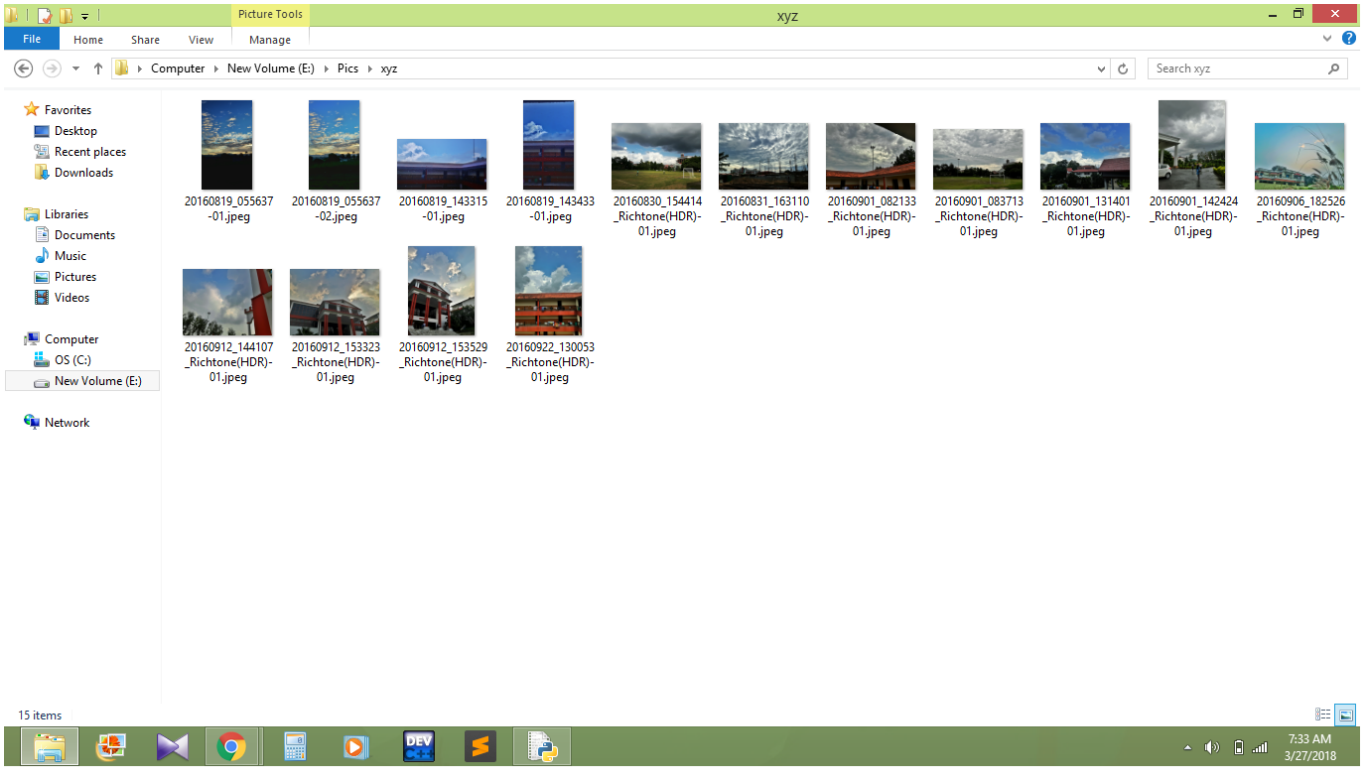
Rename Multiple Files Using C++

Overview

This project aims to automate the task of renaming multiple image files stored in a folder. Often, files are named randomly or inconsistently, making manual renaming tedious and error-prone. This solution uses basic file system handling methods to rename files in a consistent, ordered fashion such as hostel1, hostel2, and so on.

Problem Example

Original Folder Content (Before Renaming):



Solution Approach

This C++ program simulates the behavior often achieved using Python's os module by:

- Listing all files in a specified folder.
- Creating source (src) and destination (dst) file paths.
- Renaming each file iteratively using filesystem functions.

Rename Multiple Files Using C++

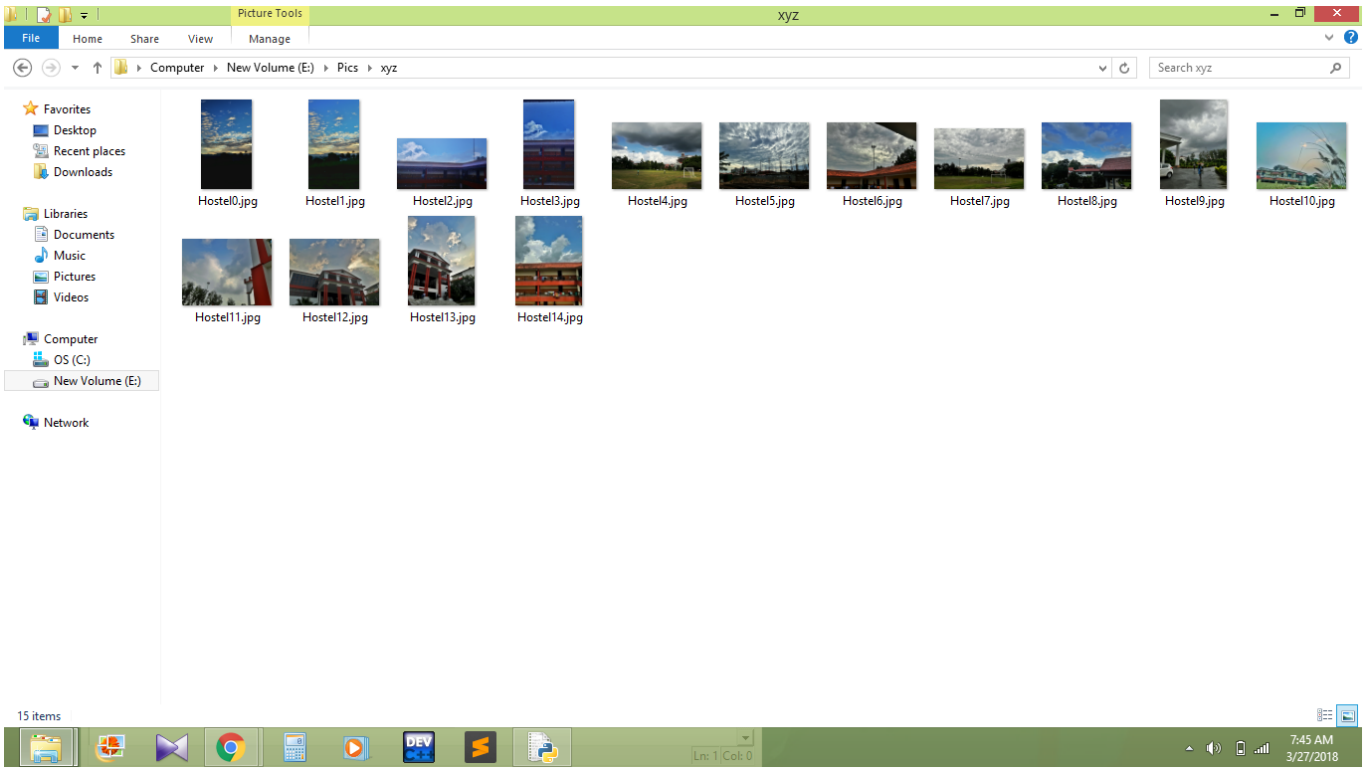
Note: While Python's os.rename() is commonly used, this project reimplements the logic using C++'s filesystem library (<filesystem> in C++17 and later).

Working Mechanism

- Traverse the folder to list all files.
- Assign a new file name based on a pattern like hostel1, hostel2, ...
- Construct full source and destination paths.
- Use rename functions to rename each file.

Expected Output

Renamed Folder Content (After Renaming):



Technologies Used

- C++17
- <filesystem> for file handling

Rename Multiple Files Using C++

- Basic command-line operations

Conclusion

This utility saves time and reduces error by automating repetitive file renaming. It's especially useful when working with large datasets like images for machine learning, organizing photos, or managing documents.

C++ Code Example

```
#include <iostream>
#include <filesystem>
#include <string>

namespace fs = std::filesystem;

int main() {
    std::string path = "E:/Pics/xyz";
    int counter = 0;

    for (const auto& entry : fs::directory_iterator(path)) {
        if (entry.is_regular_file()) {
            std::string new_name = "Hostel" + std::to_string(counter++) + ".jpg";
            fs::rename(entry.path(), entry.path().parent_path() / new_name);
        }
    }

    std::cout << "Files renamed successfully!" << std::endl;
    return 0;
}
```