MANAGING FILES AND DIRECTORIES USING PYTHON

A PROJECT REPORT

Submitted by

LOHITHRAMAN S [Reg No: RA2211042010036]

AKSHAY KHANNA T K [Reg No: RA2211042010057]

VISHNU K [Reg No: RA2211042010043]

Under the Guidance of

DR. SV. SHRI BHARATHI

Associate Professor, Department of Data Science and Business Systems

In partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

with a specialization in BUSINESS SYSTEMS

INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this B.Tech project report titled "**Managing Files and directories using Python**" is the bonafide work of Mr. Lohithraman S [Reg. No.: RA2211042010036],Mr. Akshay Khanna T K [Reg. No.RA2211042010057] and Mr. Vishnu K [Reg. No.RA2211042010043] who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

DR. SV. SHRI BHARATHI
SUPERVISOR
Associate Professor
Department of Data Science and Business Systems

DR. M. LAKSHMI
HEAD OF THE DEPARTMENT
Department of Data Science and Business Systems

Department of Data Science and Business Systems

SRM Institute of Science and Technology

Own Work Declaration Form

Degree/ Course : B.Tech in Computer Science and Engineering with a specialization in Business systems

Student Names : Lohithraman S, Akshay Khanna T K, Vishnu K

Registration Number: RA2211042010036, RA2211042010057, RA2211042010043

Title of Work : **Managing Files and directories using Python**

We hereby certify that this assessment compiles with the University’s Rules and Regulations relating to Academic misconduct and plagiarism, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is our own except where indicated, and that we have met the following conditions:

- Clearly references / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc.)
- Given the sources of all pictures, data etc. that are not my own

- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

LOHITHRAMAN S RA2211042010036	AKSHAY KHANNA T K RA2211042010057	VISHNU K RA2211042010043

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr. T.V.Gopal**, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr. M. Lakshmi,** Professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We register our immeasurable thanks to our Faculty Advisor, **Dr. G. Elangovan**, Assistant Professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Dr. SV. Shri Bharathi**, Assistant Professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under her mentorship. She provided us with the freedom and support to explore the research topics of our interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Data Science and Business Systems Department staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to thank parents, family members, and friends for their unconditional love, constant support, and encouragement.

Lohithraman S [RA2211042010036]

Akshay Khanna T K [RA2211042010057]

Vishnu K [RA2211042010043]

TABLE OF CONTENTS

1.	INTRODUCTION	1
2	OBJECTIVE	3
3	AIM OF THE PROJECT	5
4	CONCLUSION	7
5	REFERENCE	9
6	SOURCE CODE	10
7	OUTPUT	16

INTRODUCTION

Welcome to File Manager, an innovative open-source file and directory management tool meticulously crafted using the power of Python. In a world where efficient data organization is paramount, File Manager emerges as a versatile solution, offering a blend of simplicity, functionality, and collaboration.

Target Audience:
File Manager caters to a diverse audience, including developers seeking streamlined project management, system administrators aiming for robust security measures, and everyday users looking for an accessible yet powerful tool for their file organization needs. Whether you're a seasoned coder or a casual computer user, File Manager is designed to adapt to your requirements.

Key Features:

1. Open Source Collaboration:
- File Manager is not just a tool; it's a community-driven initiative. As an open-source project, it welcomes contributors from around the globe, fostering an ecosystem of shared knowledge and continuous improvement.

2. Graphical User Interface (GUI):
- Dive into a seamless user experience with File Manager's intuitive GUI. The graphical interface not only simplifies complex file operations but also invites users who prefer visual interaction. The GUI, implemented with the tkinter module, serves as a gateway to efficient file and directory management.

3. Productivity at its Core:
- Empower your workflow with the extensive capabilities of Python's os and shutil modules. File Manager optimizes file operations, allowing developers to focus on coding rather than managing files. System administrators benefit from a suite of tools that streamline tasks, enhancing overall productivity.

4. Enhanced Data Security:

- File Manager places a premium on the security of your data. With built-in measures, it ensures that your files and directories are protected against unauthorized access and potential data loss, providing peace of mind in an era where data security is non-negotiable.

5. Collaboration Tools:

- In a collaborative environment, File Manager excels. Multiple users can seamlessly interact with shared data, fostering teamwork and simplifying collaborative projects. Share, modify, and organize files collectively, enhancing overall efficiency.

Use Case Scenarios:

- Scenario 1: Developer's Delight:

- Picture a scenario where a developer, faced with a myriad of project files, effortlessly organizes, renames, and moves files using File Manager's GUI. The result? A tidy project structure, boosting code readability and developer satisfaction.

- Scenario 2: System Administrator's Toolbox:

- In the realm of system administration, File Manager becomes an indispensable tool. System administrators navigate through directories, secure sensitive data, and execute routine tasks efficiently, all within a unified and user-friendly environment.

Vision and Goals:

The vision driving File Manager is to redefine the landscape of file and directory management. Our goal is not only to provide a powerful and user-friendly tool but to create a community where users actively contribute, shaping the future of File Manager. We envision a world where efficient data organization is not just a necessity but an enjoyable and collaborative experience.

Technology Stack:

File Manager harnesses the capabilities of Python, utilizing the os, shutil, and tkinter modules to create a robust and feature-rich environment. This technology stack ensures that File Manager remains adaptable, scalable, and at the forefront of file and directory management solutions.

OBJECTIVE

The overarching objective of File Manager is to revolutionize the landscape of file and directory management, presenting a powerful, open-source solution crafted with Python. Our project is driven by a multi-faceted approach to address the intricate needs of users ranging from developers and system administrators to everyday computer users. File Manager seeks to establish itself as a versatile and user-friendly tool that not only simplifies file operations but also promotes collaboration, enhances productivity, and prioritizes data security.

Specific Objectives:

1. **Intuitive Graphical User Interface (GUI):**
   Develop a sophisticated GUI for File Manager to provide users with a visually intuitive platform for managing files and directories. The goal is to create an interface that is not only user-friendly but also aesthetically pleasing, making file operations accessible to users of all proficiency levels.

2. **Streamlined File and Directory Operations:**
   Engineer File Manager to streamline common file and directory operations. From basic tasks like file creation and deletion to more complex operations like bulk file movement and renaming, the project aims to simplify these processes, reducing the cognitive load on users.

3. **Collaborative Features:**
   Implement collaborative tools to enhance teamwork and project collaboration. File Manager strives to be more than just an individual file management tool; it aims to facilitate seamless collaboration by providing features that enable multiple users to interact with shared data effectively.

4. **Integration of Python Modules:**
   Leverage the robust capabilities of Python's os and shutil modules to optimize file operations. File Manager seeks to empower developers with a familiar and powerful programming language, allowing them to customize and extend the functionality of the tool to suit their specific needs.

5. **Data Security Measures:**
   Prioritize the security of user data by implementing robust measures within File Manager. This includes access controls, encryption options, and safeguards against data loss. The project is committed to ensuring the confidentiality and integrity of files and directories.

6. **Community-Driven Development:**
   Cultivate a thriving open-source community around File Manager. Actively encourage collaboration, feedback, and contributions from a diverse group of developers and users. The project aims to evolve through collective efforts, ensuring continuous improvement and relevance.

7. **Comprehensive Documentation and User Support:**
   Develop detailed documentation to guide users and contributors through File Manager's features and functionalities. Ongoing user support will be provided to address queries and issues, ensuring a positive and user-friendly experience.

8. **Cross-Platform Compatibility:**
   Ensure File Manager's compatibility with various operating systems, promoting accessibility and usability across different computing environments. The project aims to eliminate barriers to entry, making it accessible to users regardless of their chosen platform.

9. **Customization and Adaptability:**
   Design File Manager to be adaptable to diverse user needs. Whether utilized by developers for project organization, system administrators for routine tasks, or everyday users for personal file management, the project aims to offer customization options to cater to different use cases.

In summary, File Manager aspires to redefine file and directory management by providing an all-encompassing, collaborative, and user-centric solution. Through its innovative features, commitment to security, and community-driven development, File Manager seeks to become a pivotal tool in the hands of individuals and teams navigating the challenges of effective data organization.

AIM OF THE PROJECT

1. **Intuitive User Experience:**
   Develop an intuitive graphical user interface (GUI) that not only simplifies file and directory operations but also creates a delightful user experience. The objective is to empower users of varying technical backgrounds with a visually appealing and accessible platform.

2. **Optimized File Operations:**
   Streamline common file and directory operations to minimize user effort and enhance productivity. File Manager aims to provide a set of tools that make tasks such as file creation, deletion, renaming, and movement efficient and straightforward.

3. **Promoting Collaboration:**
   Implement features that go beyond individual file management, fostering a collaborative environment. File Manager aims to enable multiple users to interact with shared data effectively, thereby enhancing teamwork and project collaboration.

4. **Leveraging Python Capabilities:**
   Harness the power of Python's os and shutil modules to optimize file operations. The objective is to empower developers with a familiar and powerful programming language, allowing for customization and extension of File Manager's functionality to meet specific project requirements.

5. **Ensuring Data Security:**
   Prioritize the security of user data through robust measures, including access controls, encryption options, and safeguards against data loss. The objective is to instill confidence in users regarding the confidentiality and integrity of their files and directories.

6. **Building a Vibrant Community:**
   Cultivate an active and engaged open-source community around File Manager. The objective is to encourage collaboration, feedback, and contributions from a diverse group

of developers and users, fostering an environment of continuous improvement and collective evolution.

7. **Thorough Documentation and Support:**
   Develop comprehensive documentation to guide users and contributors through File Manager's features and functionalities. The objective is to ensure that users have the resources they need for a positive and user-friendly experience, supported by responsive and helpful assistance.

8. **Cross-Platform Accessibility:**
   Ensure File Manager's compatibility with various operating systems, eliminating barriers to entry and promoting accessibility across diverse computing environments. The objective is to provide users with a consistent and reliable experience regardless of their chosen platform.

9. **Tailoring for Diverse Use Cases:**
   Design File Manager to be adaptable to the diverse needs of users. The objective is to offer customization options that cater to different use cases, whether users are developers organizing projects, system administrators managing routine tasks, or everyday users handling personal files.

- In summary, the aim of File Manager is to be a transformative force in file and directory management, while the specific objectives delineate the actionable steps and goals that contribute to the realization of this overarching vision

CONCLUSION

In conclusion, File Manager stands as a testament to innovation in the realm of file and directory management. Rooted in the robust capabilities of Python and guided by the principles of open-source collaboration, this project aims to redefine the user experience and set new standards in efficiency, collaboration, and security.

As we embark on this journey, the commitment to creating an intuitive graphical user interface remains unwavering. We aspire to simplify file and directory operations, ensuring that users, regardless of their technical proficiency, find joy in managing their data.

Optimizing file operations is not just a goal; it's a commitment to enhancing user productivity. Through streamlined processes, File Manager seeks to empower users, providing them with a set of tools that make tasks such as file creation, deletion, renaming, and movement efficient and seamless.

Collaboration lies at the heart of File Manager's objectives. Beyond individual file management, our project strives to foster a collaborative environment, enabling multiple users to interact with shared data effectively. We envision a tool that not only organizes files but also facilitates teamwork and project collaboration.

By leveraging the capabilities of Python's os and shutil modules, File Manager aims to provide developers with a familiar and powerful programming language. The project encourages customization and extension, allowing users to tailor the tool to their specific project requirements.

Security is a paramount concern, and File Manager addresses it head-on. With robust measures including access controls, encryption options, and safeguards against data loss, our project is committed to ensuring the confidentiality and integrity of user files and directories.

A vibrant open-source community is the lifeblood of File Manager. We call upon developers and users alike to join us in shaping the future of this project. Together, we can create an environment of continuous improvement and collective evolution.

Thorough documentation and responsive user support underscore our dedication to providing a positive and user-friendly experience. We believe that users should have the resources they need to navigate File Manager seamlessly.

Cross-platform accessibility eliminates barriers to entry, ensuring that File Manager is available and reliable across diverse computing environments. We aim to provide users with a consistent and dependable experience, regardless of their chosen platform.

In designing File Manager to be adaptable to diverse user needs, we acknowledge that every user is unique. Whether you are a developer organizing projects, a system administrator managing routine tasks, or an everyday user handling personal files, File Manager offers customization options to cater to your specific use case.

In essence, File Manager is not just a tool; it's a vision for a future where file and directory management are synonymous with efficiency, collaboration, and security. We invite you to join us on this exciting journey as we redefine the way users interact with their data. Together, let's shape the future of File Manager and elevate the standard of file and directory management for users around the world.

REFERENCES

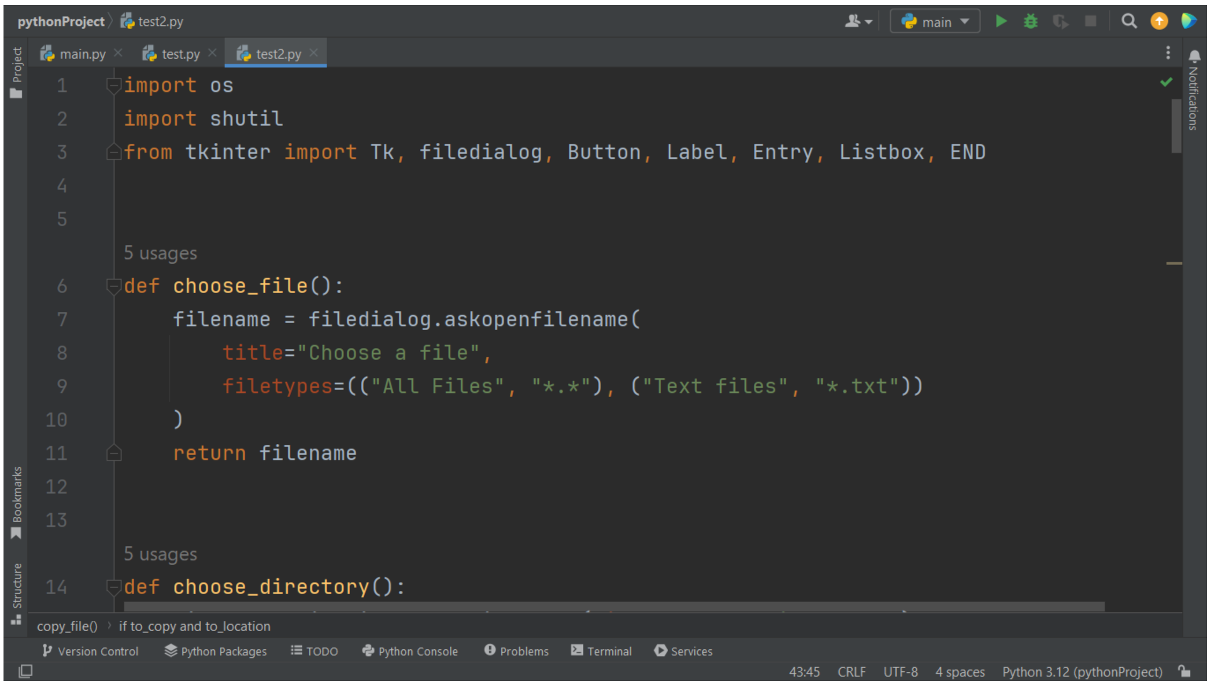
- **"Operating System Concepts" by Abraham Silberschatz, Peter B. Galvin, Greg Gagne**

- **"Modern Operating Systems" by Andrew S. Tanenbaum**

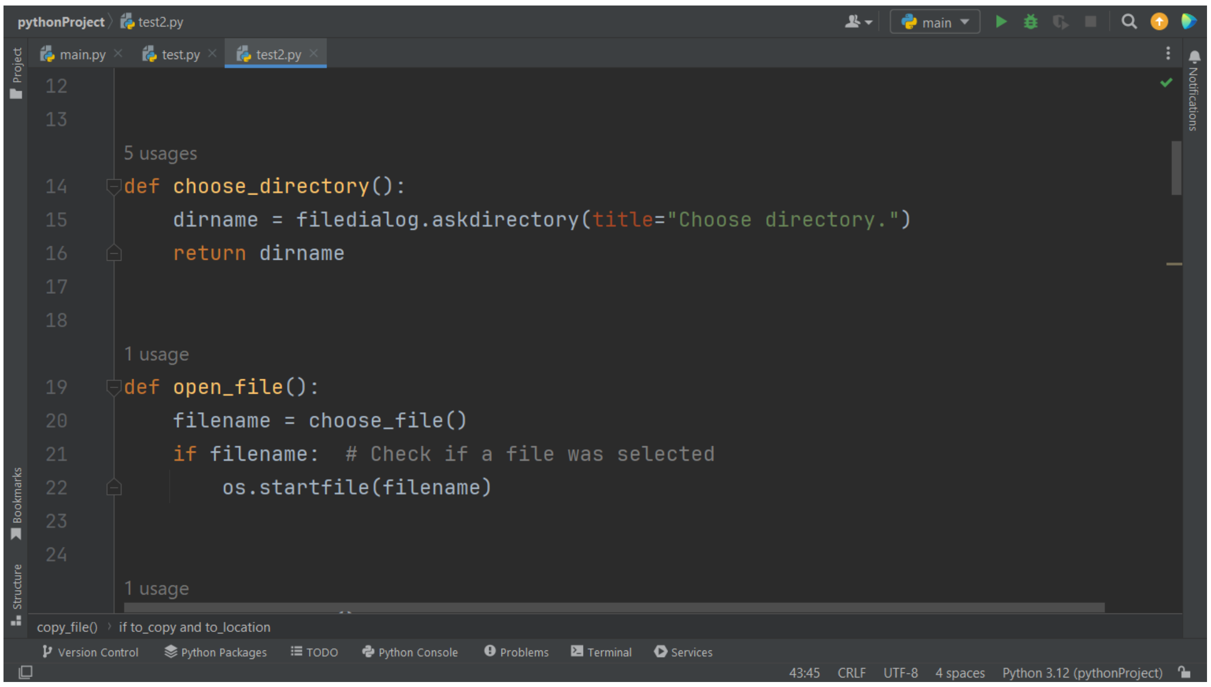
- **File Explorer (Operating System in python)**
  - **Website: https://www.javatpoint.com/file-explorer-using-tkinter-in-python**

- **Python Tkinter Tutorial (geeks of geeks)**
  - **Website: https://www.geeksforgeeks.org/python-tkinter-tutorial/**

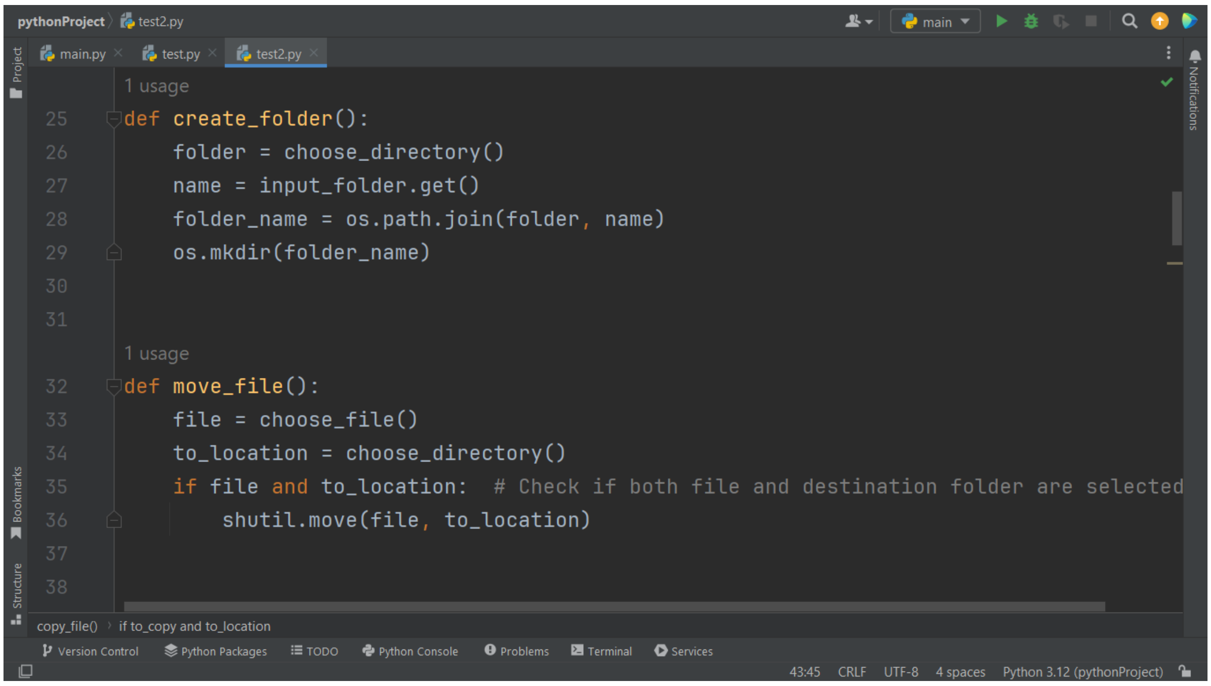
SOURCE CODE



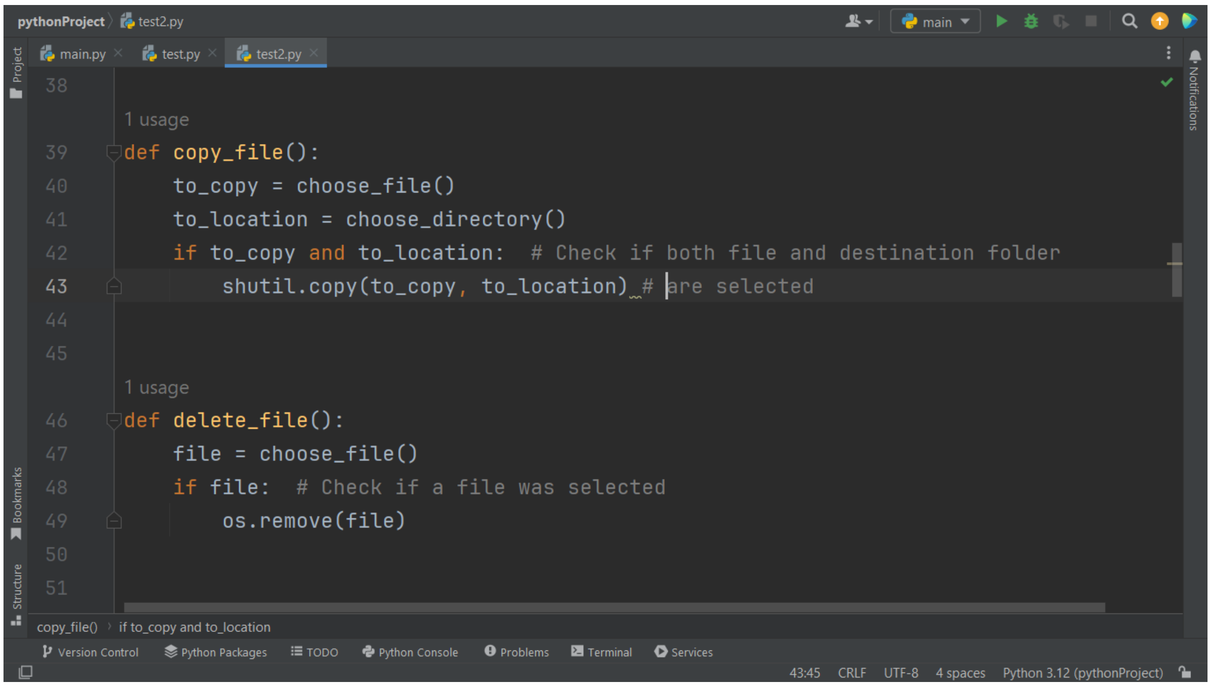
```
import os
import shutil
from tkinter import Tk, filedialog, Button, Label, Entry, Listbox, END


5 usages
def choose_file():
    filename = filedialog.askopenfilename(
        title="Choose a file",
        filetypes=(("All Files", "*.*"), ("Text files", "*.txt"))
    )
    return filename


5 usages
def choose_directory():
```



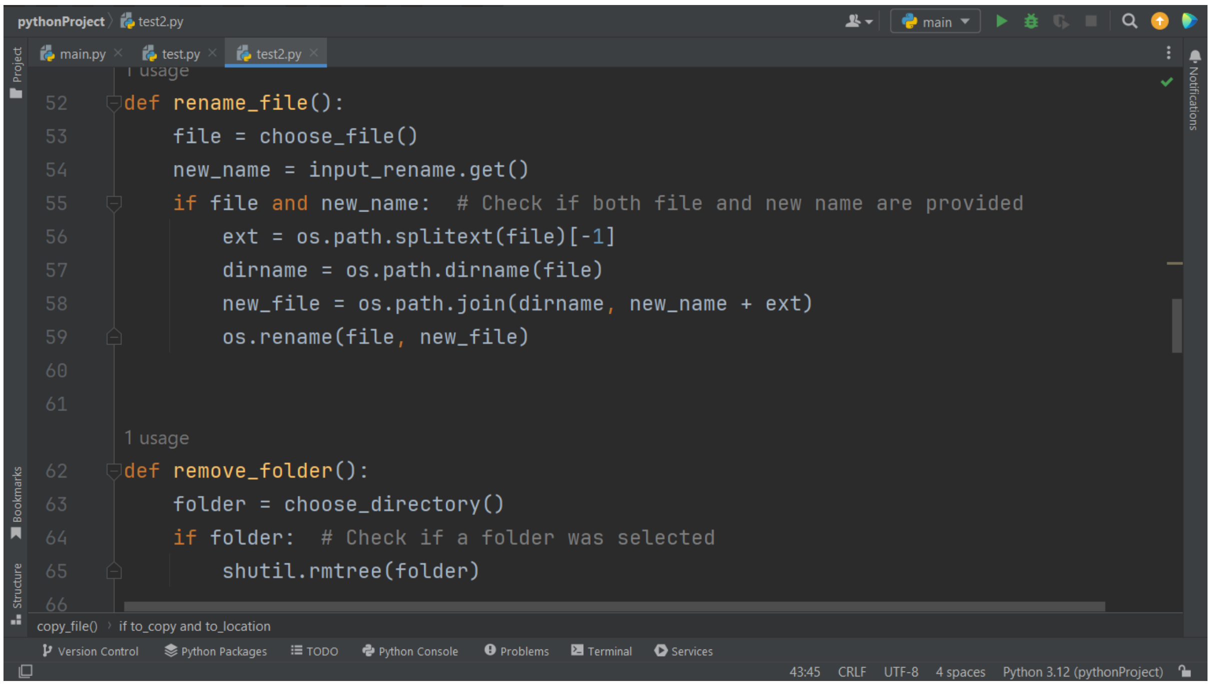
```


5 usages
def choose_directory():
    dirname = filedialog.askdirectory(title="Choose directory.")
    return dirname


1 usage
def open_file():
    filename = choose_file()
    if filename:  # Check if a file was selected
        os.startfile(filename)


1 usage
```



```
1 usage
def create_folder():
    folder = choose_directory()
    name = input_folder.get()
    folder_name = os.path.join(folder, name)
    os.mkdir(folder_name)


1 usage
def move_file():
    file = choose_file()
    to_location = choose_directory()
    if file and to_location:  # Check if both file and destination folder are selected
        shutil.move(file, to_location)


```



```

1 usage
def copy_file():
    to_copy = choose_file()
    to_location = choose_directory()
    if to_copy and to_location:  # Check if both file and destination folder
        shutil.copy(to_copy, to_location) # are selected


1 usage
def delete_file():
    file = choose_file()
    if file:  # Check if a file was selected
        os.remove(file)


```



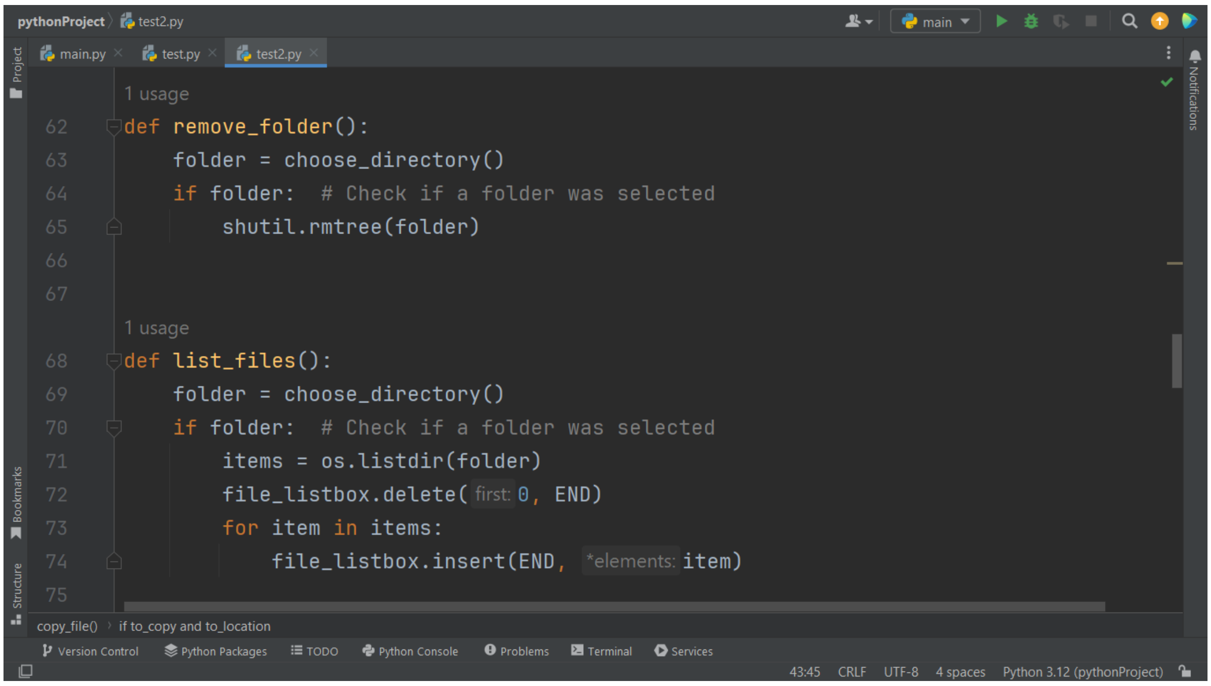
pythonProject 〉 test2.py

main.py test.py test2.py

```
1 usage
def rename_file():
    file = choose_file()
    new_name = input_rename.get()
    if file and new_name:  # Check if both file and new name are provided
        ext = os.path.splitext(file)[-1]
        dirname = os.path.dirname(file)
        new_file = os.path.join(dirname, new_name + ext)
        os.rename(file, new_file)


1 usage
def remove_folder():
    folder = choose_directory()
    if folder:  # Check if a folder was selected
        shutil.rmtree(folder)
```

copy_file() 〉 if to_copy and to_location

43:45 CRLF UTF-8 4 spaces Python 3.12 (pythonProject)



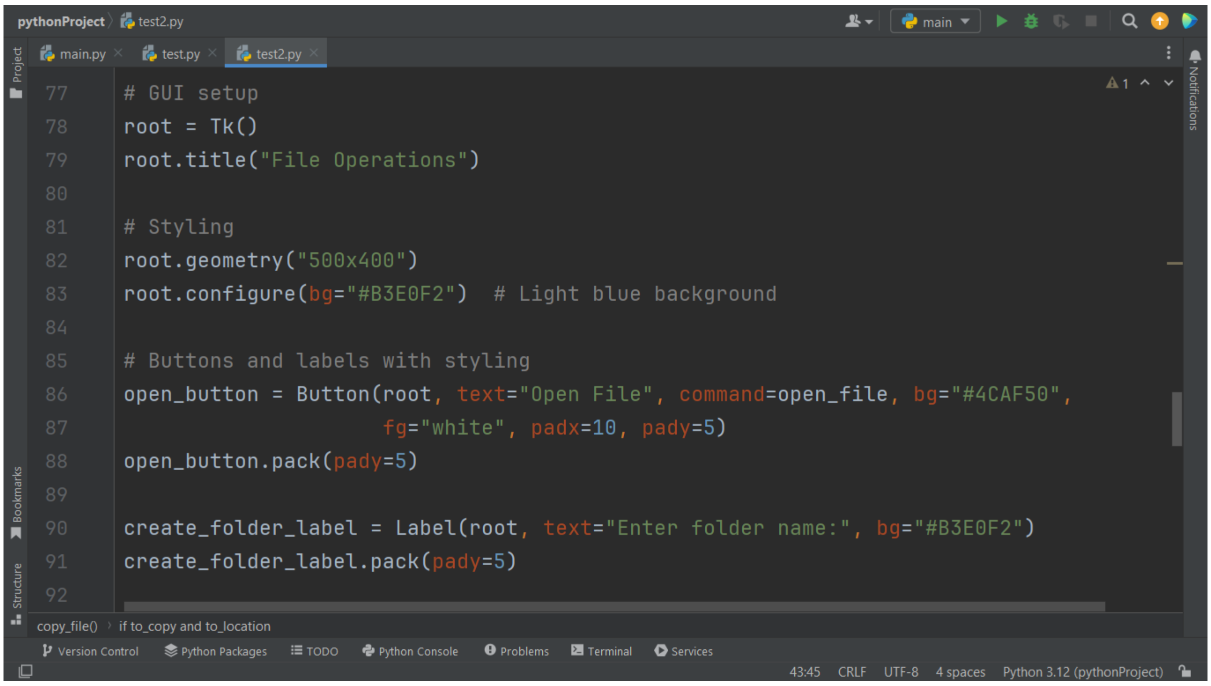
pythonProject 〉 test2.py

main.py test.py test2.py

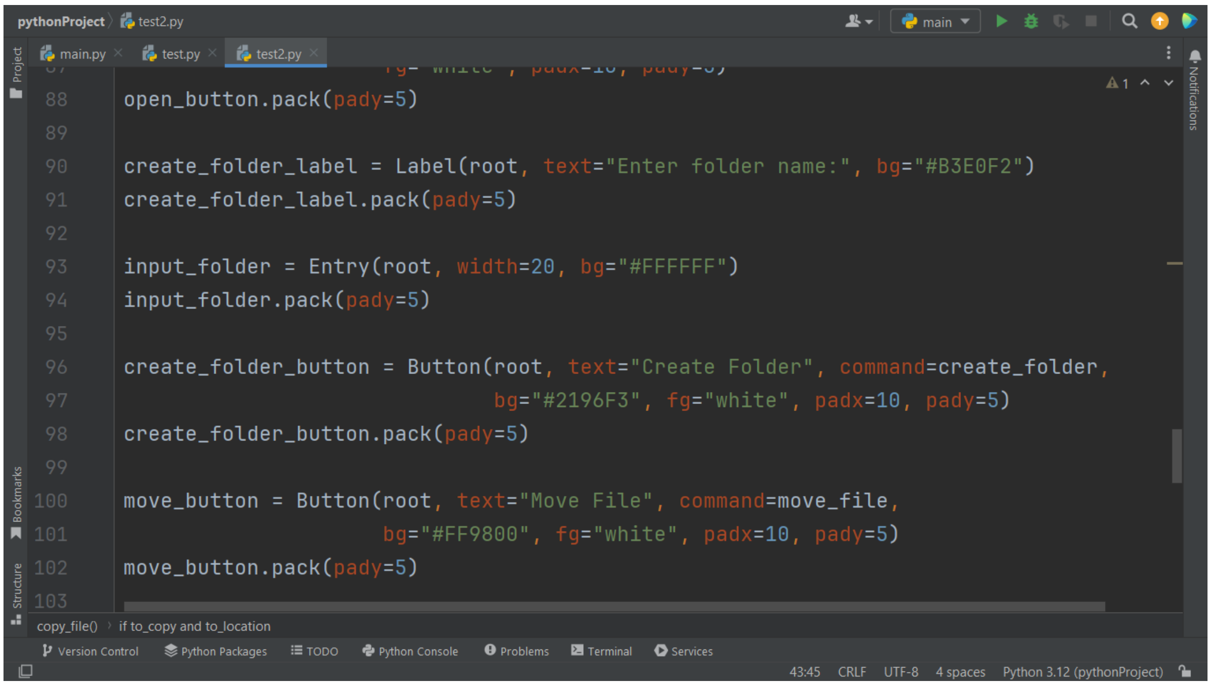
```
1 usage
def remove_folder():
    folder = choose_directory()
    if folder:  # Check if a folder was selected
        shutil.rmtree(folder)


1 usage
def list_files():
    folder = choose_directory()
    if folder:  # Check if a folder was selected
        items = os.listdir(folder)
        file_listbox.delete( first: 0, END)
        for item in items:
            file_listbox.insert(END,  *elements: item)
```

copy_file() 〉 if to_copy and to_location

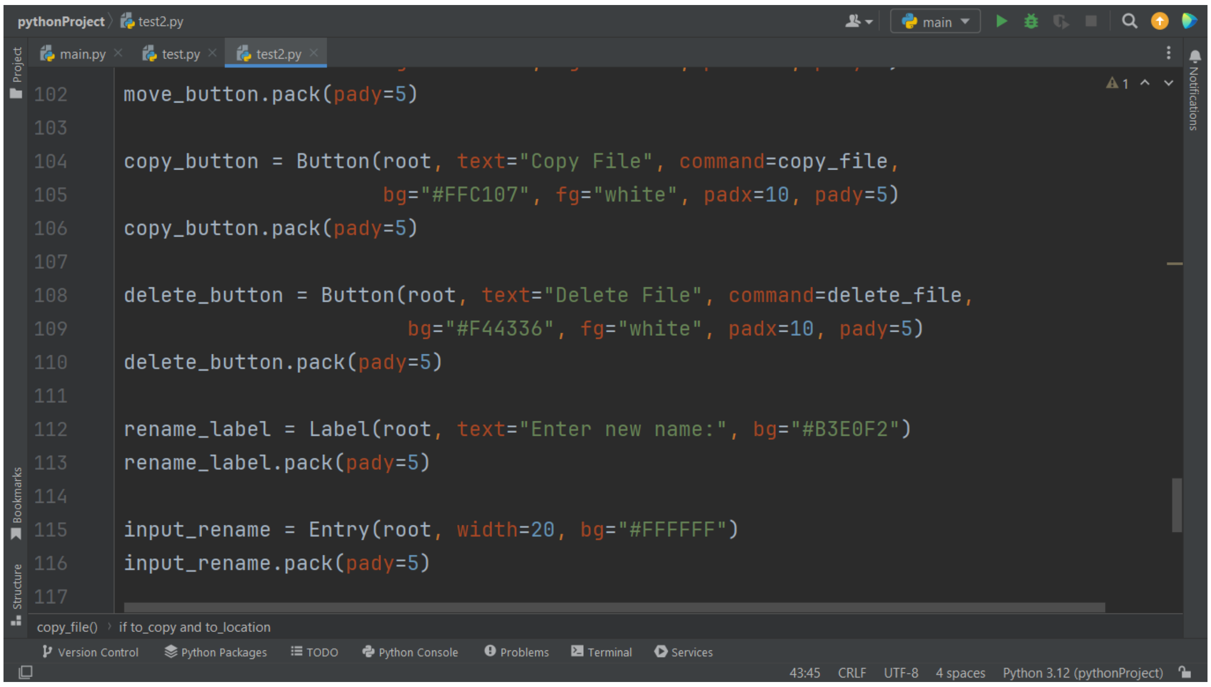
43:45 CRLF UTF-8 4 spaces Python 3.12 (pythonProject)



```python
# GUI setup
root = Tk()
root.title("File Operations")

# Styling
root.geometry("500x400")
root.configure(bg="#B3E0F2")  # Light blue background

# Buttons and labels with styling
open_button = Button(root, text="Open File", command=open_file, bg="#4CAF50",
                     fg="white", padx=10, pady=5)
open_button.pack(pady=5)

create_folder_label = Label(root, text="Enter folder name:", bg="#B3E0F2")
create_folder_label.pack(pady=5)

```



```python
open_button.pack(pady=5)

create_folder_label = Label(root, text="Enter folder name:", bg="#B3E0F2")
create_folder_label.pack(pady=5)

input_folder = Entry(root, width=20, bg="#FFFFFF")
input_folder.pack(pady=5)

create_folder_button = Button(root, text="Create Folder", command=create_folder,
                              bg="#2196F3", fg="white", padx=10, pady=5)
create_folder_button.pack(pady=5)

move_button = Button(root, text="Move File", command=move_file,
                     bg="#FF9800", fg="white", padx=10, pady=5)
move_button.pack(pady=5)

```

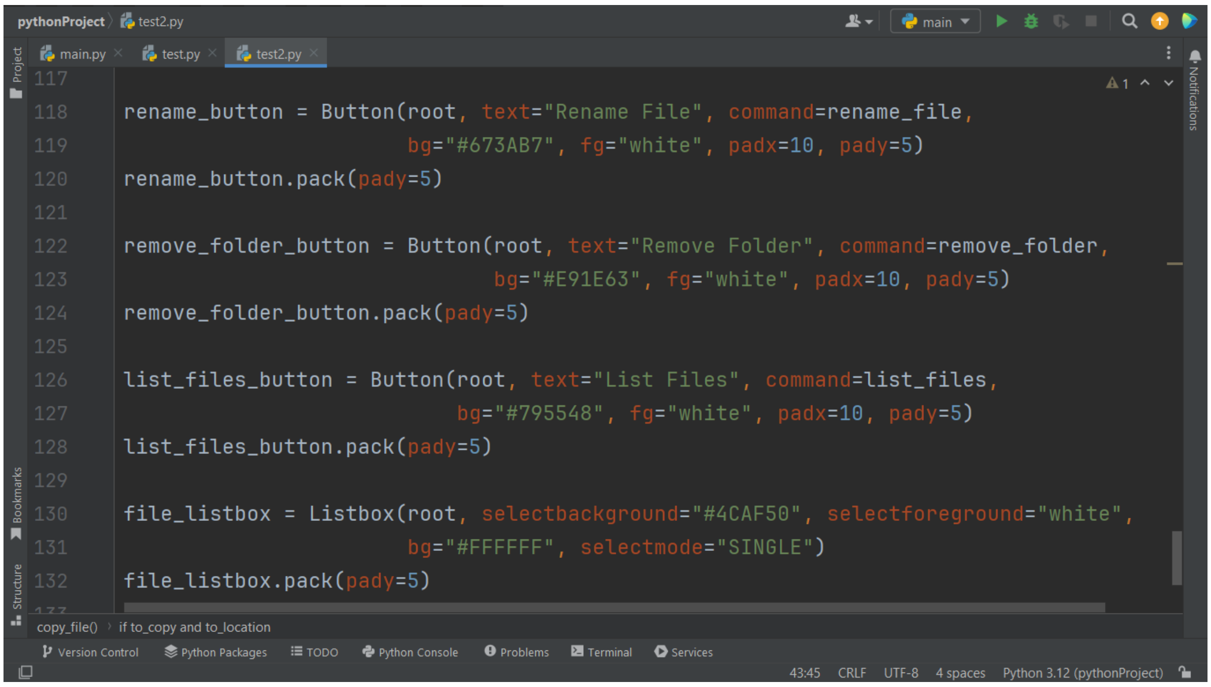


pythonProject 〉 test2.py
main.py test.py test2.py

```
move_button.pack(pady=5)

copy_button = Button(root, text="Copy File", command=copy_file,
                     bg="#FFC107", fg="white", padx=10, pady=5)
copy_button.pack(pady=5)

delete_button = Button(root, text="Delete File", command=delete_file,
                       bg="#F44336", fg="white", padx=10, pady=5)
delete_button.pack(pady=5)

rename_label = Label(root, text="Enter new name:", bg="#B3E0F2")
rename_label.pack(pady=5)

input_rename = Entry(root, width=20, bg="#FFFFFF")
input_rename.pack(pady=5)

```

copy_file() 〉 if to_copy and to_location

43:45 CRLF UTF-8 4 spaces Python 3.12 (pythonProject)

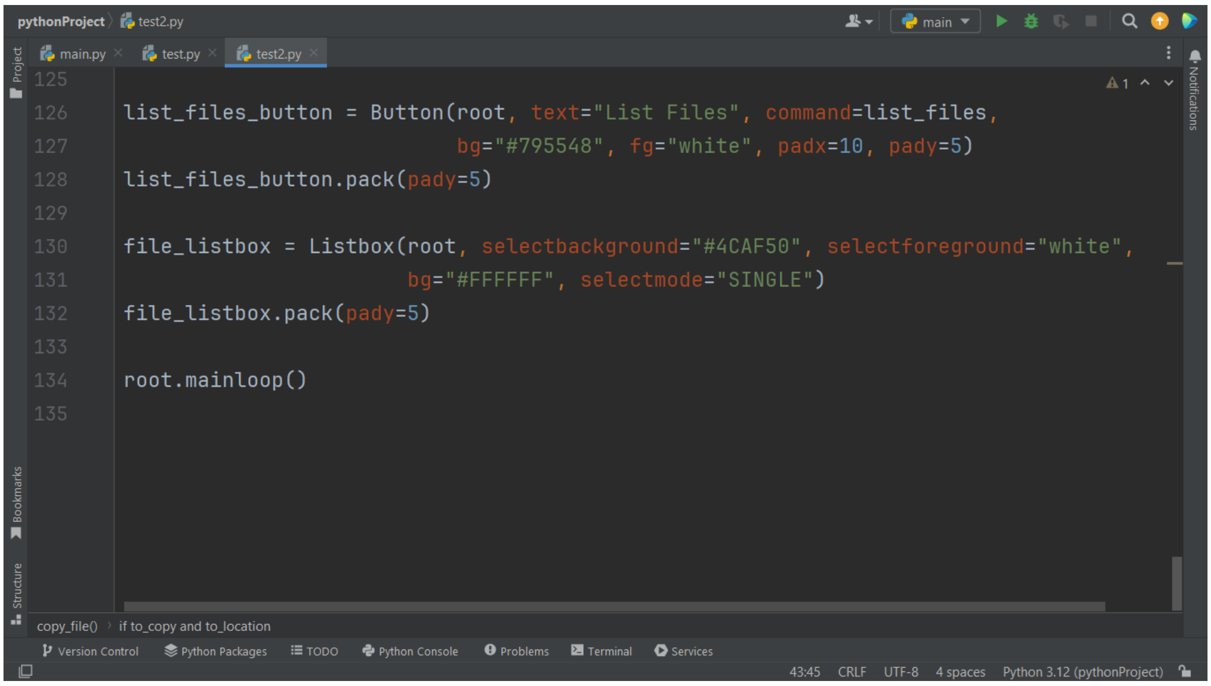


pythonProject 〉 test2.py
main.py test.py test2.py

```

rename_button = Button(root, text="Rename File", command=rename_file,
                       bg="#673AB7", fg="white", padx=10, pady=5)
rename_button.pack(pady=5)

remove_folder_button = Button(root, text="Remove Folder", command=remove_folder,
                              bg="#E91E63", fg="white", padx=10, pady=5)
remove_folder_button.pack(pady=5)

list_files_button = Button(root, text="List Files", command=list_files,
                           bg="#795548", fg="white", padx=10, pady=5)
list_files_button.pack(pady=5)

file_listbox = Listbox(root, selectbackground="#4CAF50", selectforeground="white",
                       bg="#FFFFFF", selectmode="SINGLE")
file_listbox.pack(pady=5)
```

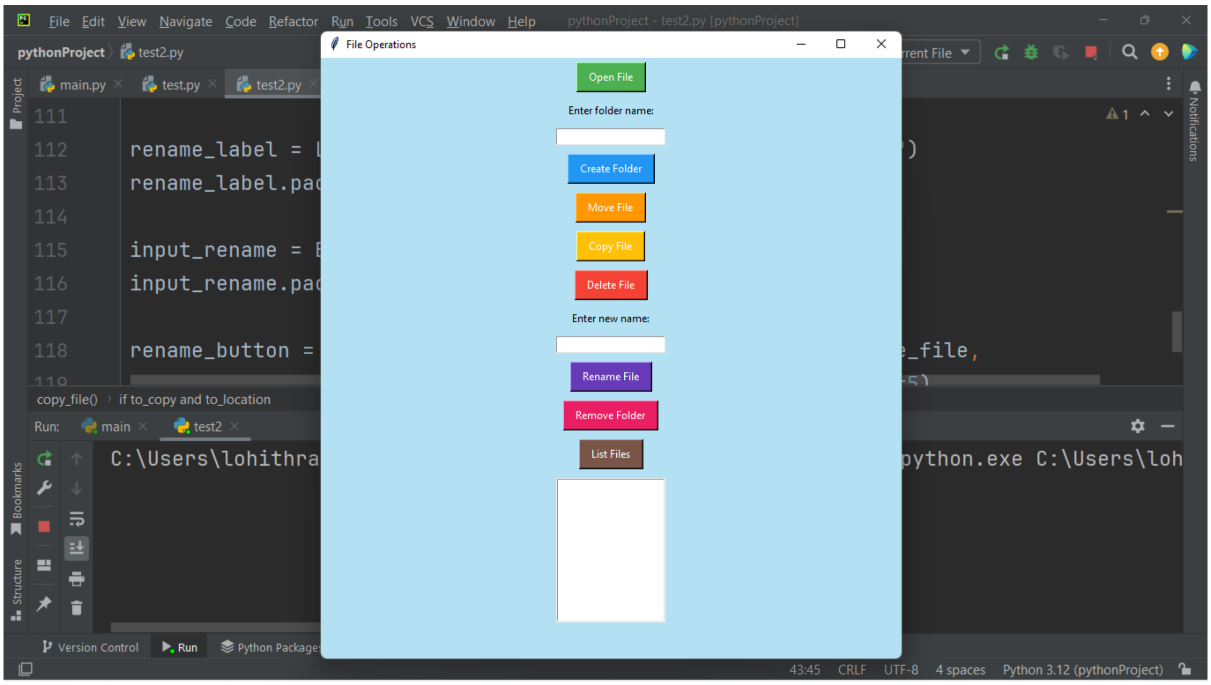
copy_file() 〉 if to_copy and to_location

43:45 CRLF UTF-8 4 spaces Python 3.12 (pythonProject)



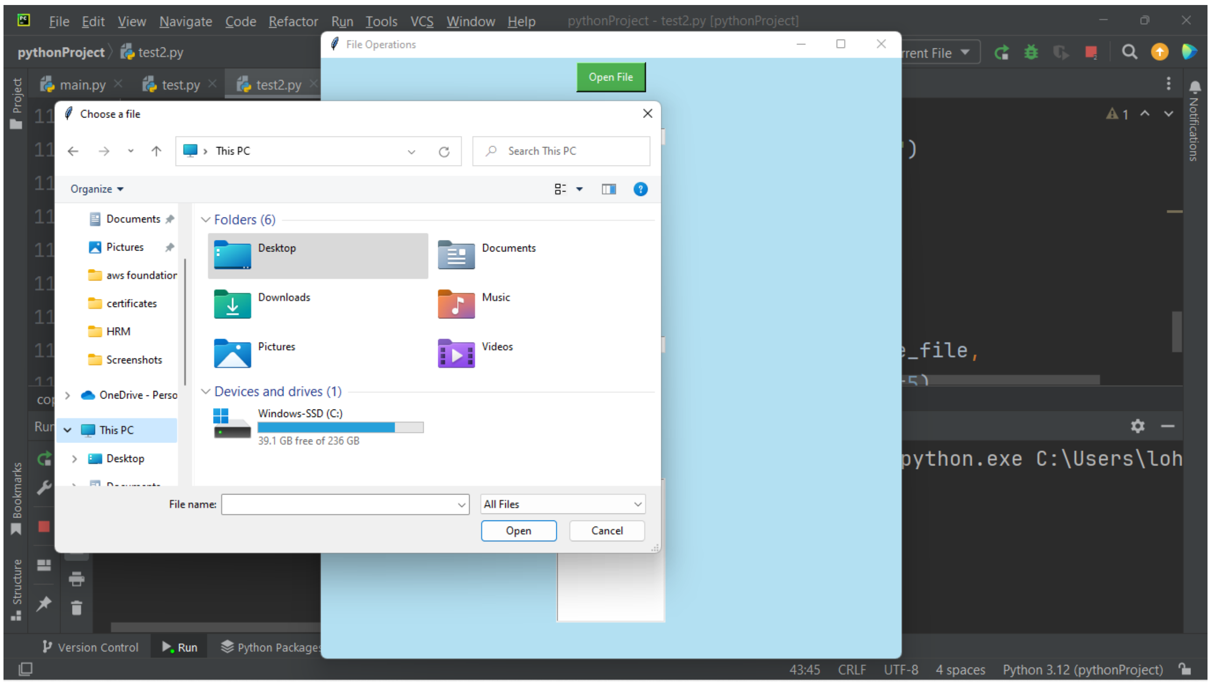
```

list_files_button = Button(root, text="List Files", command=list_files,
                           bg="#795548", fg="white", padx=10, pady=5)
list_files_button.pack(pady=5)

file_listbox = Listbox(root, selectbackground="#4CAF50", selectforeground="white",
                       bg="#FFFFFF", selectmode="SINGLE")
file_listbox.pack(pady=5)

root.mainloop()

```

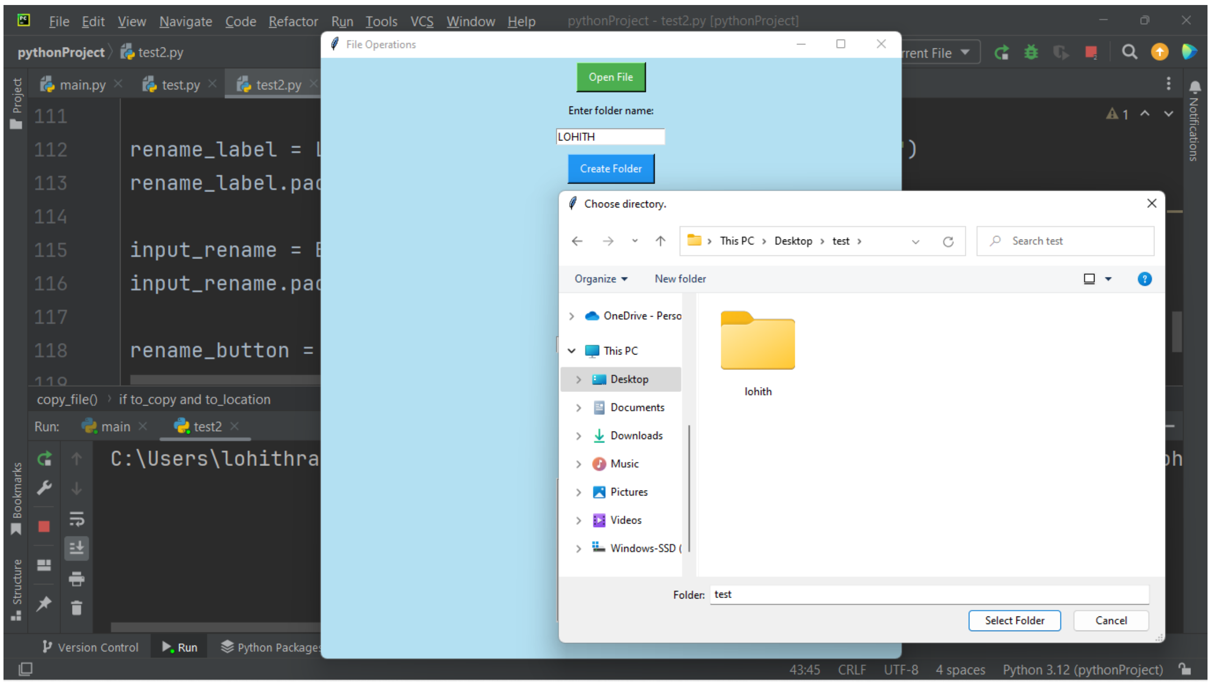
OUTPUT



```
rename_label = L                                                     )
rename_label.pac

input_rename = E
input_rename.pac

rename_button =                                                      _file,
```

```
C:\Users\lohithra                                        python.exe C:\Users\loh
```





```
rename_label = L
rename_label.pac

input_rename = E
input_rename.pac

rename_button =
```

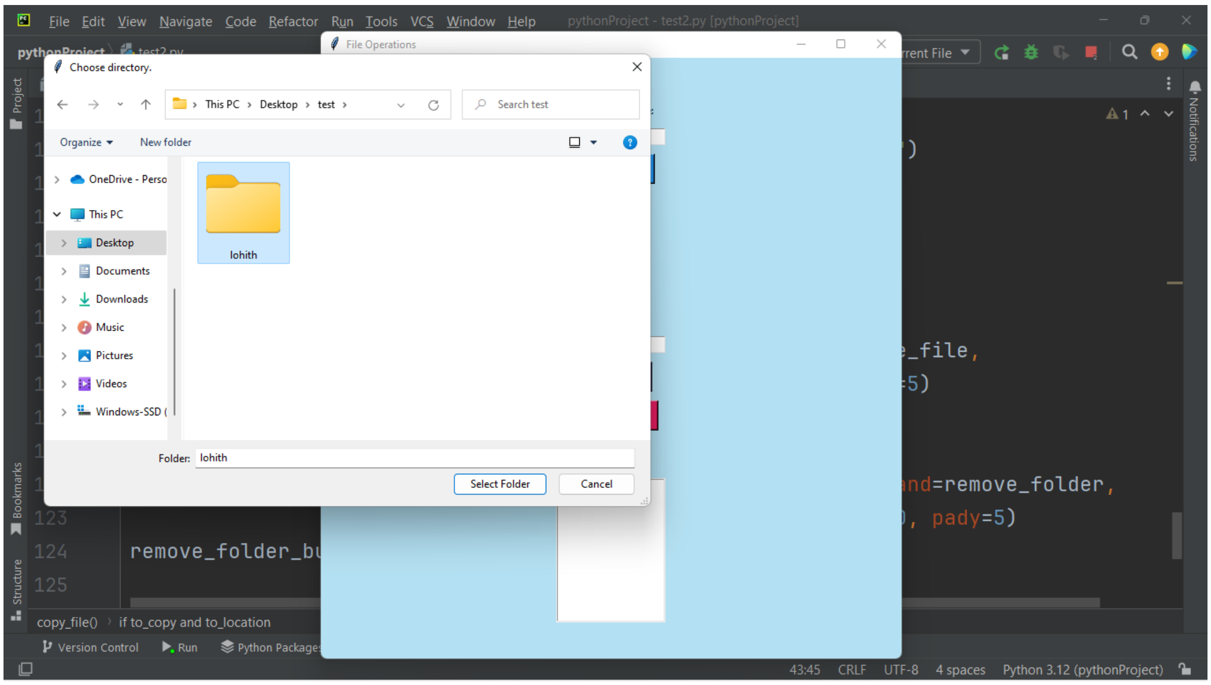
copy_file() › if to_copy and to_location

Run: main test2

```
C:\Users\lohithra
```

Enter folder name: LOHITH

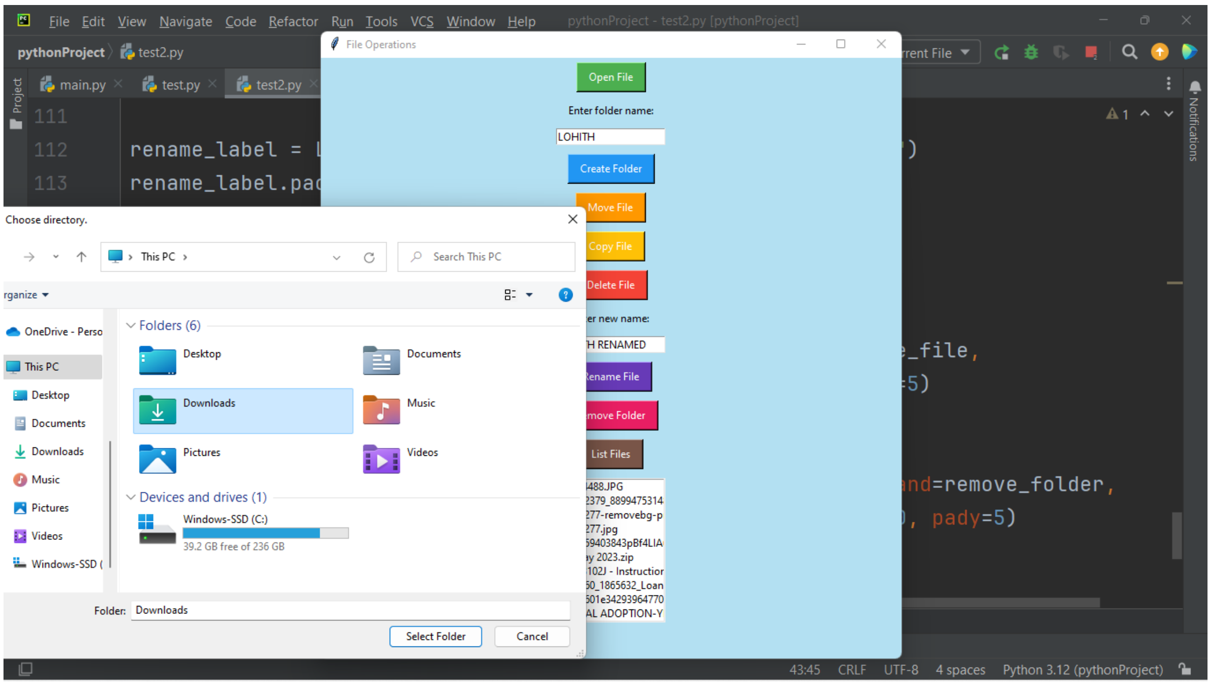
Folder: test


File Edit View Navigate Code Refactor Run Tools VCS Window Help
pythonProject - test2.py [pythonProject]
File Operations
Choose directory.
This PC > Desktop > test
Search test
Organize New folder
OneDrive - Perso
This PC
Desktop
Documents
Downloads
Music
Pictures
Videos
Windows-SSD (
lohith
Select Folder
Cancel
Version Control Run Python Packages
43:45 CRLF UTF-8 4 spaces Python 3.12 (pythonProject)

```
_file,
5)

nd=remove_folder,
), pady=5)
remove_folder_bu
```

copy_file() › if to_copy and to_location

Folder: lohith



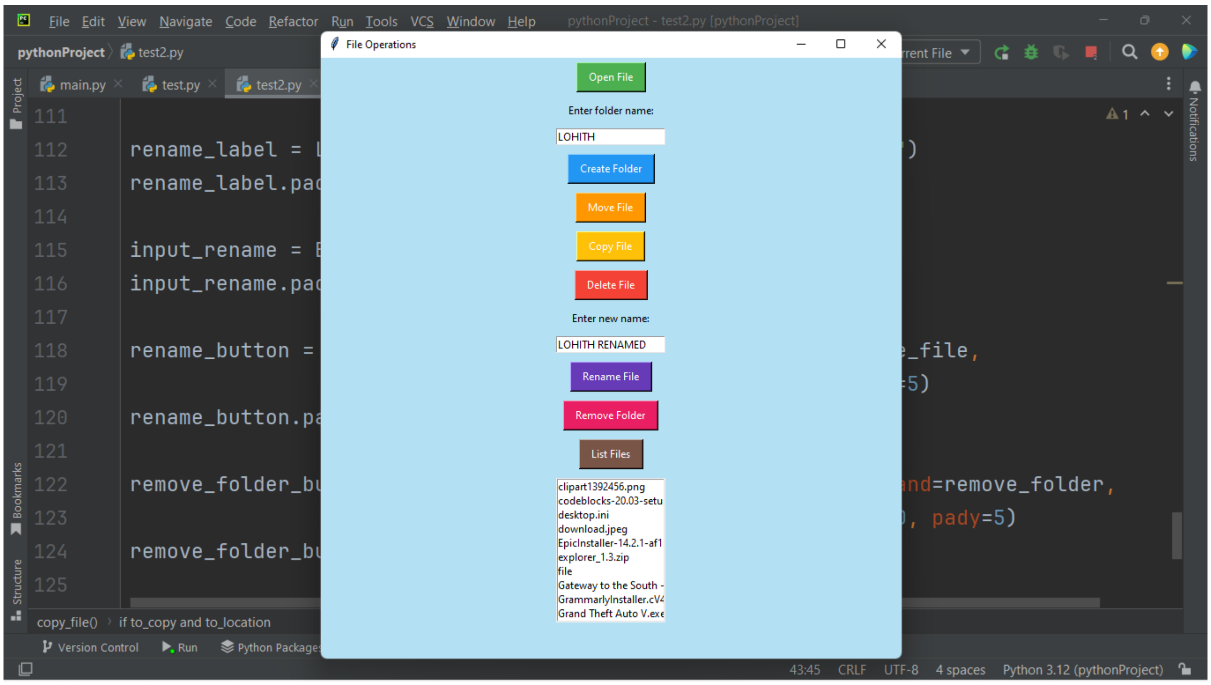
```
rename_label = L                                              )
rename_label.pac
                                                       _file,
                                                       5)

                                                       nd=remove_folder,
                                                       ), pady=5)
```

Enter folder name: LOHITH

er new name: H RENAMED

```
488.JPG
2379_8899475314
277-removebg-p
277.jpg
59403843pBf4LIA
ay 2023.zip
102J - Instructior
50_1865632_Loan
501e34293964770
AL ADOPTION-Y
```

Folder: Downloads


File Edit View Navigate Code Refactor Run Tools VCS Window Help
pythonProject - test2.py [pythonProject]
pythonProject test2.py
main.py test.py test2.py
File Operations
Open File
Create Folder
Move File
Copy File
Delete File
Rename File
Remove Folder
List Files
Version Control Run Python Packages
43:45 CRLF UTF-8 4 spaces Python 3.12 (pythonProject)

```
rename_label = L                                              )
rename_label.pac

input_rename = E
input_rename.pac

rename_button =                                               _file,
                                                              5)
rename_button.pa

remove_folder_bu                                              nd=remove_folder,
                                                              ), pady=5)
remove_folder_bu
```

copy_file() › if to_copy and to_location

Enter folder name: LOHITH

Enter new name: LOHITH RENAMED

```
clipart1392456.png
codeblocks-20.03-setu
desktop.ini
download.jpeg
EpicInstaller-14.2.1-af1
explorer_1.3.zip
file
Gateway to the South -
GrammarlyInstaller.cV4
Grand Theft Auto V.exe
```