Project Report: Project Manager using Function Calls (Windows C++)

Introduction

This project is a simple Project Management tool built using C++ for Windows. It helps users manage folders and files like creating, writing, reading, renaming, and deleting them, all through a text-based menu. The main goal is to learn how to use basic file-handling functions provided by the Windows operating system.

Modules and Functionality

The program shows a menu and lets the user choose what they want to do. Each option performs an action related to managing files and folders.

1. **Create Project Folder**
2. **Create File**
3. **Write Content to File**
4. **Read Content from File**
5. **Rename File**
6. **Delete File**
7. **Search Files in Project**
8. **Backup Project Folder**

System Call Explanations

1. CreateDirectory()
This function asks Windows to create a new folder. Think of it like making a new folder on your desktop. If the folder already exists, it won't create a new one.

2. CreateFile() or ofstream
These are used to make new files. It's like opening a blank notebook where you can start writing. This is used when the user selects "Add File" in the menu.

3. ofstream << (Write to File)
Once the file is created, this command writes content into it. Think of it like writing notes in your notebook.

4. ifstream >> (Read from File)
This command reads and displays the contents of a file. It's like opening a notebook and reading what you wrote earlier.

5. MoveFile()
This function renames a file. You're not changing the contents, just giving the file a new name—like putting a new label on your notebook.

6. DeleteFile()
This tells Windows to delete a file permanently. It's like throwing your notebook in the trash—once deleted, it's gone.

7. filesystem::directory_iterator

This is used to look inside the project folder and list all the files. It’s like opening a folder and seeing what’s inside.

8. system("powershell Compress-Archive")
This command compresses (zips) your project folder. It’s like putting all your notebooks in a zipped bag so you can carry or back them up easily.

Menu Structure

The user sees a menu with options from 1 to 9. They can choose actions like creating a folder, adding a file, or backing up the project. Each menu option connects to a specific function that does the job behind the scenes.

Code Structure

The code is organized into different files to keep things neat and understandable:
- **main.cpp**: Shows the menu and handles user choices.
- **project_ops.cpp**: Contains the logic for creating, reading, writing, etc.
- **project_ops.hpp**: Lists the function names used in the project.
- **README.md**: Explains how the project works.

Future Enhancements

- Add a user-friendly interface with buttons (GUI).
- Keep a history of all file changes.
- Add user login for project security.
- Allow undo and redo actions.

Conclusion

This project helped explore how file operations work in a computer using C++. By connecting system-level functions with real-life tasks like making folders or saving notes, it makes learning about the Windows operating system simple and practical.