung bao gồm ngôn ngữ lập trình Python, thư viện Tkinter dùng để thiết kế giao diện, cách thao tác với hệ thống file qua thư viện os, và phương pháp mở file bằng chương trình mặc định. Ngoài ra, chương cũng đề cập đến lập trình hướng đối tượng để tổ chức mã nguồn hiệu quả, cấu trúc dữ liệu cơ bản như danh sách và tuple để lưu trữ thông tin file, cùng việc xử lý sự kiện và ngoại lệ trong ứng dụng. Những kiến

thức này là nền tảng giúp phát triển ứng dụng dễ sử dụng, ổn định và đáp ứng yêu cầu đề tài.

CHƯƠNG III. THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

3.1. Sơ đồ khối hệ thống



Hình 3.1 Sơ đồ khối hệ thống

3.1.1. Các module chính trong chương trình

* Giao diện GUI (Tkinter)

- Tạo cửa sổ chính.
- Chứa các nút bấm như “Chọn thư mục”, “Mở”.
- Giao diện chứa bảng danh sách file sử dụng Treeview.

* Hàm show_files()

- Hiện thị hộp thoại để người dùng chọn thư mục.
- Duyệt các file trong thư mục bằng os.listdir().
- Lọc file theo đuôi .txt, .py, .jpg.
- Đưa danh sách file vào Treeview.

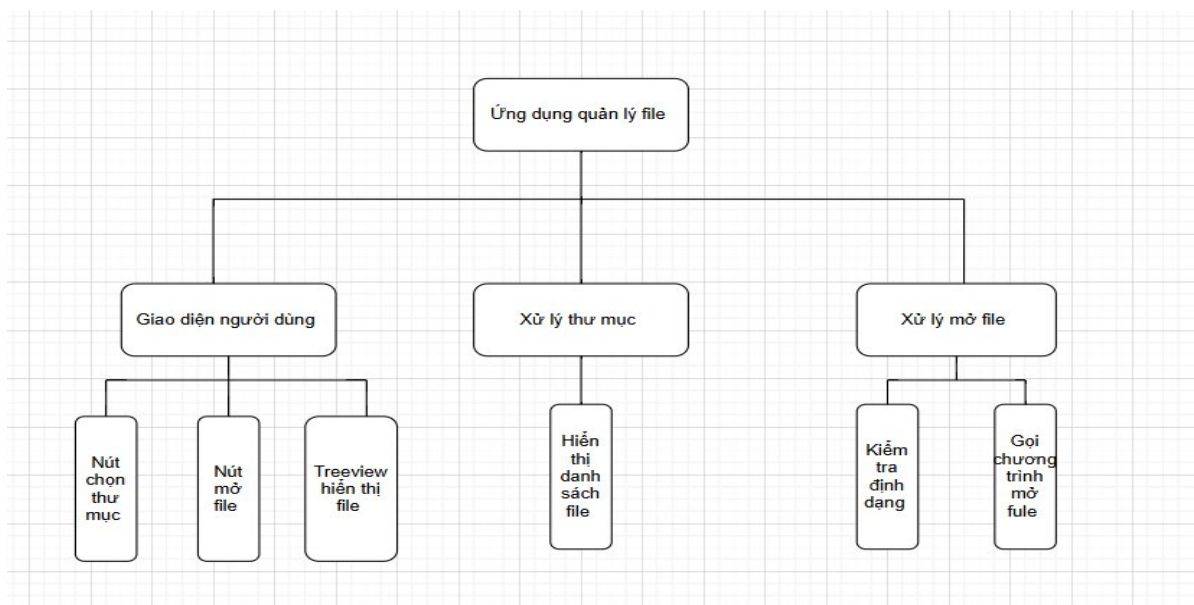
* Hàm open_selected_file()

- Lấy đường dẫn file được chọn.
- Kiểm tra đuôi file.
- Mở bằng chương trình mặc định:
 - .py: mở bằng IDLE (subprocess).
 - Các file khác: mở bằng os.startfile().

* Xử lý tương tác người dùng

- Double-click vào file sẽ gọi open_selected_file().
- Click nút “Mở” cũng gọi open_selected_file().

3.2. Biểu đồ phân cấp chức năng



Hình 3.2 Biểu đồ phân cấp chức năng

3.3. Cấu trúc dữ liệu

Trong chương trình quản lý và mở file từ thư mục, cấu trúc dữ liệu không sử dụng cơ sở dữ liệu theo dạng bảng như SQL, mà chủ yếu sử dụng các cấu trúc dữ liệu cơ bản trong Python để lưu trữ và quản lý thông tin. Dưới đây là mô tả các cấu trúc dữ liệu được sử dụng:

1. Danh sách file (List of Tuples)

- Được dùng để lưu danh sách các file hợp lệ được lọc từ thư mục người dùng chọn.
 - Mỗi phần tử trong danh sách là một tuple gồm tên file và đường dẫn file.
- *Cấu trúc: “file_list = [(ten_file1, duong_dan1), (ten_file2, duong_dan2), ...]”

Ý nghĩa:

- ten_file: Tên file để hiển thị trong giao diện.
- duong_dan: Đường dẫn tuyệt đối của file, dùng để mở bằng chương trình mặc định.

2. Biến chuỗi đường dẫn thư mục (String)

- Lưu trữ đường dẫn thư mục được chọn từ hộp thoại askdirectory().

Ví dụ: “folder_path = “C:/Users/Admin/Documents””

3. Treeview (tkinter.ttk.Treeview)

- Là widget được sử dụng để hiển thị danh sách file trên giao diện người dùng.
- Dữ liệu trong Treeview được nạp từ danh sách file_list.
- Người dùng có thể click chọn hoặc double-click vào file trong Treeview để mở.

4. Lựa chọn file từ Treeview

- Sử dụng phương thức `selection()` của Treeview để xác định file mà người dùng đang chọn.
- Ví dụ: “ `selected_item = tree.selection()`
`file_info = tree.item(selected_item, 'values')`”
- `file_info` sẽ chứa tên và đường dẫn file, giúp xử lý thao tác mở file.

3.4 Chương Trình

Chương trình được xây dựng bằng ngôn ngữ Python, sử dụng thư viện tkinter để tạo giao diện người dùng (GUI), và các module `os`, `subprocess`, `filedialog`, `messagebox` để thao tác với hệ thống file. Dưới đây là mô tả chi tiết các hàm chính trong chương trình:

1. Hàm `show_files()`

Chức năng: Hiển thị danh sách các file trong thư mục mà người dùng chọn.

Chi tiết xử lý:

- Mở hộp thoại để chọn thư mục với `filedialog.askdirectory()`.
- Sử dụng `os.listdir()` để liệt kê tất cả các file trong thư mục.
- Kiểm tra và lọc các file có đuôi mở rộng thuộc các loại: `.txt`, `.py`, `.jpg`.
- Kết quả được hiển thị lên Treeview trong giao diện người dùng.

*Code:

```
def show_files():
    folder = filedialog.askdirectory()
    if not folder:
        return
    ...
    for file in files:
        ...
        if ext in ['.txt', '.py', '.jpg']:
```

```
tree.insert("", "end", values=(file, ext, filepath))
```

2. Hàm open_selected_file()

Chức năng: Mở file mà người dùng đã chọn trong danh sách.

Chi tiết xử lý:

- Lấy file đang được chọn trong Treeview.
- Kiểm tra phần mở rộng của file:
 - Nếu là .py: mở bằng IDLE sử dụng subprocess.
 - Nếu là các loại file khác: mở bằng ứng dụng mặc định với `os.startfile()` (chỉ hoạt động trên Windows).
- Xử lý ngoại lệ và thông báo lỗi nếu không mở được file.

*Code:

```
def open_selected_file():  
    selected = tree.focus()  
    ...  
    if ext == ".py":  
        subprocess.Popen([sys.executable, "-m", "idlelib", filepath])  
    else:  
        os.startfile(filepath)
```

3. Hàm on_double_click(event)

- Chức năng: Bắt sự kiện khi người dùng double-click vào một file trong danh sách, và gọi hàm open_selected_file() để mở file tương ứng.

*Code: `def on_double_click(event):`

`open_selected_file()`

4. Khởi tạo giao diện chính

- Tạo cửa sổ chính với `tk.Tk()` và cấu hình tiêu đề.
- Tạo các nút chức năng:
- Chọn thư mục: gọi hàm `show_files()`.
- Mở: gọi hàm `open_selected_file()`.

Hiển thị danh sách file bằng widget `ttk.Treeview` với 3 cột: Tên file, Loại, Đường dẫn.

Tóm tắt chương

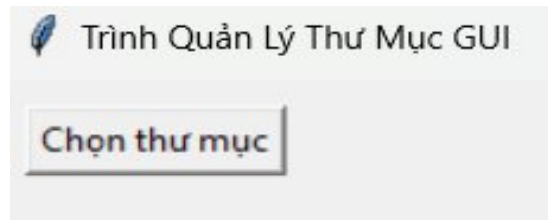
Chương này trình bày quá trình thiết kế và xây dựng chương trình quản lý thư mục với giao diện đồ họa sử dụng Python và thư viện Tkinter. Chương mô tả các module chính như chọn thư mục, hiển thị danh sách file theo định dạng (.txt, .py, .jpg) và mở file bằng chương trình mặc định. Các sơ đồ khối và biểu đồ phân cấp chức năng thể hiện rõ luồng xử lý và các chức năng chính của hệ thống. Dữ liệu được tổ chức dưới dạng bảng trong Treeview, lưu thông tin tên file, loại và đường dẫn. Cuối cùng, chương trình gồm các hàm chính như `show_files`, `open_selected_file` và `on_double_click` để xử lý thao tác người dùng một cách trực quan và hiệu quả.

CHƯƠNG IV. THỰC NGHIỆM VÀ KẾT LUẬN

4.1. Thực nghiệm

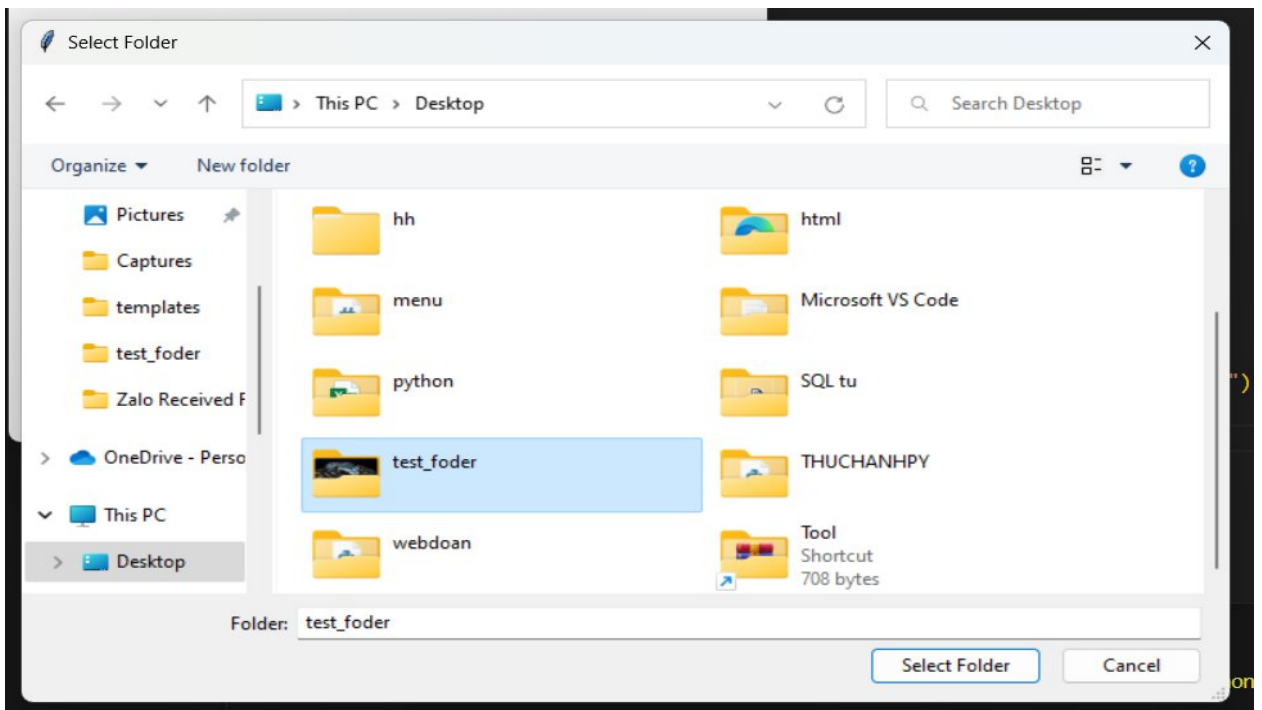
Để kiểm tra tính đúng đắn và hiệu quả của chương trình "Trình Quản Lý Thư Mục GUI", nhóm đã tiến hành thực nghiệm thông qua các bài kiểm tra chức năng chính của phần mềm.

4.1.1. Kiểm tra chức năng chọn thư mục.



Hình 4.1 Nút chọn thư mục

Người dùng nhấn nút "Chọn thư mục", một hộp thoại mở ra cho phép duyệt đến thư mục mong muốn.

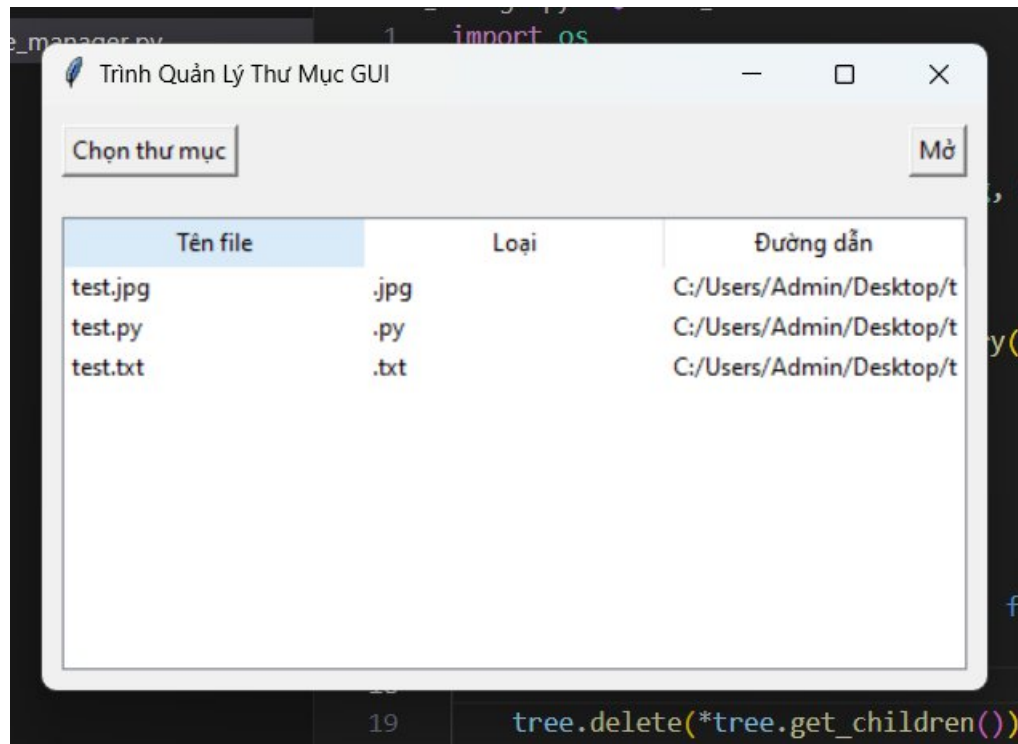


Hình 4.2 Hộp thoại chọn thư mục

Kết quả: Thư mục được chọn hiển thị danh sách các tập tin có định dạng .txt, .py, .jpg trong bảng hiển thị.

4.1.2. Kiểm tra hiển thị danh sách file

Chương trình sử dụng `os.listdir()` để lấy danh sách các file, lọc ra các định dạng cần thiết, và hiển thị trong Treeview.

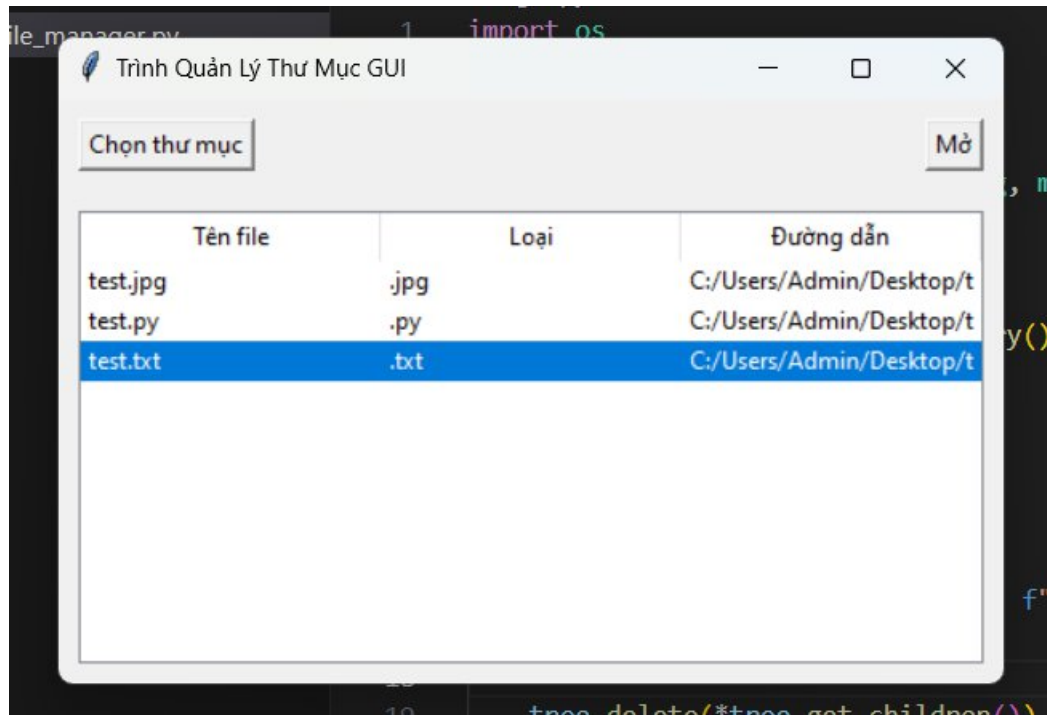


Hình 4.3 Các file của thư mục

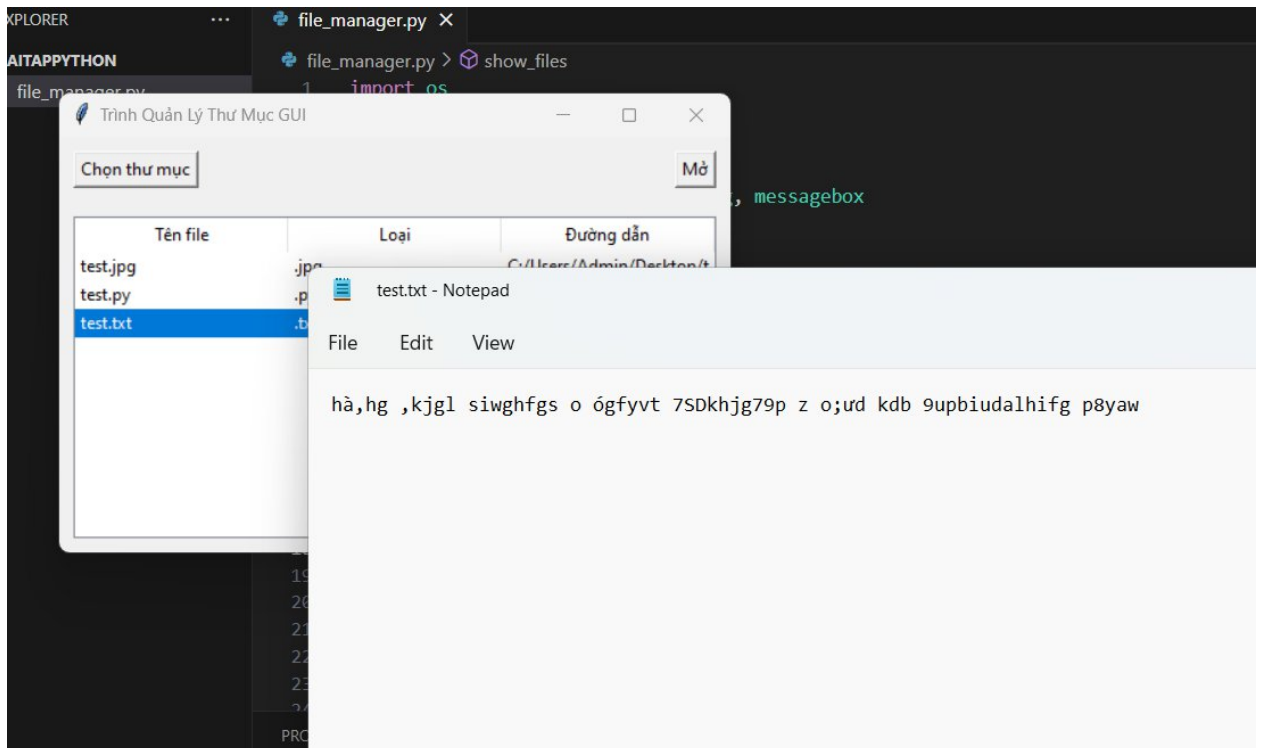
Kết quả: Danh sách file được hiển thị chính xác, gồm tên file, loại file, và đường dẫn đầy đủ.

4.1.3. Kiểm tra chức năng mở file bằng nút “ Mở “

Người dùng chọn một file từ danh sách và nhấn nút "Mở".



Hình 4.4 Chọn file để mở



Hình 4.5 Kết quả khi nhấn nút mở

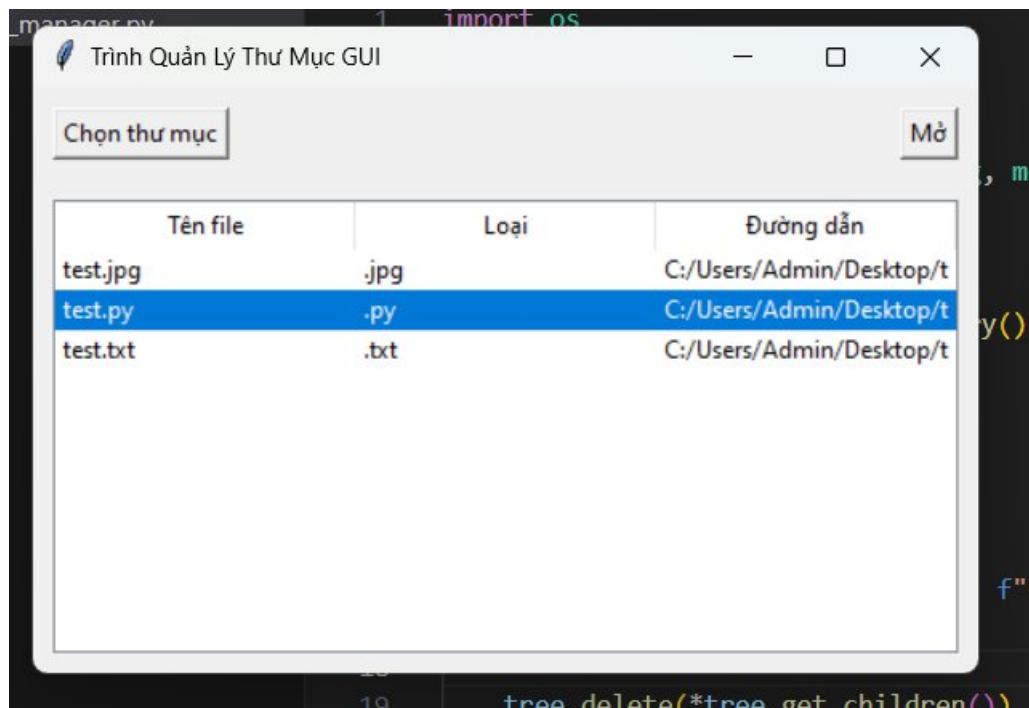
Kết quả:

Nếu là file .txt hoặc .jpg, hệ điều hành mở file bằng ứng dụng mặc định.

Nếu là file .py, chương trình mở file bằng trình soạn thảo Python IDLE.

4.1.4 Kiểm tra mở file bằng double-click

Người dùng double-click vào dòng file trong danh sách (người dùng sẽ click vào file 2 cái).

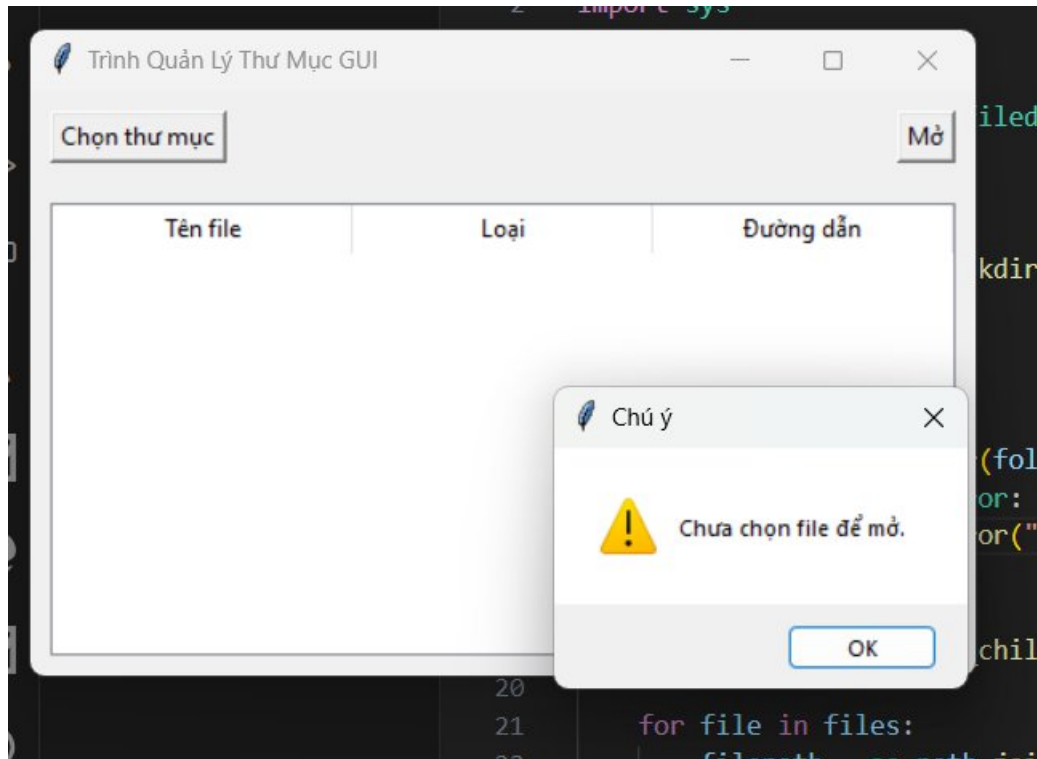


Hình 4.6 double-click file

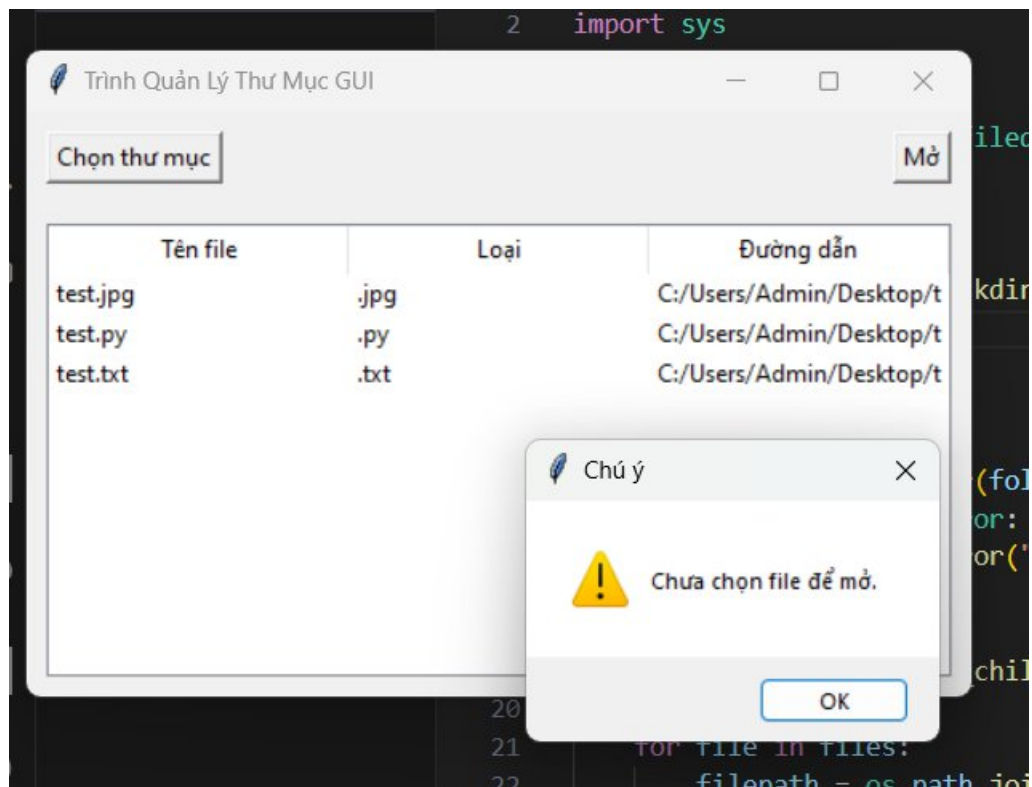
Kết quả: Chương trình tự động gọi hàm `open_selected_file()` để mở file tương ứng.

4.1.5 Kiểm tra báo lỗi

Kiểm tra báo lỗi khi người dùng chưa chọn thư mục hoặc chọn file mà lại đi nhấn nút mở khi đó sẽ hiển thị thông báo lỗi cảnh báo hoặc lỗi bằng messagebox.



Hình 4.7 Chưa chọn thư mục



Hình 4.8 Chưa chọn file để mở

4.2 Kết luận

- Sản phẩm đã hoàn thiện một ứng dụng giao diện đồ họa (GUI) bằng Python giúp người dùng quản lý thư mục một cách trực quan. Chương trình cho phép chọn thư mục, liệt kê các tệp có định dạng .txt, .py, .jpg, và hiển thị thông tin chi tiết như tên file, loại file và đường dẫn. Người dùng có thể mở các file trực tiếp từ giao diện, hỗ trợ mở bằng phần mềm tương ứng như IDLE hoặc chương trình mặc định. Toàn bộ hệ thống hoạt động ổn định, đáp ứng mục tiêu xây dựng một công cụ quản lý file đơn giản, tiện lợi và dễ sử dụng.

- Qua quá trình xây dựng ứng dụng, em đã học được cách sử dụng thư viện Tkinter để tạo giao diện người dùng trong Python, biết cách xử lý các sự kiện như chọn thư mục, nhấn nút, double-click, cũng như cách thao tác với Treeview để hiển thị danh sách dữ liệu. Bên cạnh đó, em còn rèn luyện được kỹ năng phân tích yêu cầu, thiết kế chương trình theo cấu trúc rõ ràng và lập trình theo hướng đối tượng.

- Trong các phiên bản tiếp theo, em dự định sẽ cải tiến sản phẩm bằng cách:

+ Bổ sung chức năng tìm kiếm và lọc file theo tên hoặc định dạng để người dùng dễ dàng tra cứu.

+ Thêm hỗ trợ nhiều loại file hơn, ví dụ như PDF, Excel, Word.

+ Hiển thị nội dung xem trước của file văn bản ngay trong giao diện mà không cần mở bằng phần mềm ngoài.

+ Tăng tính bảo mật và kiểm tra lỗi khi mở file, giúp chương trình hoạt động ổn định hơn.

+ Đồng thời, giao diện sẽ được thiết kế lại theo hướng hiện đại, trực quan hơn để nâng cao trải nghiệm người dùng.