**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

Automate File Copying with a Script

Create a script to copy files from one folder to another automatically.

Name: Subiksha R Department: AML



**Introduction:**

This task is about creating a Python script to copy files from one folder to another. It uses Python’s shutil library to handle file copying, ensures all files are transferred safely, and creates the destination folder if it doesn’t exist. It’s useful for automating tasks like backups or moving files easily.

**Objective:**

The objective of this task is to simplify and automate the process of copying files between two folders using Python. Instead of manually moving files, the script will handle everything, ensuring efficiency and consistency. Here's how the objective is broken down:

1. **Copy Files**: The script will take all the files in a **source folder** and copy them to a **destination folder**.
2. **Preserve Metadata**: When copying files, it’s important to keep metadata intact, such as the original file’s creation date and last modified time. This is achieved using Python's shutil.copy2() function.
3. **Create Destination Folder Automatically**: If the destination folder doesn’t exist, the script will create it. This ensures the process doesn’t fail due to missing directories.
4. **Feedback on File Copying**: After the script runs, it will let the user know how many files were successfully copied. This helps confirm the task was completed correctly.
5. **Handle Errors Gracefully**: If the source folder doesn’t exist or there’s an issue accessing the destination, the script will display an error message without crashing.

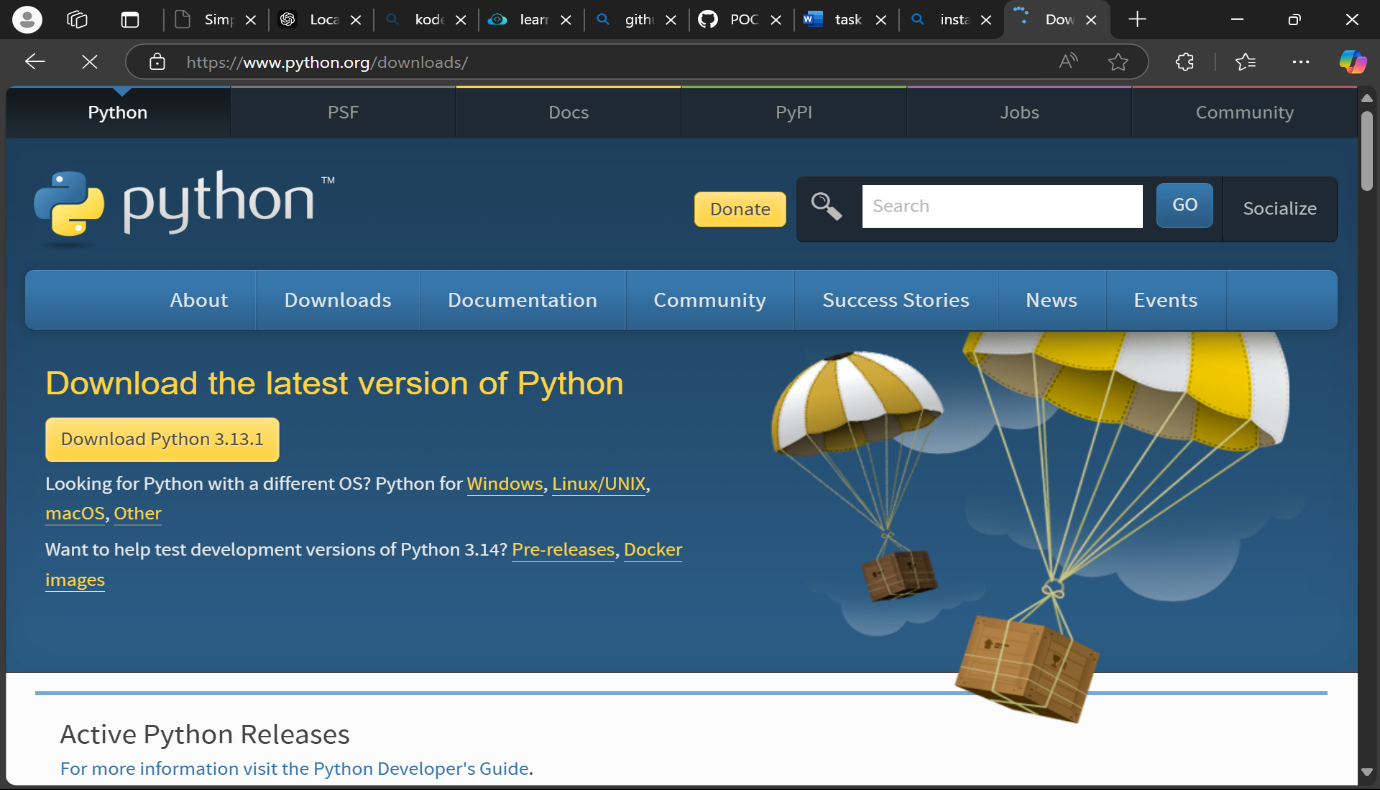
In short, the script makes file copying faster, safer, and more reliable, reducing manual effort and potential mistakes.

**Importance:**

Automating file copying is important for saving time and reducing manual effort. It ensures all files are copied accurately without missing any. The process is efficient for handling large numbers of files and repetitive tasks like backups. Automation reduces errors and can handle issues like missing folders or permissions gracefully. It is scalable, working well for small and large tasks. Additionally, the script can be customized for specific needs, like copying only certain file types. Overall, it simplifies workflows and improves reliability in file management.

**Step-by-Step procedure:**

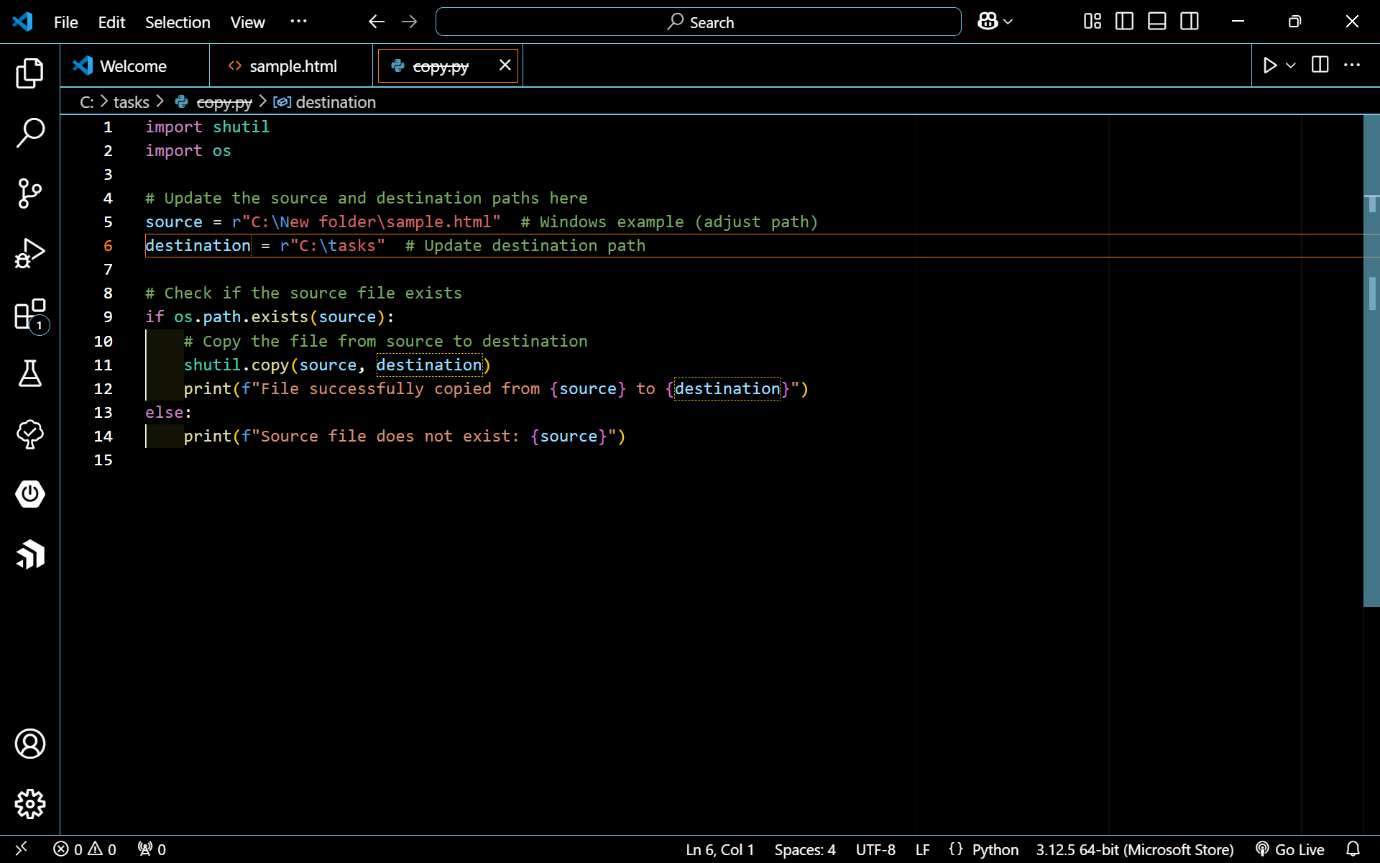
**STEP 1: Install Python** 1. Ensure Python is installed on your system. Download it from [python.org](https://www.python.org/downloads/) if necessary.

****

**STEP 2:** **Create the Python Script**

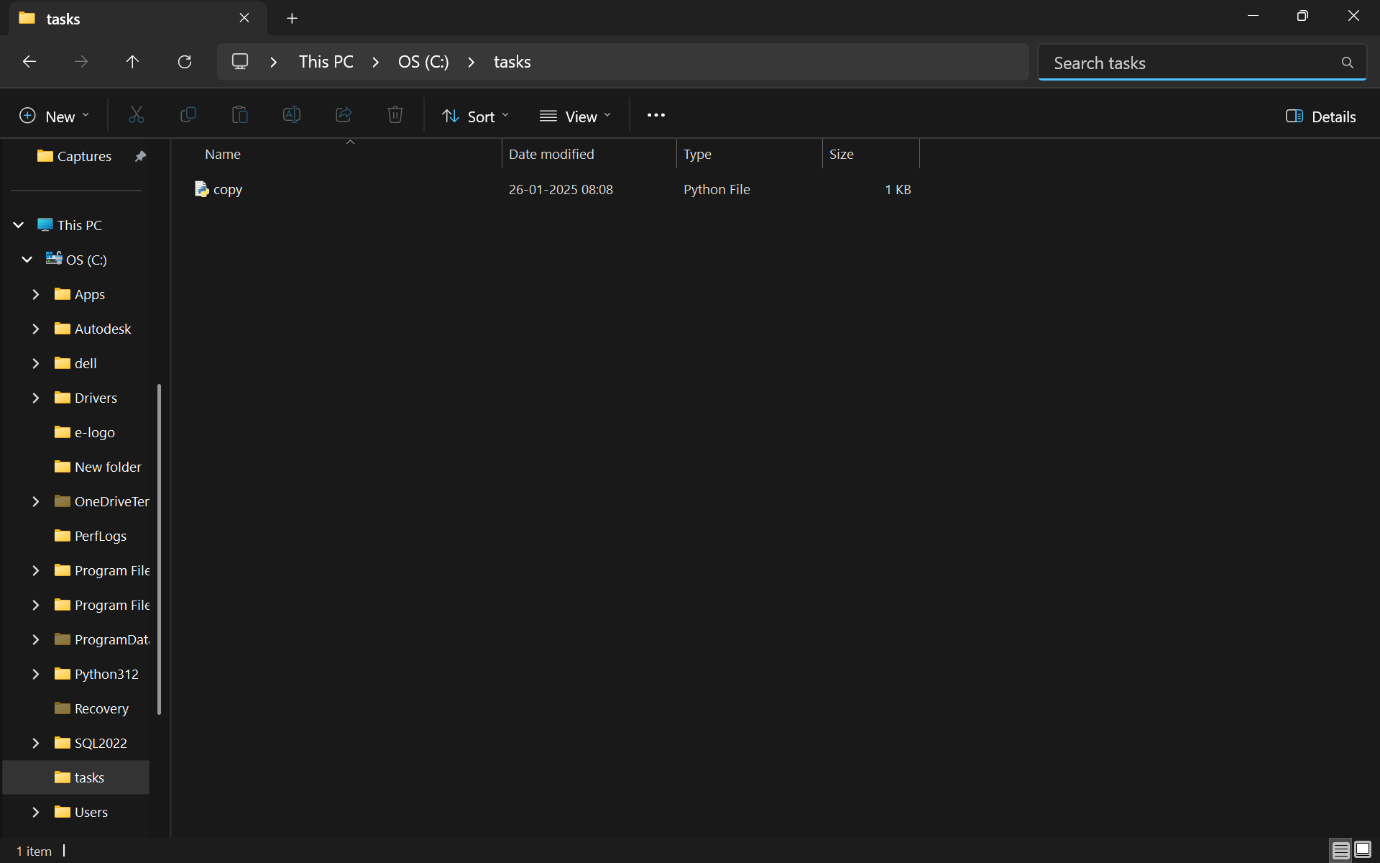
2 . Open a text editor (e.g., Notepad, VS Code, or PyCharm).

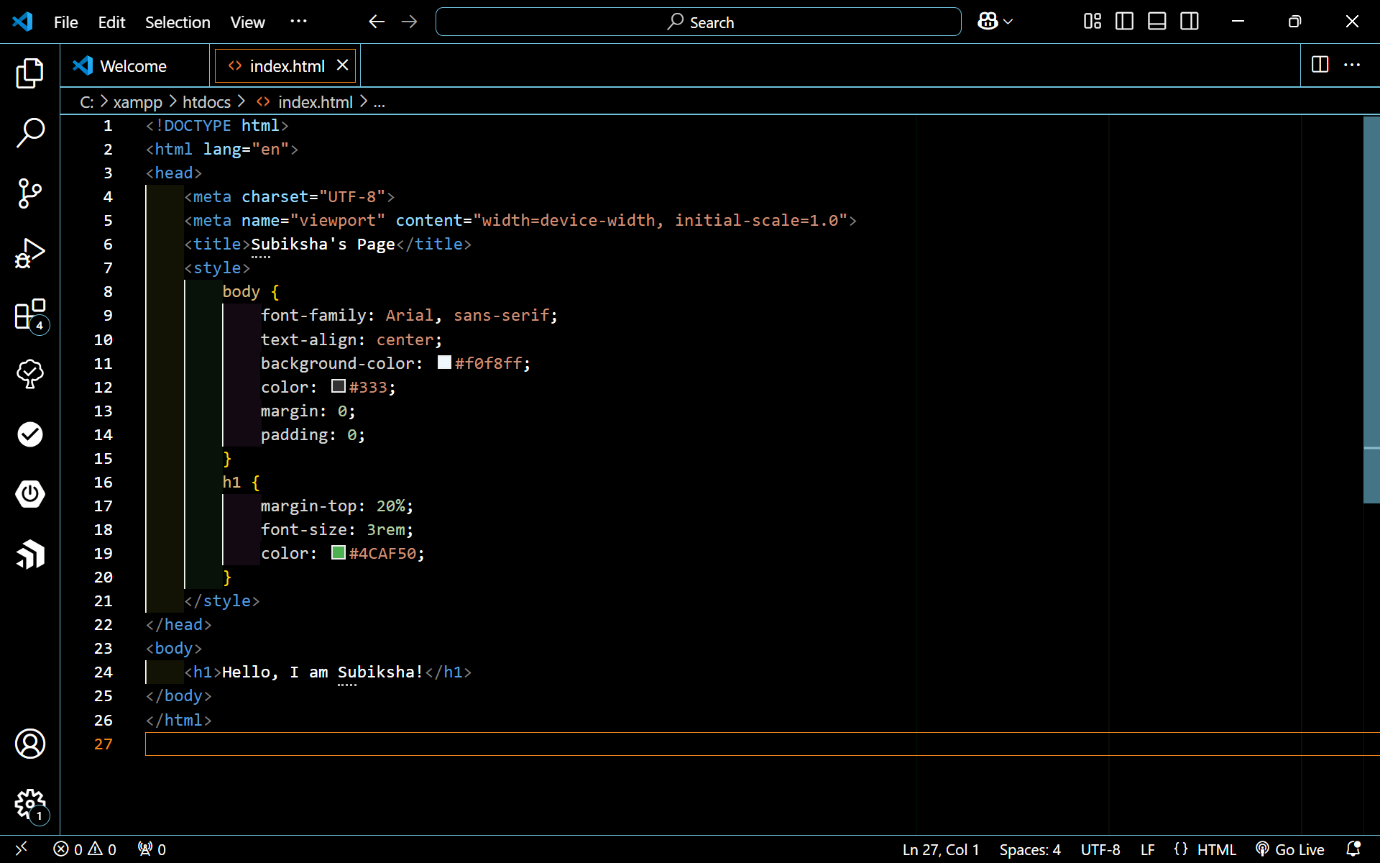
3. Write a script to copy files using the shutil module.



**STEP 3:** **Save the Script**

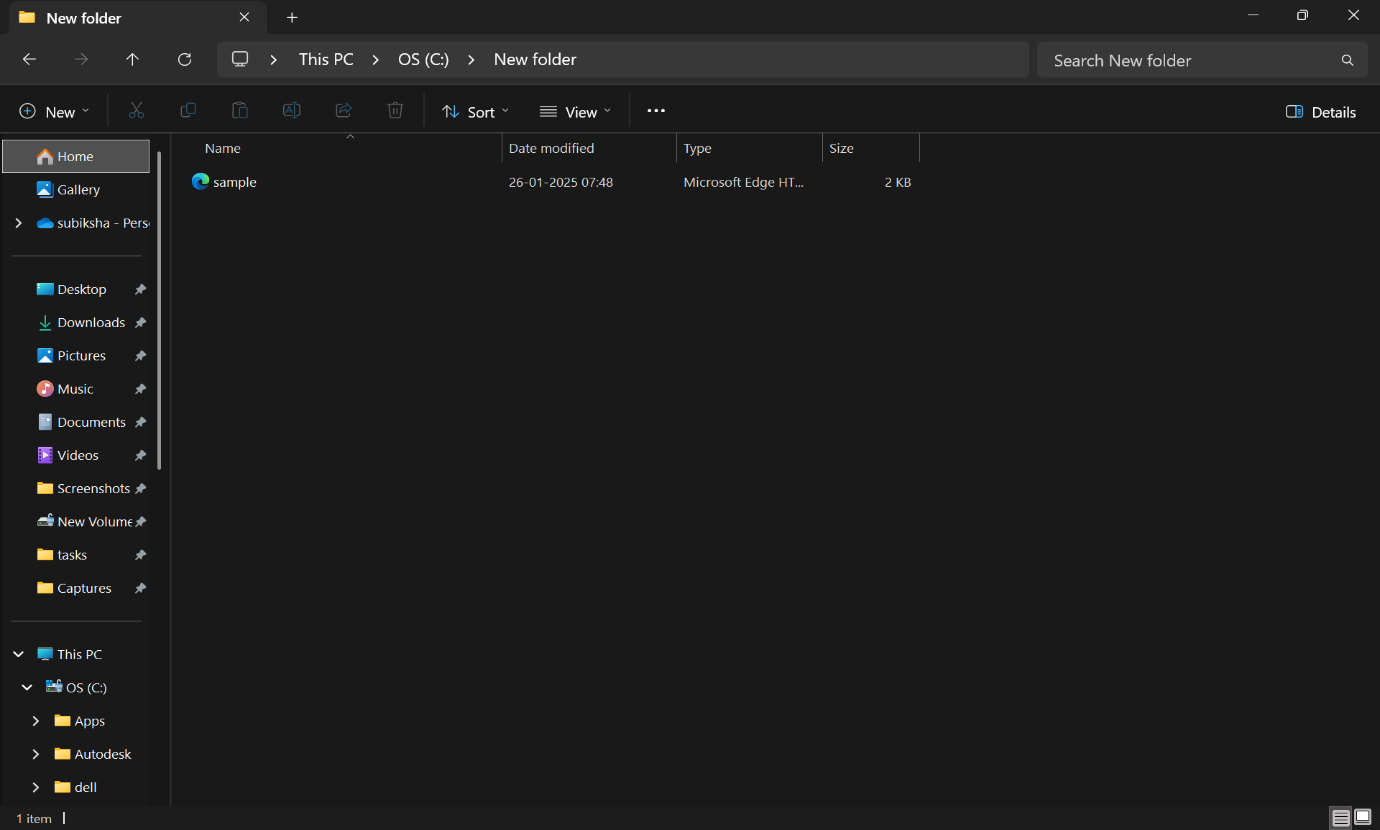
4. Save the file with a .py extension (e.g., copy\_files.py).

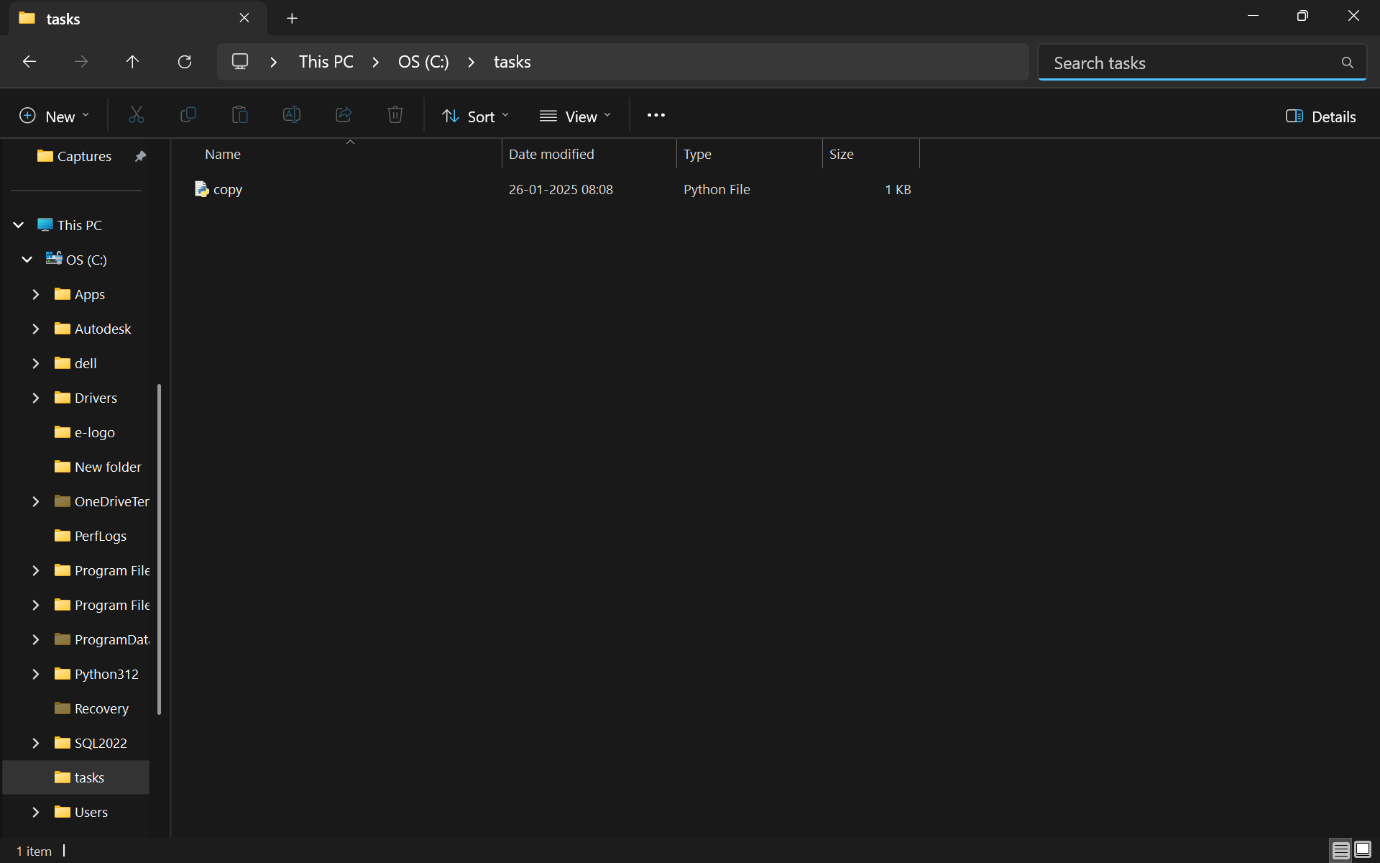
**STEP 4 : Create a file in a folder.**

****

**STEP 5 : Select the source and destination folder.**

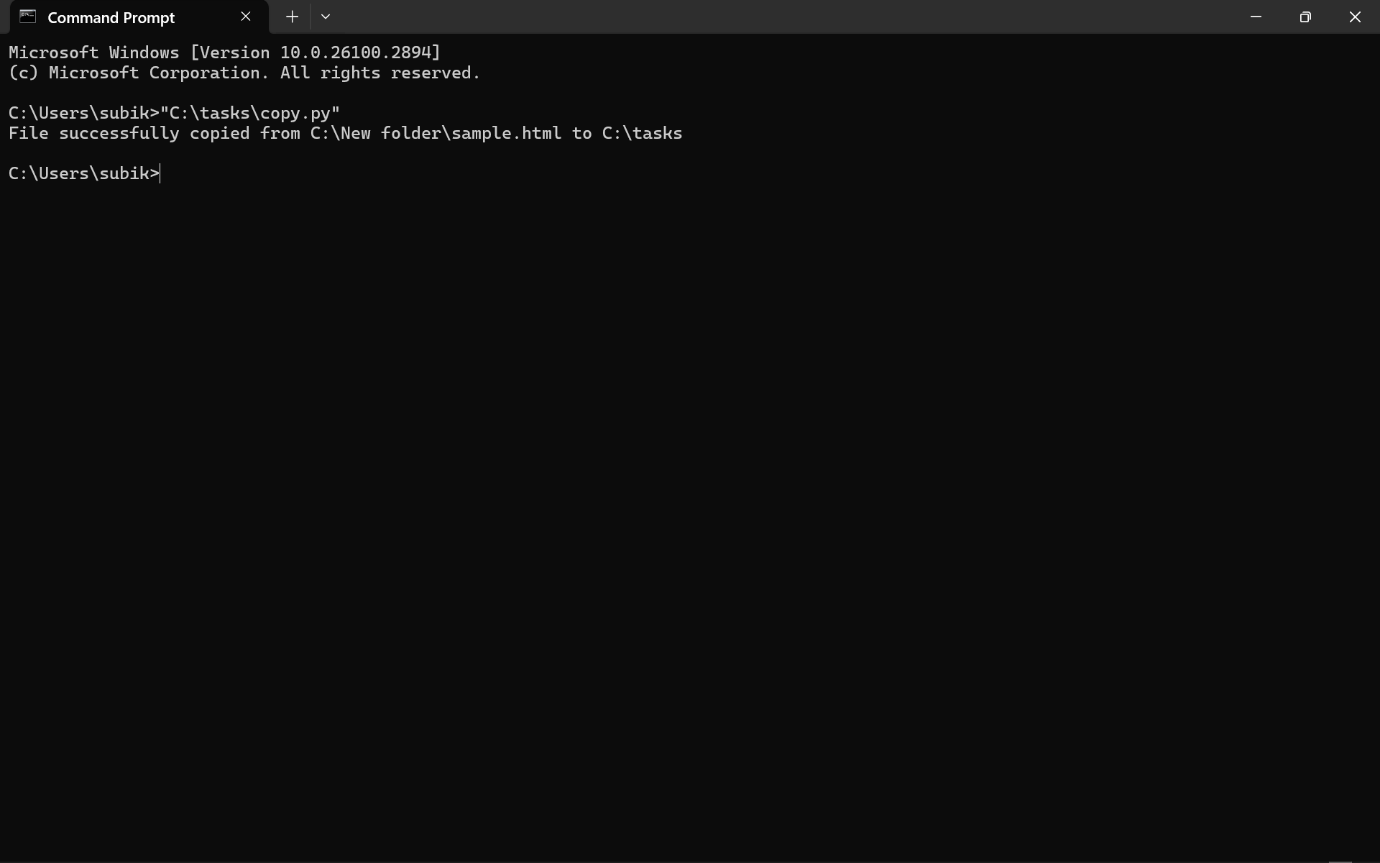
1. Copy the source and destination folder path.

****

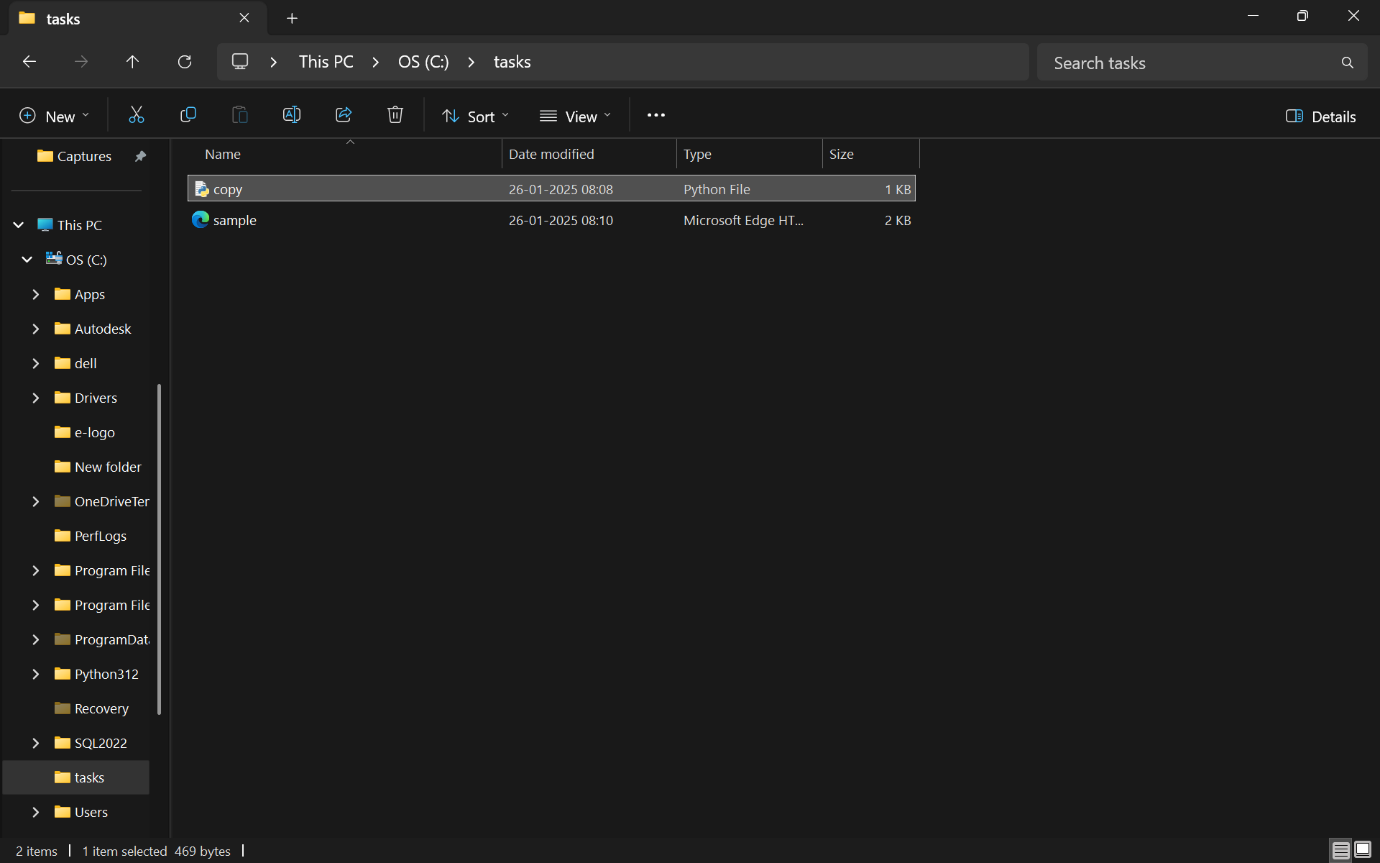
****

**STEP 6 : Run the Python Script.**

1. Open command prompt and run the file.



1. Then, the file will be copied to the destination folder automatically.



**Outcome :**

The outcome of copying files from a source folder to a destination folder will be:

1. Files are copied successfully from the source folder to the destination folder.
2. The destination folder is created if it doesn't already exist.
3. File metadata, such as modification timestamps, are preserved during the copy process.
4. A confirmation message is displayed for each file that is successfully copied.
5. If any errors occur, such as permission issues, an error message is shown for the problematic file.
6. If the source folder is empty, a message is shown indicating no files are available to copy.
7. If the source folder does not exist, a message is displayed notifying the user.