

FILE EXPLORER APPLICATION

- CODE:

```
#include <iostream>
#include <filesystem>
#include <fstream>
#include <string>
#include <cstdlib>
#include <unistd.h>
#include <sys/stat.h>
#include <sstream>
#include <regex>

namespace fs = std::filesystem;

// ----- Utility Functions -----
void listFiles(const fs::path& path) {
    std::cout << "\n📂 Files in: " << path << "\n";
    for (const auto& entry : fs::directory_iterator(path)) {
        if (entry.is_directory())
            std::cout << "[DIR] ";
        else
            std::cout << "         ";
        std::cout << entry.path().filename().string() << "\n";
    }
}

void changeDirectory(fs::path& currentPath, const std::string& folder) {
    fs::path newPath = currentPath / folder;
    if (fs::exists(newPath) && fs::is_directory(newPath))
        currentPath = fs::canonical(newPath);
    else
        std::cout << "✗ Folder not found!\n";
}

void createFolder(const fs::path& path) {
    if (fs::create_directory(path))
        std::cout << "✓ Folder created: " << path << "\n";
    else
        std::cout << "✗ Could not create folder.\n";
}

void createFile(const fs::path& path) {
```

```

    std::ofstream file(path);
    if (file)
        std::cout << "☑ File created: " << path << "\n";
    else
        std::cout << "☒ Could not create file.\n";
}

void deleteItem(const fs::path& path) {
    if (fs::remove_all(path))
        std::cout << "☒ Deleted: " << path << "\n";
    else
        std::cout << "☒ Item not found.\n";
}

void copyItem(const fs::path& src, const fs::path& dest) {
    try {
        fs::copy(src, dest, fs::copy_options::recursive |
        fs::copy_options::overwrite_existing);
        std::cout << "☑ Copied to " << dest << "\n";
    } catch (const std::exception& e) {
        std::cout << "☒ Copy failed: " << e.what() << "\n";
    }
}

void moveItem(const fs::path& src, const fs::path& dest) {
    try {
        fs::rename(src, dest);
        std::cout << "☑ Moved to " << dest << "\n";
    } catch (const std::exception& e) {
        std::cout << "☒ Move failed: " << e.what() << "\n";
    }
}

void searchFile(const fs::path& dir, const std::string& name) {
    std::cout << "🔍 Searching for \" " << name << "\" in " << dir << "\n";
    for (auto& entry : fs::recursive_directory_iterator(dir)) {
        if (entry.path().filename().string().find(name) != std::string::npos)
            std::cout << "    Found: " << entry.path() << "\n";
    }
}

// ---- Numeric-only chmod ---- //
void setPermissions(const fs::path& path, const std::string& modeStr) {
    if (!std::regex_match(modeStr, std::regex("^[0-7]{3}$")))
        std::cout << "☒ Invalid mode! Use numeric values like 400, 644, 755, 777.\n";
    return;
}

```

```

mode_t mode = std::stoi(modeStr, nullptr, 8); // Convert octal string to integer

if (chmod(path.c_str(), mode) == 0)
    std::cout << "✓ Permissions updated for " << path << " (mode " << modeStr <<
")\n";
else
    perror("✗ chmod failed");
}

void showHelp() {
    std::cout << R"
=====
    └── Linux File Explorer Application (C++)
=====

    ~Commands:~

    ls                  - List files and folders
    ls -l <file>        - View details
    cd <folder>         - Go inside a folder
    back                - Go to previous folder
    mkdir <name>         - Create new folder
    touch <name>         - Create new file
    del <name>           - Delete file or folder
    copy <src> <dest>    - Copy file/folder
    move <src> <dest>    - Move file/folder
    search <name>        - Search file/folder
    chmod <file> <mode>   - Change file permissions
                           (e.g., 400, 644, 755, 777)
    clear               - Clear screen
    help                - Show commands
    exit                - Quit program

=====
);

// ----- Main Program ----- //

int main() {
    fs::path currentPath = fs::current_path();
    std::string command;

    std::cout << "      ⚡ Welcome to Linux File Explorer Application!      \n";
    std::cout << "Type 'help' for available commands.\n";

    while (true) {
        std::cout << "\n[" << currentPath << "]$ ";
}

```

```

    std::getline(std::cin, command);

    if (command == "exit") break;
    else if (command == "help") showHelp();
    else if (command == "ls") listFiles(currentPath);
    else if (command.rfind("ls ", 0) == 0) {
        std::string fullCmd = "cd \\" + currentPath.string() + "\"" && " " + command;
        system(fullCmd.c_str());
    }
    else if (command.rfind("cd ", 0) == 0) changeDirectory(currentPath,
command.substr(3));
    else if (command == "back") currentPath = currentPath.parent_path();
    else if (command.rfind("mkdir ", 0) == 0) createFolder(currentPath /
command.substr(6));
    else if (command.rfind("touch ", 0) == 0) createFile(currentPath /
command.substr(6));
    else if (command.rfind("del ", 0) == 0) deleteItem(currentPath /
command.substr(4));
    else if (command.rfind("copy ", 0) == 0) {
        std::istringstream iss(command.substr(5));
        std::string src, dest; iss >> src >> dest;
        copyItem(currentPath / src, currentPath / dest);
    }
    else if (command.rfind("move ", 0) == 0) {
        std::istringstream iss(command.substr(5));
        std::string src, dest; iss >> src >> dest;
        moveItem(currentPath / src, currentPath / dest);
    }
    else if (command.rfind("search ", 0) == 0) searchFile(currentPath,
command.substr(7));
    else if (command.rfind("chmod ", 0) == 0) {
        std::istringstream iss(command.substr(6));
        std::string name, mode; iss >> name >> mode;
        setPermissions(currentPath / name, mode);
    }
    else if (command == "clear") system("clear");
    else {
        std::string fullCmd = "cd \\" + currentPath.string() + "\"" && " " + command;
        int ret = system(fullCmd.c_str());
        if (ret != 0)
            std::cout << "? Unknown or failed command.\n";
    }
}

std::cout << "👋 Exiting File Explorer. Goodbye!\n";
return 0;
}

```

● SCREENSHOTS:

Before Entering to any commands

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with a project named 'CAPSTONE_PROJECT' containing files '.vscode', 'explorer', and 'FileExplorer.cpp'. The 'FileExplorer.cpp' file is selected and shown in the main code editor area. The code includes standard library includes like iostream, filesystem, fstream, string, cstdlib, unistd.h, sys/stat.h, sstream, regex, and std::filesystem. It also defines a namespace fs and utility functions. Below the code editor is a terminal window displaying a Linux command-line session. The user runs 'g++ -std=c++17 FileExplorer.cpp -o explorer' and then './explorer'. The application starts and prints a welcome message: 'Welcome to Linux File Explorer Application! Type 'help' for available commands.' The terminal prompt is shown as '/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project\$'.

After Entering 'help' command

This screenshot is similar to the previous one but shows the result of entering the 'help' command in the terminal. The terminal output now displays the available commands for the 'Linux File Explorer Application (C++)'. The commands listed include ls, cd, back, mkdir, touch, del, copy, move, search, chmod, clear, help, and exit, each with a brief description. The terminal prompt is shown as '/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project\$'.

```
=====
■ Linux File Explorer Application (C++)
=====

~Commands:~
ls          - List files and folders
ls -l <file>   - View details
cd <folder>    - Go inside a folder
back        - Go to previous folder
mkdir <name>   - Create new folder
touch <name>   - Create new file
del <name>     - Delete file or folder
copy <src> <dest> - Copy file/folder
move <src> <dest> - Move file/folder
search <name>  - Search file/folder
chmod <file> <mode> - Change file permissions
(e.g., 400, 644, 755, 777)
clear       - Clear screen
help        - Show commands
exit        - Quit program
=====
```

Ls , Back ,Cd Commands

The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following command-line session:

```
=====
["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ ls
[DIR] .vscode
explorer
FileExplorer.cpp

["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ back
["/mnt/c/Users/prati/Desktop/Pratiti"]$ cd CapStone_Project
["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ mkdir hello
[✓] Folder created: "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project/hello"

["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$
```

Commands for Creating File And Folder

The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following command-line session:

```
=====
FileExplorer.cpp

["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ back
["/mnt/c/Users/prati/Desktop/Pratiti"]$ cd CapStone_Project
["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ mkdir hello
[✓] Folder created: "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project/hello"

["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ touch yaya.cpp
[✓] File created: "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project/yaya.cpp"

["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$
```

Delete Command

A screenshot of the Visual Studio Code interface. The Explorer sidebar shows a project named 'CAPSTONE_PROJECT' containing files '.vscode', 'explorer', 'FileExplorer.cpp', and 'yaya.cpp'. The 'File Explorer' tab is selected in the top bar. In the center, the code editor shows 'FileExplorer.cpp' with the following content:

```
#include <iostream>
#include <filesystem>
#include <fstream>
#include <string>
#include <cstdlib>
#include <unistd.h> // For chmod and access()
#include <sys/stat.h> // For file permissions
#include <sstream> // For parsing copy/move commands
#include <regex>
namespace fs = std::filesystem;
```

The terminal at the bottom shows the following session:

```
["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ back
["/mnt/c/Users/prati/Desktop/Pratiti"]$ cd Capstone_Project
["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ mkdir hello
Folder created: "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project/hello"
["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ touch yaya.cpp
File created: "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project/yaya.cpp"
["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ del hello
Deleted: "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project/hello"
["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$
```

Copy Command

A screenshot of the Visual Studio Code interface. The Explorer sidebar shows a project named 'CAPSTONE_PROJECT' containing files '.vscode', 'explorer', 'FileExplorer.cpp', 'hii.cpp', and 'yaya.cpp'. The 'File Explorer' tab is selected in the top bar. In the center, the code editor shows 'FileExplorer.cpp' with the following content:

```
#include <iostream>
#include <filesystem>
#include <fstream>
#include <string>
#include <cstdlib>
#include <unistd.h>
#include <sys/stat.h>
#include <sstream>
#include <regex>
namespace fs = std::filesystem;

// ----- Utility Functions -----
void listFiles(const fs::path& path) {
    std::cout << "\n📁 Files in: " << path << "\n";
    for (const auto& entry : fs::directory_iterator(path)) {
        if (entry.is_directory()) {
```

The terminal at the bottom shows the following session:

```
["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ copy FileExplorer.cpp yaya.cpp
Copied to "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project/yaya.cpp"
["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$
```

A screenshot of the Visual Studio Code interface. The Explorer sidebar shows a project named 'CAPSTONE_PROJECT' containing files '.vscode', 'explorer', 'FileExplorer.cpp', 'hii.cpp', and 'yaya.cpp'. The 'File Explorer' tab is selected in the top bar. In the center, the code editor shows 'yaya.cpp' with the following content:

```
#include <iostream>
#include <filesystem>
#include <fstream>
#include <string>
#include <cstdlib>
#include <unistd.h>
#include <sys/stat.h>
#include <sstream>
#include <regex>
namespace fs = std::filesystem;

// ----- Utility Functions -----
void listFiles(const fs::path& path) {
    std::cout << "\n📁 Files in: " << path << "\n";
    for (const auto& entry : fs::directory_iterator(path)) {
        if (entry.is_directory()) {
```

The terminal at the bottom shows the following session:

```
["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ copy FileExplorer.cpp yaya.cpp
Copied to "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project/yaya.cpp"
["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$
```

Move command

A screenshot of the Visual Studio Code interface. The title bar shows "CapStone_Project". The Explorer sidebar on the left has a project named "CAPSTONE_PROJECT" expanded, showing files like ".vscode", "explorer", "FileExplorer.cpp", and "hii.cpp". The current file being edited is "hii.cpp". The code editor displays the following C++ code:

```
#include <iostream>
#include <filesystem>
#include <fstream>
#include <string>
#include <cstdlib>
#include <unistd.h>
#include <sys/stat.h>
#include <sstream>
#include <regex>

namespace fs = std::filesystem;

// ----- Utility Functions -----
void listFiles(const fs::path& path) {
    std::cout << "\n📁 Files in: " << path << "\n";
    for (const auto& entry : fs::directory_iterator(path)) {
        if (entry.is_directory())
    }
}
```

The terminal at the bottom shows the command "move yaya.cpp hii.cpp" being run, with a success message: "Moved to "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project/hii.cpp"

Search Command

A screenshot of the Visual Studio Code interface. The title bar shows "CapStone_Project". The Explorer sidebar on the left has a project named "CAPSTONE_PROJECT" expanded, showing files like ".vscode", "explorer", "FileExplorer.cpp", and "hii.cpp". The current file being edited is "hii.cpp". The code editor displays the same C++ code as the previous screenshot.

The terminal at the bottom shows the command "search yaya.cpp" being run, followed by "searching for "yaya.cpp" in "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"". It then shows the command "search hii.cpp" being run, followed by "Searching for "hii.cpp" in "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project" Found: "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project/hii.cpp"

Permission Features (Chmod command)

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows a project named "CAPSTONE_PROJECT" containing files ".vscode", "explorer", "FileExplorer.cpp", and "hii.cpp".
- Editor:** The "hii.cpp" file is open, displaying C++ code related to file permissions and chmod.
- Terminal:** The terminal window shows the execution of the chmod command on "FileExplorer.cpp" to mode 777, followed by an "ls -l" command confirming the file permissions.

```
// ---- Numeric-only chmod ---- //
void setPermissions(const fs::path& path, const std::string& modeStr) {
    if (!std::regex_match(modeStr, std::regex("^[0-7]{3}$"))) {
        std::cout << "X Invalid mode! Use numeric values like 400, 644, 755, 777.\n";
        return;
    }

    mode_t mode = std::stoi(modeStr, nullptr, 8); // Convert octal string to integer

    if (chmod(path.c_str(), mode) == 0)
        std::cout << "✓ Permissions updated for " << path << " (mode " << modeStr << ")\n";
    else
        perror("X chmod failed");
}
```

```
["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ chmod FileExplorer.cpp 777
✓ Permissions updated for "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project/FileExplorer.cpp" (mode 777)

["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ ls -l FileExplorer.cpp
-rwxrwxrwx 1 pratichi pratichi 6700 Nov 8 17:22 FileExplorer.cpp

["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ chmod FileExplorer.cpp 400
✓ Permissions updated for "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project/FileExplorer.cpp" (mode 400)

["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ ls -l FileExplorer.cpp
-rxr-xr-x 1 pratichi pratichi 6700 Nov 8 17:22 FileExplorer.cpp
```

Exit Command

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows a project named "CAPSTONE_PROJECT" containing files ".vscode", "explorer", "FileExplorer.cpp", and "hii.cpp".
- Editor:** The "hii.cpp" file is open, displaying C++ code related to file permissions and chmod.
- Terminal:** The terminal window shows the execution of the chmod command on "FileExplorer.cpp" to mode 400, followed by an "ls -l" command, then the "exit" command, and finally a goodbye message from the application.

```
void searchFile(const fs::path& dir, const std::string& name) {
}

// ---- Numeric-only chmod ---- //
void setPermissions(const fs::path& path, const std::string& modeStr) {
    if (!std::regex_match(modeStr, std::regex("^[0-7]{3}$"))) {
        std::cout << "X Invalid mode! Use numeric values like 400, 644, 755, 777.\n";
        return;
    }

    mode_t mode = std::stoi(modeStr, nullptr, 8); // Convert octal string to integer
```

```
✓ Permissions updated for "/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project/FileExplorer.cpp" (mode 400)

["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ ls -l FileExplorer.cpp
-rxr-xr-x 1 pratichi pratichi 6700 Nov 8 17:22 FileExplorer.cpp

["/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project"]$ exit
✖ Exiting File Explorer. Goodbye!
pratichi@PRATICHI:/mnt/c/Users/prati/Desktop/Pratiti/CapStone_Project$
```