File Explorer Application with Enhanced File History and Log Tracking

```
/*
    ------------------------------------------------------------
-
    SIMPLE CONSOLE FILE EXPLORER (WITH ACTIVITY LOG)
    ------------------------------------------------------------
-
    Author  : Abhisekh k Panigrahy
    Purpose : A small command-line file explorer that allows
              users to perform common file operations such as
              create, delete, copy, move, list, and search.
              It also keeps a log of all performed actions.
    ------------------------------------------------------------
-
*/

#include <iostream>
#include <fstream>
#include <string>
#include <vector>
#include <sstream>
#include <dirent.h>
#include <sys/stat.h>
#include <cstdio>
#include <cstring>
#include <cstdlib>
#include <ctime>
#include <unistd.h>

#ifdef _WIN32
#include <windows.h>
#define PATH_SEP '\\'
#else
#define PATH_SEP '/'
#endif
```

```cpp
using namespace std;

/*-------------------------------------------------------------
    Write each action performed by user to a log file
-------------------------------------------------------------*/
void logAction(const string &action) {
    ofstream log("activity_log.txt", ios::app);
    if (!log) return;

    time_t now = time(NULL);
    tm *t = localtime(&now);

    char timeStr[64];
    strftime(timeStr, sizeof(timeStr), "%Y-%m-%d %H:%M:%S", t);

    log << "[" << timeStr << "] " << action << endl;
    log.close();
}

/*-------------------------------------------------------------
    Split input line into words
-------------------------------------------------------------*/
vector<string> split(const string &line) {
    stringstream ss(line);
    string word;
    vector<string> parts;
    while (ss >> word)
        parts.push_back(word);
    return parts;
}

/*-------------------------------------------------------------
    Check if path refers to a directory
-------------------------------------------------------------*/
bool isDirectory(const string &path) {
    struct stat st;
    if (stat(path.c_str(), &st) == 0)
        return (st.st_mode & S_IFDIR);
    return false;
```

```
}

/*------------------------------------------------------------
    List all files and folders in a directory
------------------------------------------------------------*/
void listFiles(const string &path = ".") {
    DIR *dir = opendir(path.c_str());
    if (!dir) {
        perror("ls");
        return;
    }

    cout << "Contents of " << path << ":\n";
    struct dirent *entry;
    while ((entry = readdir(dir)) != NULL) {
        string name = entry->d_name;
        if (name == "." || name == "..") continue;

        string fullPath = path + PATH_SEP + name;
        struct stat st;
        if (stat(fullPath.c_str(), &st) == 0) {
            if (isDirectory(fullPath))
                cout << "[DIR]  ";
            else
                cout << "       ";
            cout << name << "\t(" << st.st_size << " bytes)\n";
        }
    }
    closedir(dir);
    logAction("Listed contents of: " + path);
}

/*------------------------------------------------------------
    Change current working directory
------------------------------------------------------------*/
void changeDir(const string &path) {
    if (chdir(path.c_str()) == 0) {
        cout << "Changed directory to: " << path << endl;
        logAction("Changed directory to: " + path);
    } else {
```

```
        perror("cd");
    }
}

/*---------------------------------------------------------------
    Print current working directory path
---------------------------------------------------------------*/
void printPwd() {
    char cwd[1024];
    if (getcwd(cwd, sizeof(cwd)))
        cout << cwd << endl;
    else
        perror("pwd");

    logAction("Checked current directory.");
}

/*---------------------------------------------------------------
    Copy a file from one location to another
---------------------------------------------------------------*/
bool copyFile(const string &src, const string &dest) {
    FILE *in = fopen(src.c_str(), "rb");
    if (!in) return false;

    FILE *out = fopen(dest.c_str(), "wb");
    if (!out) {
        fclose(in);
        return false;
    }

    char buffer[4096];
    size_t n;
    while ((n = fread(buffer, 1, sizeof(buffer), in)) > 0)
        fwrite(buffer, 1, n, out);

    fclose(in);
    fclose(out);
    logAction("Copied file: " + src + " -> " + dest);
    return true;
}
```

```
/*------------------------------------------------------------
    Delete a file or folder (recursively)
------------------------------------------------------------*/
void removeRecursive(const string &path) {
    if (isDirectory(path)) {
        DIR *dir = opendir(path.c_str());
        if (!dir) return;
        struct dirent *entry;

        while ((entry = readdir(dir)) != NULL) {
            string name = entry->d_name;
            if (name == "." || name == "..") continue;
            string subPath = path + PATH_SEP + name;
            removeRecursive(subPath);
        }
        closedir(dir);

#ifdef _WIN32
        _rmdir(path.c_str());
#else
        rmdir(path.c_str());
#endif
    } else {
        remove(path.c_str());
    }
    logAction("Removed: " + path);
}

/*------------------------------------------------------------
    Search for a file by name (recursive)
------------------------------------------------------------*/
/*------------------------------------------------------------
    Search for a file by name (recursive)
    ✅ Shows full absolute path
    ✅ Displays message if no match is found
------------------------------------------------------------*/
void searchFile(const string &pattern, const string &path =
".") {
    DIR *dir = opendir(path.c_str());
```

```
    if (!dir) return;

    struct dirent *entry;
    bool found = false;  // Track if any file matched

    while ((entry = readdir(dir)) != NULL) {
        string name = entry->d_name;
        if (name == "." || name == "..") continue;

        string fullPath = path + PATH_SEP + name;
        struct stat st;

        if (stat(fullPath.c_str(), &st) != 0)
            continue;

        // If name matches pattern, print absolute path
        if (name.find(pattern) != string::npos) {
            found = true;

            char absPath[1024];
#ifdef _WIN32
            _fullpath(absPath, fullPath.c_str(),
sizeof(absPath));
#else
            realpath(fullPath.c_str(), absPath);
#endif
            cout << absPath << endl;
        }

        // Continue searching inside subdirectories
        if (isDirectory(fullPath))
            searchFile(pattern, fullPath);
    }

    closedir(dir);

    // If nothing found and this is the top-level call
    if (path == "." && !found)
        cout << "No file found matching: " << pattern << endl;
```

```
    logAction("Searched for: " + pattern + " in " + path);
}

/*------------------------------------------------------------
    Create an empty file (similar to Linux 'touch')
------------------------------------------------------------*/
void touchFile(const string &path) {
    FILE *f = fopen(path.c_str(), "ab");
    if (f) {
        fclose(f);
        cout << "File created/updated: " << path << endl;
        logAction("Created or updated file: " + path);
    } else {
        perror("touch");
    }
}

/*------------------------------------------------------------
    Create a new folder
------------------------------------------------------------*/
void makeDir(const string &path) {
#ifdef _WIN32
    if (CreateDirectoryA(path.c_str(), NULL))
        cout << "Directory created: " << path << endl;
    else
        perror("mkdir");
#else
    if (mkdir(path.c_str(), 0755) == 0)
        cout << "Directory created: " << path << endl;
    else
        perror("mkdir");
#endif
    logAction("Created directory: " + path);
}

/*------------------------------------------------------------
    Move or rename a file
------------------------------------------------------------*/
void moveFile(const string &src, const string &dest) {
    if (rename(src.c_str(), dest.c_str()) == 0) {
```

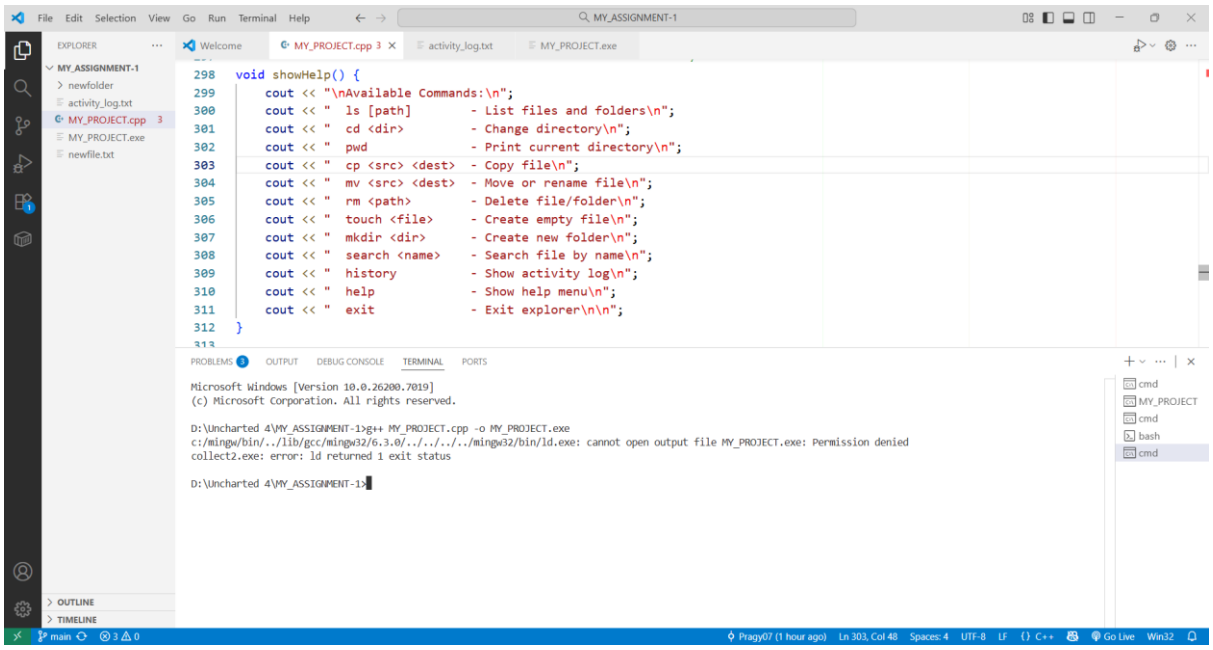
```
        cout << "Moved: " << src << " -> " << dest << endl;
        logAction("Moved/Renamed: " + src + " -> " + dest);
    } else {
        perror("mv");
    }
}

/*-------------------------------------------------------------
    Show all previously logged activities
-------------------------------------------------------------*/
void showHistory() {
    ifstream log("activity_log.txt");
    if (!log) {
        cout << "No activity history found yet.\n";
        return;
    }

    cout << "----------- ACTIVITY LOG -----------\n";
    string line;
    while (getline(log, line))
        cout << line << endl;
    cout << "------------------------------------\n";
    log.close();
}

/*-------------------------------------------------------------
    Display list of available commands
-------------------------------------------------------------*/
void showHelp() {
    cout << "\nAvailable Commands:\n";
    cout << "  ls [path]        - List files and folders\n";
    cout << "  cd <dir>         - Change directory\n";
    cout << "  pwd              - Print current directory\n";
    cout << "  cp <src> <dest>  - Copy file\n";
    cout << "  mv <src> <dest>  - Move or rename file\n";
    cout << "  rm <path>        - Delete file/folder\n";
    cout << "  touch <file>     - Create empty file\n";
    cout << "  mkdir <dir>      - Create new folder\n";
    cout << "  search <name>    - Search file by name\n";
    cout << "  history          - Show activity log\n";
```

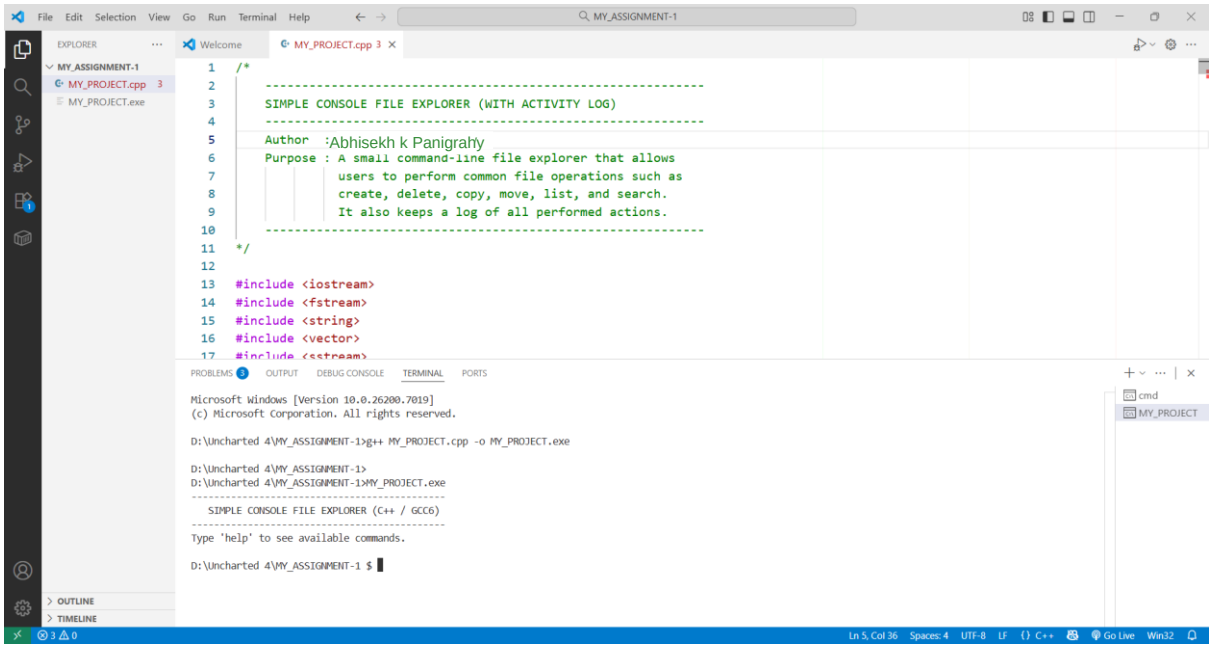
```
    cout << "  help             - Show help menu\n";
    cout << "  exit             - Exit explorer\n\n";
}

/*------------------------------------------------------------
    MAIN PROGRAM
------------------------------------------------------------*/
int main() {
    cout << "-----------------------------------------------\n";
    cout << "   SIMPLE CONSOLE FILE EXPLORER (C++ / GCC6)\n";
    cout << "-----------------------------------------------\n";
    cout << "Type 'help' to see available commands.\n\n";

    string line;
    while (true) {
        char cwd[1024];
        getcwd(cwd, sizeof(cwd));
        cout << cwd << " $ ";

        if (!getline(cin, line)) break;
        vector<string> args = split(line);
        if (args.empty()) continue;

        string cmd = args[0];

        if (cmd == "exit" || cmd == "quit")
            break;
        else if (cmd == "help")
            showHelp();
        else if (cmd == "ls")
            listFiles(args.size() > 1 ? args[1] : ".");
        else if (cmd == "cd") {
            if (args.size() > 1) changeDir(args[1]);
            else cout << "Usage: cd <dir>\n";
        }
        else if (cmd == "pwd")
            printPwd();
        else if (cmd == "cp") {
            if (args.size() > 2) {
                if (copyFile(args[1], args[2]))
```

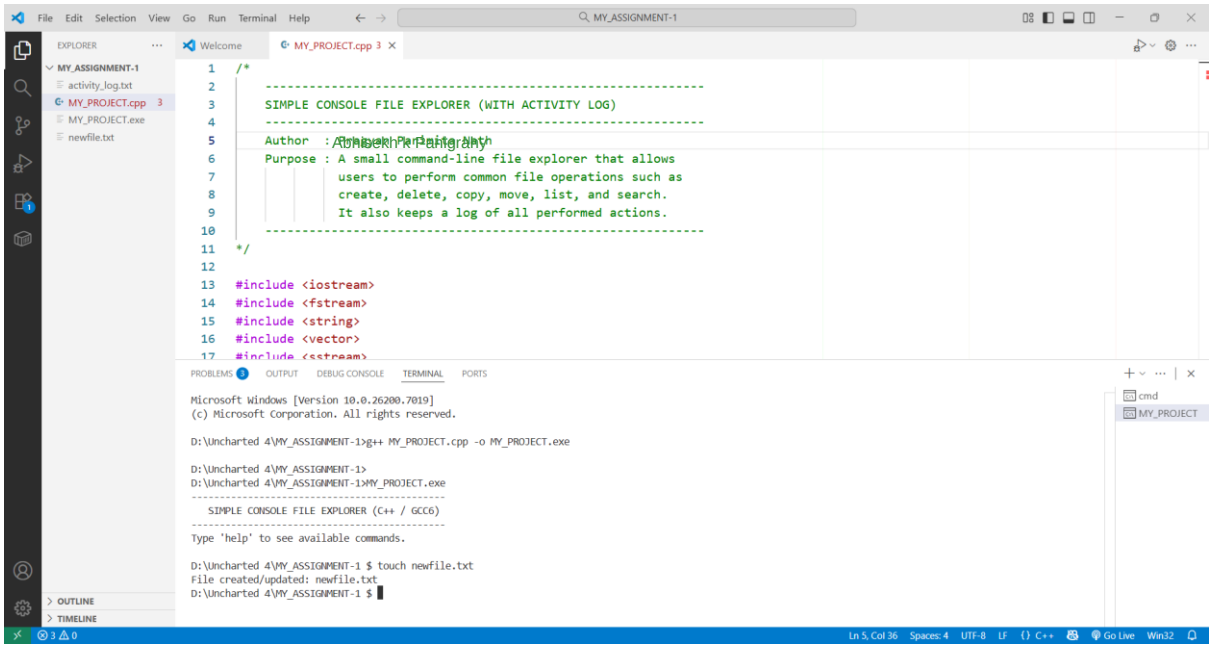
```
                    cout << "Copied: " << args[1] << " -> " <<
args[2] << endl;
                else
                    perror("cp");
            } else cout << "Usage: cp <src> <dest>\n";
        }
        else if (cmd == "mv") {
            if (args.size() > 2)
                moveFile(args[1], args[2]);
            else
                cout << "Usage: mv <src> <dest>\n";
        }
        else if (cmd == "rm") {
            if (args.size() > 1)
                removeRecursive(args[1]);
            else
                cout << "Usage: rm <path>\n";
        }
        else if (cmd == "touch") {
            if (args.size() > 1)
                touchFile(args[1]);
            else
                cout << "Usage: touch <file>\n";
        }
        else if (cmd == "mkdir") {
            if (args.size() > 1)
                makeDir(args[1]);
            else
                cout << "Usage: mkdir <dir>\n";
        }
        else if (cmd == "search") {
            if (args.size() > 1)
                searchFile(args[1]);
            else
                cout << "Usage: search <pattern>\n";
        }
        else if (cmd == "history")
            showHistory();
        else
            cout << "Unknown command. Type 'help' for list.\n";
```

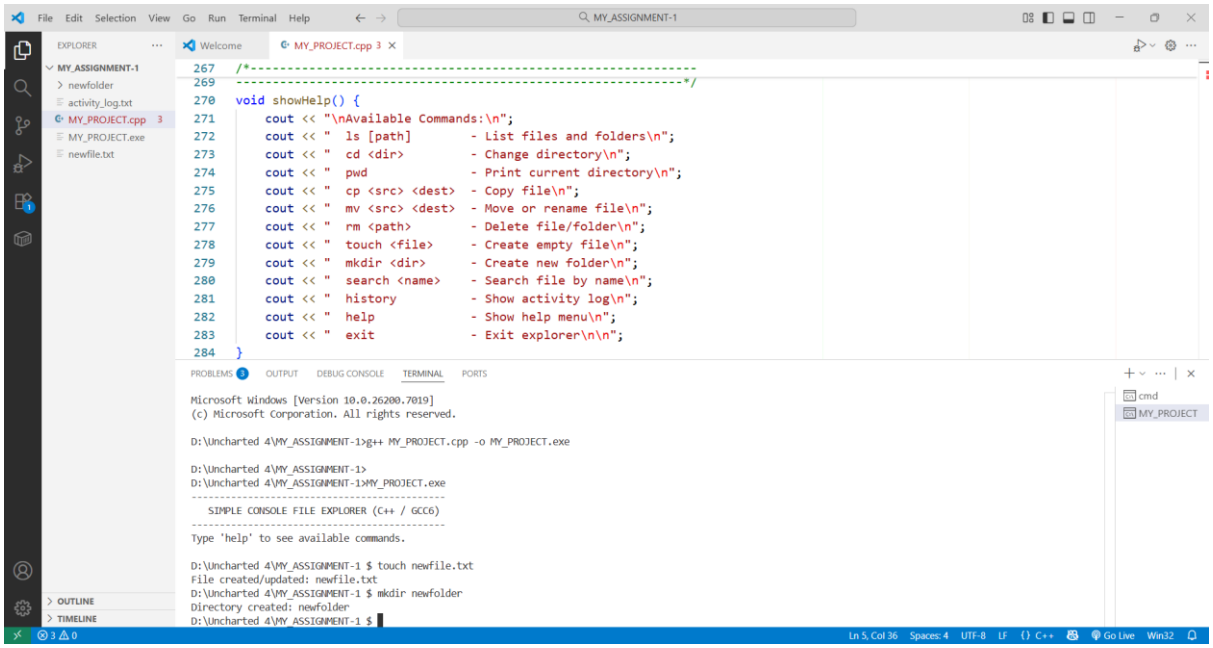
```
    }

    cout << "\nGoodbye! Have a nice day :)\n";
    return 0;
}
```

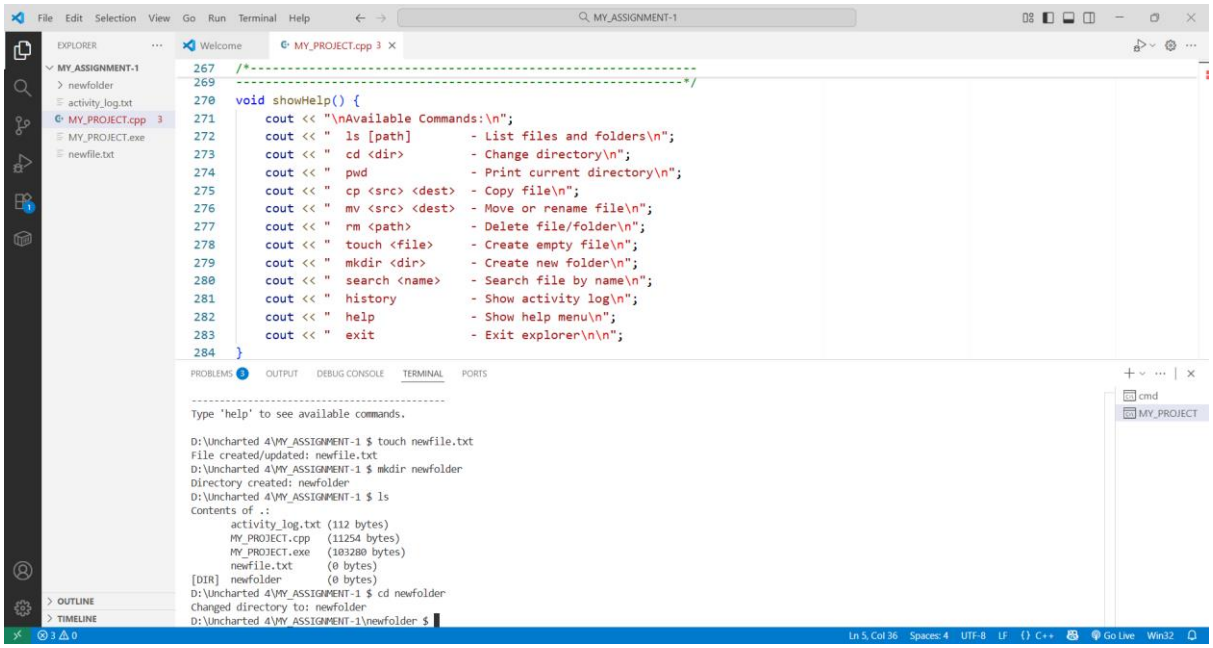
SCREENSHOTS

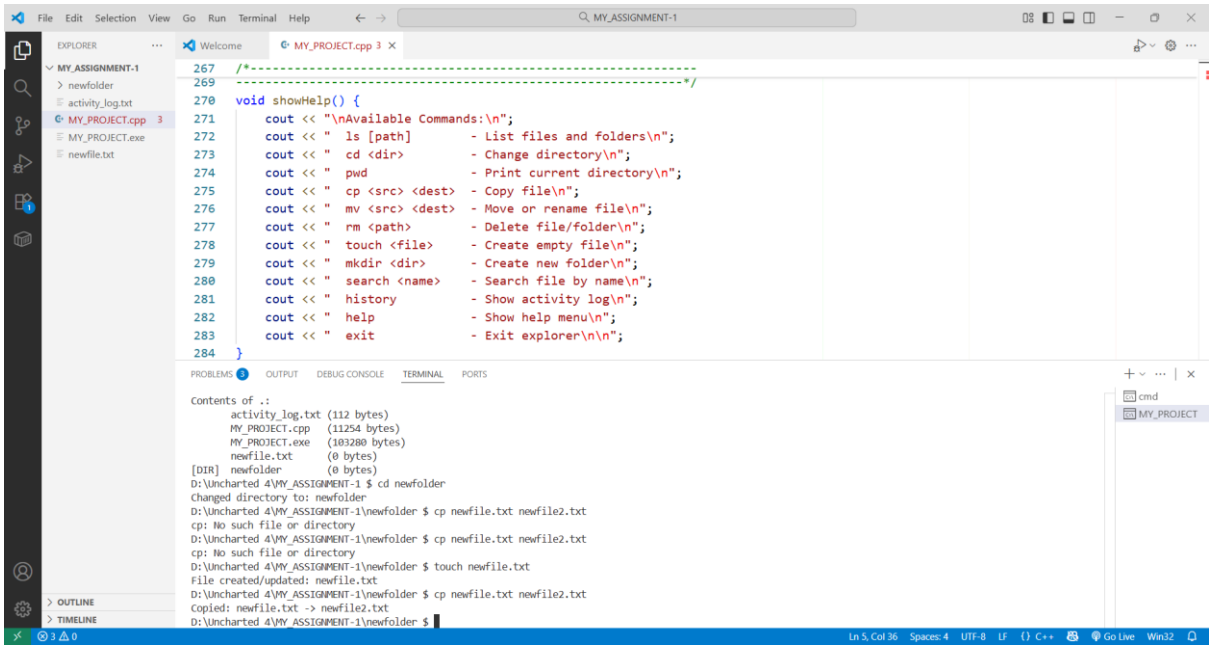


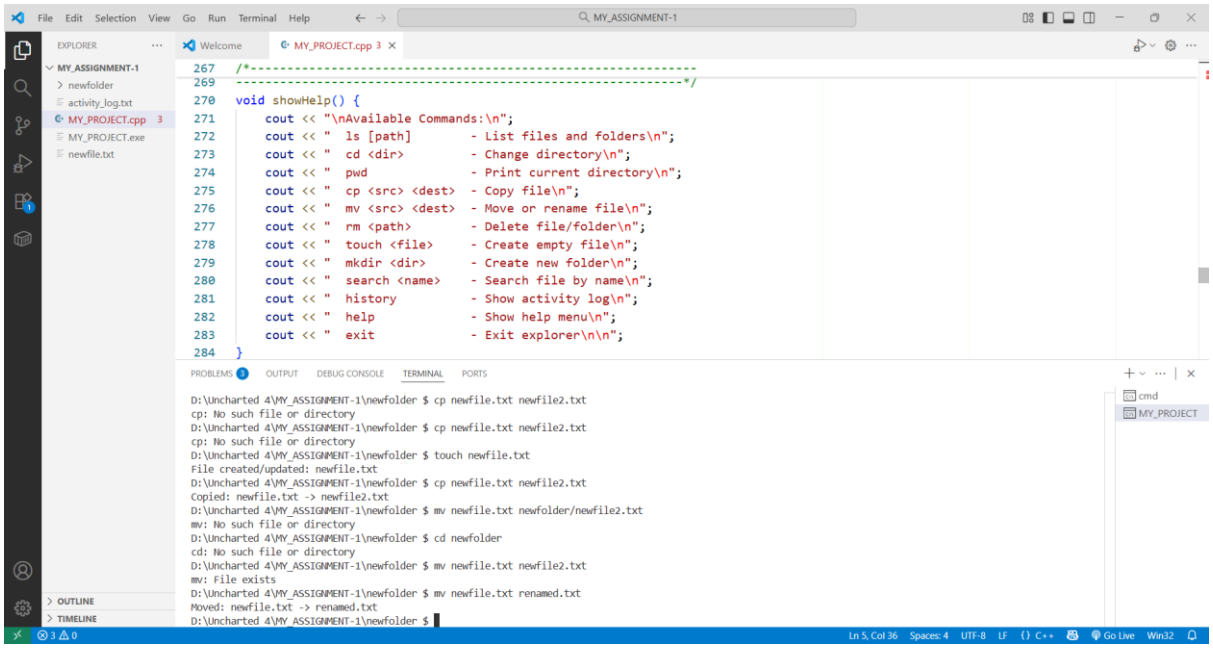


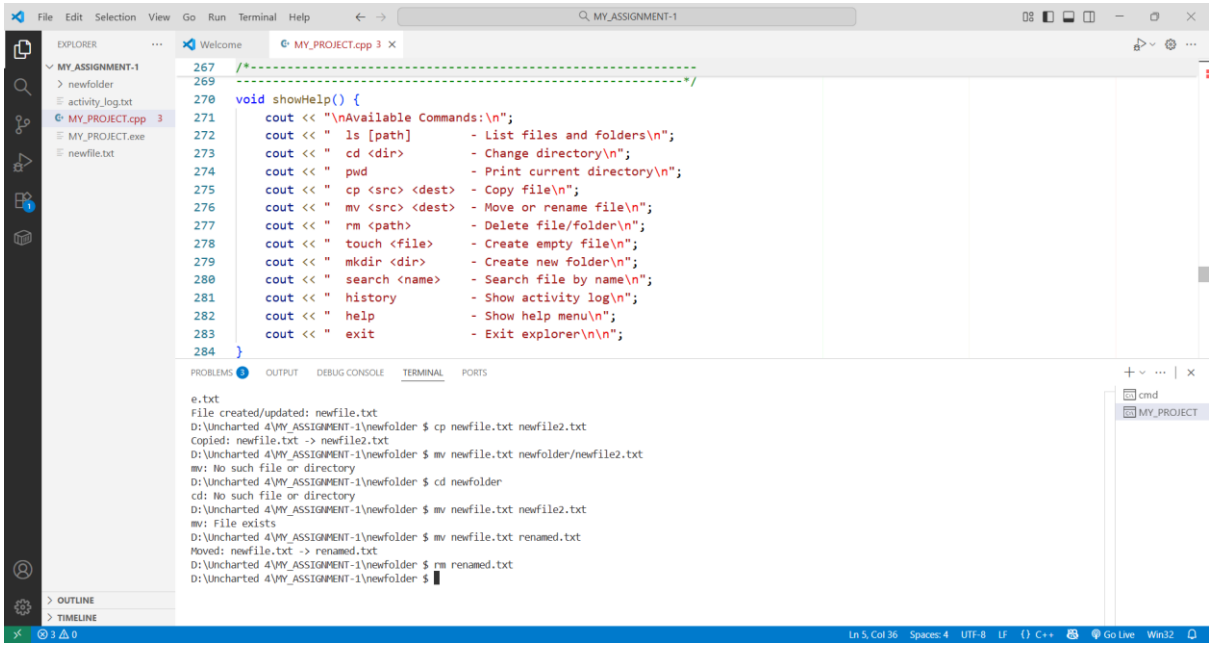

File Edit Selection View Go Run Terminal Help
MY_ASSIGNMENT-1
EXPLORER
MY_ASSIGNMENT-1
activity_log.txt
MY_PROJECT.cpp 3
MY_PROJECT.exe
newfile.txt
Welcome
MY_PROJECT.cpp 3
/*
SIMPLE CONSOLE FILE EXPLORER (WITH ACTIVITY LOG)
Author : Ashutosh R Tangirala
Purpose : A small command-line file explorer that allows
users to perform common file operations such as
create, delete, copy, move, list, and search.
It also keeps a log of all performed actions.
*/
#include <iostream>
#include <fstream>
#include <string>
#include <vector>
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Windows [Version 10.0.26200.7019]
(c) Microsoft Corporation. All rights reserved.
D:\Uncharted 4\MY_ASSIGNMENT-1>g++ MY_PROJECT.cpp -o MY_PROJECT.exe
D:\Uncharted 4\MY_ASSIGNMENT-1>
D:\Uncharted 4\MY_ASSIGNMENT-1>MY_PROJECT.exe
SIMPLE CONSOLE FILE EXPLORER (C++ / GCC6)
Type 'help' to see available commands.
D:\Uncharted 4\MY_ASSIGNMENT-1 $ touch newfile.txt
File created/updated: newfile.txt
D:\Uncharted 4\MY_ASSIGNMENT-1 $
cmd
MY_PROJECT
OUTLINE
TIMELINE
Ln 5, Col 36 Spaces: 4 UTF-8 LF C++ Go Live Win32

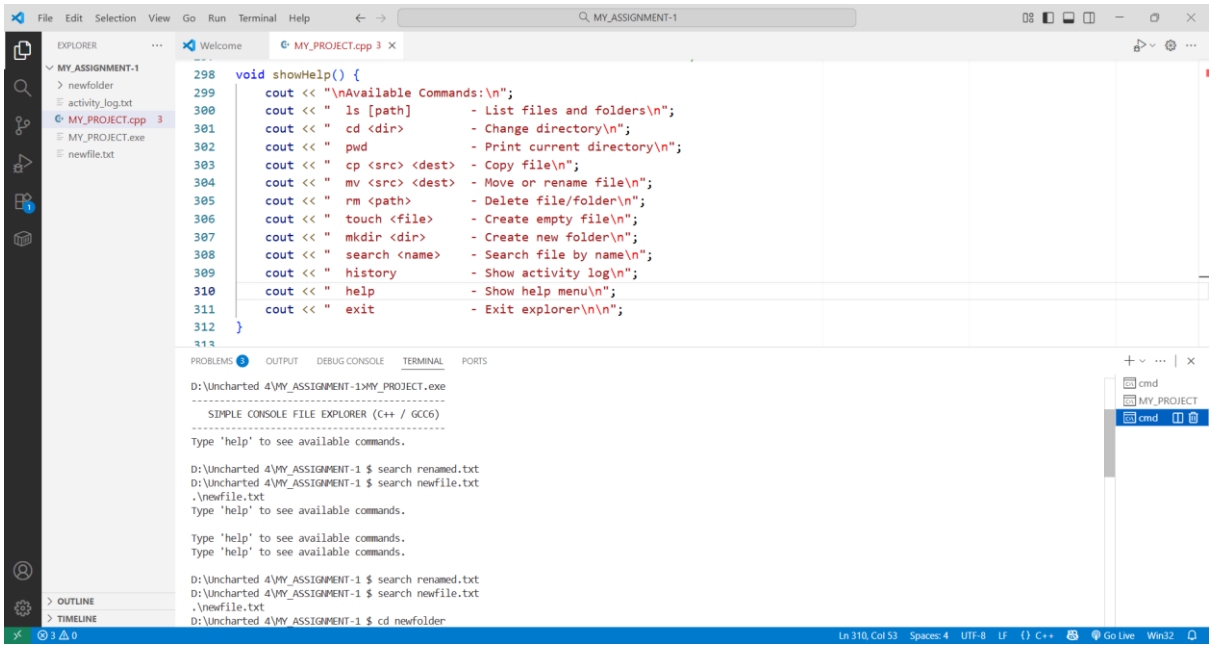

File Edit Selection View Go Run Terminal Help
MY_ASSIGNMENT-1
EXPLORER
MY_ASSIGNMENT-1
newfolder
activity_log.txt
MY_PROJECT.cpp 3
MY_PROJECT.exe
newfile.txt
Welcome
MY_PROJECT.cpp 3
/*----------*/
void showHelp() {
cout << "\nAvailable Commands:\n";
cout << " ls [path] - List files and folders\n";
cout << " cd <dir> - Change directory\n";
cout << " pwd - Print current directory\n";
cout << " cp <src> <dest> - Copy file\n";
cout << " mv <src> <dest> - Move or rename file\n";
cout << " rm <path> - Delete file/folder\n";
cout << " touch <file> - Create empty file\n";
cout << " mkdir <dir> - Create new folder\n";
cout << " search <name> - Search file by name\n";
cout << " history - Show activity log\n";
cout << " help - Show help menu\n";
cout << " exit - Exit explorer\n\n";
}
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Windows [Version 10.0.26200.7019]
(c) Microsoft Corporation. All rights reserved.
D:\Uncharted 4\MY_ASSIGNMENT-1>g++ MY_PROJECT.cpp -o MY_PROJECT.exe
D:\Uncharted 4\MY_ASSIGNMENT-1>
D:\Uncharted 4\MY_ASSIGNMENT-1>MY_PROJECT.exe
SIMPLE CONSOLE FILE EXPLORER (C++ / GCC6)
Type 'help' to see available commands.
D:\Uncharted 4\MY_ASSIGNMENT-1 $ touch newfile.txt
File created/updated: newfile.txt
D:\Uncharted 4\MY_ASSIGNMENT-1 $ mkdir newfolder
Directory created: newfolder
D:\Uncharted 4\MY_ASSIGNMENT-1 $
cmd
MY_PROJECT
OUTLINE
TIMELINE
Ln 5, Col 36 Spaces: 4 UTF-8 LF C++ Go Live Win32

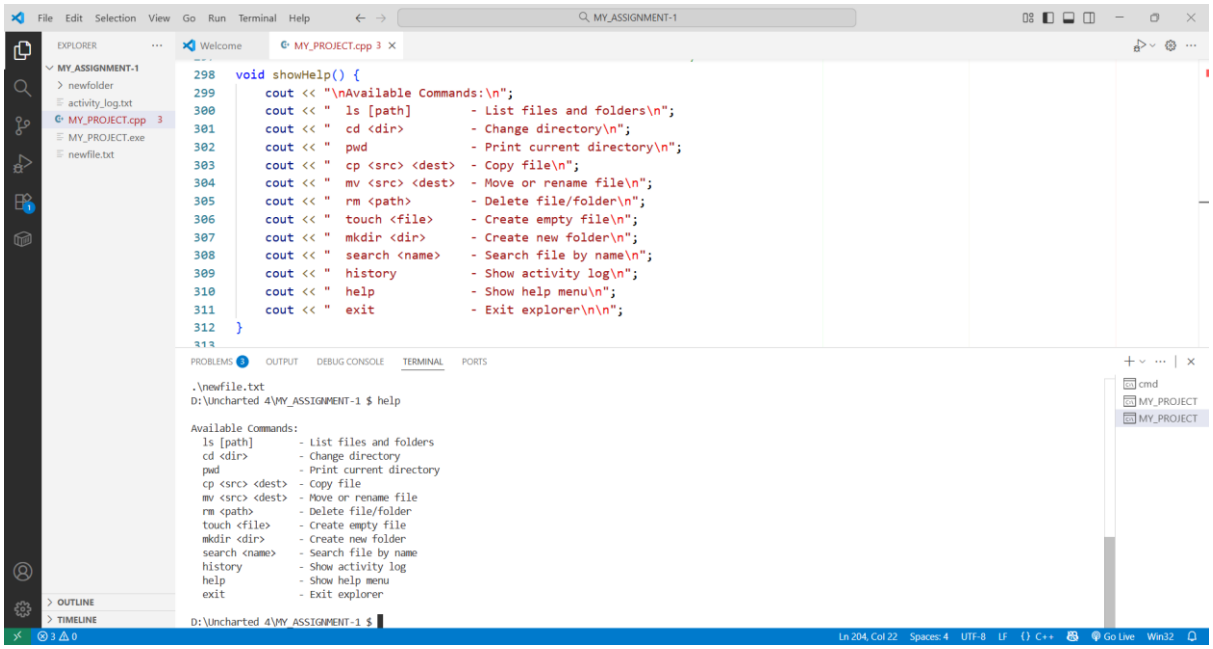

File Edit Selection View Go Run Terminal Help
MY_ASSIGNMENT-1
EXPLORER
MY_ASSIGNMENT-1
newfolder
activity_log.txt
MY_PROJECT.cpp 3
MY_PROJECT.exe
newfile.txt
Welcome
MY_PROJECT.cpp 3
267 /*--
269 --*/
270 void showHelp() {
271 cout << "\nAvailable Commands:\n";
272 cout << " ls [path] - List files and folders\n";
273 cout << " cd <dir> - Change directory\n";
274 cout << " pwd - Print current directory\n";
275 cout << " cp <src> <dest> - Copy file\n";
276 cout << " mv <src> <dest> - Move or rename file\n";
277 cout << " rm <path> - Delete file/folder\n";
278 cout << " touch <file> - Create empty file\n";
279 cout << " mkdir <dir> - Create new folder\n";
280 cout << " search <name> - Search file by name\n";
281 cout << " history - Show activity log\n";
282 cout << " help - Show help menu\n";
283 cout << " exit - Exit explorer\n\n";
284 }
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Type 'help' to see available commands.
D:\Uncharted 4\MY_ASSIGNMENT-1 $ touch newfile.txt
File created/updated: newfile.txt
D:\Uncharted 4\MY_ASSIGNMENT-1 $ mkdir newfolder
Directory created: newfolder
D:\Uncharted 4\MY_ASSIGNMENT-1 $ ls
Contents of .:
activity_log.txt (112 bytes)
MY_PROJECT.cpp (11254 bytes)
MY_PROJECT.exe (103280 bytes)
newfile.txt (0 bytes)
[DIR] newfolder (0 bytes)
D:\Uncharted 4\MY_ASSIGNMENT-1 $ cd newfolder
Changed directory to: newfolder
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $
cmd
MY_PROJECT
OUTLINE
TIMELINE
Ln 5, Col 36 Spaces: 4 UTF-8 LF C++ Go Live Win32

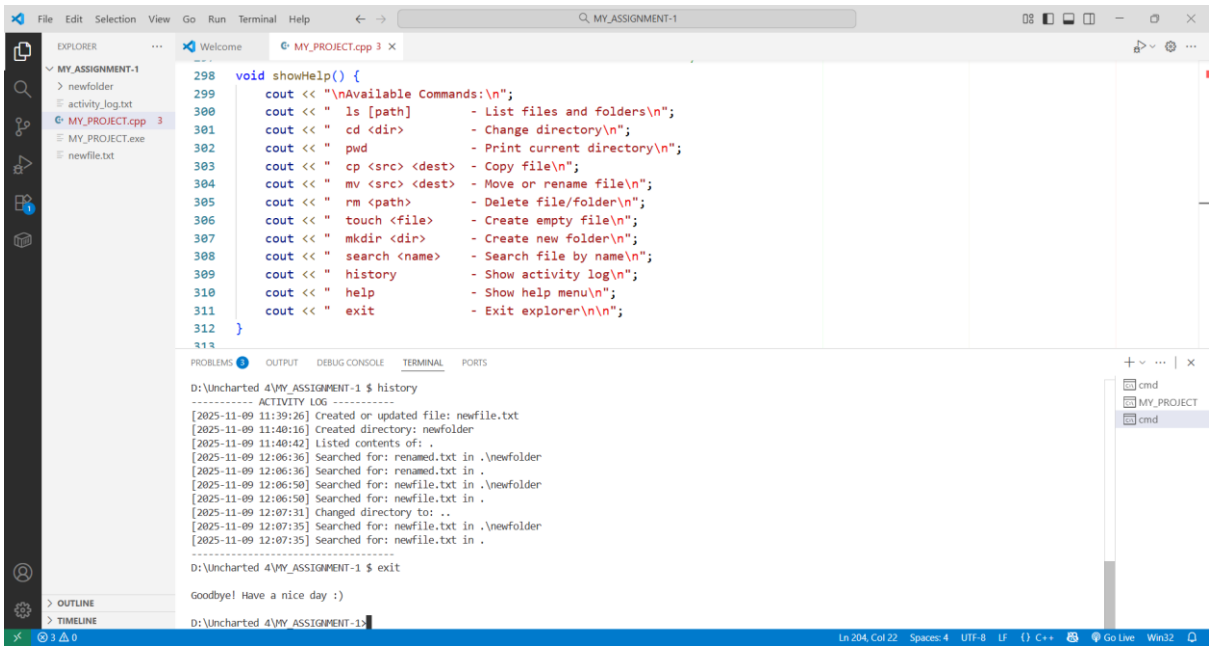

File Edit Selection View Go Run Terminal Help
MY_ASSIGNMENT-1
EXPLORER
MY_ASSIGNMENT-1
newfolder
activity_log.txt
MY_PROJECT.cpp 3
MY_PROJECT.exe
newfile.txt
Welcome
MY_PROJECT.cpp 3
267 /*--
269 --*/
270 void showHelp() {
271 cout << "\nAvailable Commands:\n";
272 cout << " ls [path] - List files and folders\n";
273 cout << " cd <dir> - Change directory\n";
274 cout << " pwd - Print current directory\n";
275 cout << " cp <src> <dest> - Copy file\n";
276 cout << " mv <src> <dest> - Move or rename file\n";
277 cout << " rm <path> - Delete file/folder\n";
278 cout << " touch <file> - Create empty file\n";
279 cout << " mkdir <dir> - Create new folder\n";
280 cout << " search <name> - Search file by name\n";
281 cout << " history - Show activity log\n";
282 cout << " help - Show help menu\n";
283 cout << " exit - Exit explorer\n\n";
284 }
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Contents of .:
activity_log.txt (112 bytes)
MY_PROJECT.cpp (11254 bytes)
MY_PROJECT.exe (103280 bytes)
newfile.txt (0 bytes)
[DIR] newfolder (0 bytes)
D:\Uncharted 4\MY_ASSIGNMENT-1 $ cd newfolder
Changed directory to: newfolder
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ cp newfile.txt newfile2.txt
cp: No such file or directory
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ cp newfile.txt newfile2.txt
cp: No such file or directory
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ touch newfile.txt
File created/updated: newfile.txt
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ cp newfile.txt newfile2.txt
Copied: newfile.txt -> newfile2.txt
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $
cmd
MY_PROJECT
OUTLINE
TIMELINE
Ln 5, Col 36 Spaces: 4 UTF-8 LF C++ Go Live Win32


File Edit Selection View Go Run Terminal Help
MY_ASSIGNMENT-1
EXPLORER
MY_ASSIGNMENT-1
newfolder
activity_log.txt
MY_PROJECT.cpp 3
MY_PROJECT.exe
newfile.txt
Welcome
MY_PROJECT.cpp 3
267 /*--
269 --*/
270 void showHelp() {
271 cout << "\nAvailable Commands:\n";
272 cout << " ls [path] - List files and folders\n";
273 cout << " cd <dir> - Change directory\n";
274 cout << " pwd - Print current directory\n";
275 cout << " cp <src> <dest> - Copy file\n";
276 cout << " mv <src> <dest> - Move or rename file\n";
277 cout << " rm <path> - Delete file/folder\n";
278 cout << " touch <file> - Create empty file\n";
279 cout << " mkdir <dir> - Create new folder\n";
280 cout << " search <name> - Search file by name\n";
281 cout << " history - Show activity log\n";
282 cout << " help - Show help menu\n";
283 cout << " exit - Exit explorer\n\n";
284 }
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ cp newfile.txt newfile2.txt
cp: No such file or directory
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ cp newfile.txt newfile2.txt
cp: No such file or directory
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ touch newfile.txt
File created/updated: newfile.txt
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ cp newfile.txt newfile2.txt
Copied: newfile.txt -> newfile2.txt
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ mv newfile.txt newfolder/newfile2.txt
mv: No such file or directory
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ cd newfolder
cd: No such file or directory
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ mv newfile.txt newfile2.txt
mv: File exists
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ mv newfile.txt renamed.txt
Moved: newfile.txt -> renamed.txt
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $
cmd
MY_PROJECT
OUTLINE
TIMELINE
Ln 5, Col 36 Spaces: 4 UTF-8 LF C++ Go Live Win32


File Edit Selection View Go Run Terminal Help
MY_ASSIGNMENT-1
EXPLORER
MY_ASSIGNMENT-1
newfolder
activity_log.txt
MY_PROJECT.cpp 3
MY_PROJECT.exe
newfile.txt
Welcome
MY_PROJECT.cpp 3
267 /*--
269 --*/
270 void showHelp() {
271 cout << "\nAvailable Commands:\n";
272 cout << " ls [path] - List files and folders\n";
273 cout << " cd <dir> - Change directory\n";
274 cout << " pwd - Print current directory\n";
275 cout << " cp <src> <dest> - Copy file\n";
276 cout << " mv <src> <dest> - Move or rename file\n";
277 cout << " rm <path> - Delete file/folder\n";
278 cout << " touch <file> - Create empty file\n";
279 cout << " mkdir <dir> - Create new folder\n";
280 cout << " search <name> - Search file by name\n";
281 cout << " history - Show activity log\n";
282 cout << " help - Show help menu\n";
283 cout << " exit - Exit explorer\n\n";
284 }
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
e.txt
File created/updated: newfile.txt
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ cp newfile.txt newfile2.txt
Copied: newfile.txt -> newfile2.txt
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ mv newfile.txt newfolder/newfile2.txt
mv: No such file or directory
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ cd newfolder
cd: No such file or directory
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ mv newfile.txt newfile2.txt
mv: File exists
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ mv newfile.txt renamed.txt
Moved: newfile.txt -> renamed.txt
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $ rm renamed.txt
D:\Uncharted 4\MY_ASSIGNMENT-1\newfolder $
cmd
MY_PROJECT
OUTLINE
TIMELINE
Ln 5, Col 36 Spaces: 4 UTF-8 LF C++ Go Live Win32


File Edit Selection View Go Run Terminal Help
MY_ASSIGNMENT-1
EXPLORER
MY_ASSIGNMENT-1
newfolder
activity_log.txt
MY_PROJECT.cpp 3
MY_PROJECT.exe
newfile.txt
Welcome
MY_PROJECT.cpp 3
298 void showHelp() {
299 cout << "\nAvailable Commands:\n";
300 cout << " ls [path] - List files and folders\n";
301 cout << " cd <dir> - Change directory\n";
302 cout << " pwd - Print current directory\n";
303 cout << " cp <src> <dest> - Copy file\n";
304 cout << " mv <src> <dest> - Move or rename file\n";
305 cout << " rm <path> - Delete file/folder\n";
306 cout << " touch <file> - Create empty file\n";
307 cout << " mkdir <dir> - Create new folder\n";
308 cout << " search <name> - Search file by name\n";
309 cout << " history - Show activity log\n";
310 cout << " help - Show help menu\n";
311 cout << " exit - Exit explorer\n\n";
312 }
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
D:\Uncharted 4\MY_ASSIGNMENT-1>MY_PROJECT.exe
SIMPLE CONSOLE FILE EXPLORER (C++ / GCC6)
Type 'help' to see available commands.
D:\Uncharted 4\MY_ASSIGNMENT-1 $ search renamed.txt
D:\Uncharted 4\MY_ASSIGNMENT-1 $ search newfile.txt
.\newfile.txt
Type 'help' to see available commands.
Type 'help' to see available commands.
Type 'help' to see available commands.
D:\Uncharted 4\MY_ASSIGNMENT-1 $ search renamed.txt
D:\Uncharted 4\MY_ASSIGNMENT-1 $ search newfile.txt
.\newfile.txt
D:\Uncharted 4\MY_ASSIGNMENT-1 $ cd newfolder
cmd
MY_PROJECT
cmd
OUTLINE
TIMELINE
Ln 310, Col 53 Spaces: 4 UTF-8 LF C++ Go Live Win32


File Edit Selection View Go Run Terminal Help
MY_ASSIGNMENT-1
EXPLORER
MY_ASSIGNMENT-1
newfolder
activity_log.txt
MY_PROJECT.cpp 3
MY_PROJECT.exe
newfile.txt
Welcome
MY_PROJECT.cpp 3
298 void showHelp() {
299 cout << "\nAvailable Commands:\n";
300 cout << " ls [path] - List files and folders\n";
301 cout << " cd <dir> - Change directory\n";
302 cout << " pwd - Print current directory\n";
303 cout << " cp <src> <dest> - Copy file\n";
304 cout << " mv <src> <dest> - Move or rename file\n";
305 cout << " rm <path> - Delete file/folder\n";
306 cout << " touch <file> - Create empty file\n";
307 cout << " mkdir <dir> - Create new folder\n";
308 cout << " search <name> - Search file by name\n";
309 cout << " history - Show activity log\n";
310 cout << " help - Show help menu\n";
311 cout << " exit - Exit explorer\n\n";
312 }
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
.\newfile.txt
D:\Uncharted 4\MY_ASSIGNMENT-1 $ help
Available Commands:
ls [path] - List files and folders
cd <dir> - Change directory
pwd - Print current directory
cp <src> <dest> - Copy file
mv <src> <dest> - Move or rename file
rm <path> - Delete file/folder
touch <file> - Create empty file
mkdir <dir> - Create new folder
search <name> - Search file by name
history - Show activity log
help - Show help menu
exit - Exit explorer
D:\Uncharted 4\MY_ASSIGNMENT-1 $
cmd
MY_PROJECT
MY_PROJECT
OUTLINE
TIMELINE
Ln 204, Col 22 Spaces: 4 UTF-8 LF C++ Go Live Win32


File Edit Selection View Go Run Terminal Help
MY_ASSIGNMENT-1
EXPLORER
MY_ASSIGNMENT-1
newfolder
activity_log.txt
MY_PROJECT.cpp 3
MY_PROJECT.exe
newfile.txt
Welcome
MY_PROJECT.cpp 3
298 void showHelp() {
299 cout << "\nAvailable Commands:\n";
300 cout << " ls [path] - List files and folders\n";
301 cout << " cd <dir> - Change directory\n";
302 cout << " pwd - Print current directory\n";
303 cout << " cp <src> <dest> - Copy file\n";
304 cout << " mv <src> <dest> - Move or rename file\n";
305 cout << " rm <path> - Delete file/folder\n";
306 cout << " touch <file> - Create empty file\n";
307 cout << " mkdir <dir> - Create new folder\n";
308 cout << " search <name> - Search file by name\n";
309 cout << " history - Show activity log\n";
310 cout << " help - Show help menu\n";
311 cout << " exit - Exit explorer\n\n";
312 }
313
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
D:\Uncharted 4\MY_ASSIGNMENT-1 $ history
---------- ACTIVITY LOG ----------
[2025-11-09 11:39:26] Created or updated file: newfile.txt
[2025-11-09 11:40:16] Created directory: newfolder
[2025-11-09 11:40:42] Listed contents of: .
[2025-11-09 12:06:36] Searched for: renamed.txt in .\newfolder
[2025-11-09 12:06:36] Searched for: renamed.txt in .
[2025-11-09 12:06:50] Searched for: newfile.txt in .\newfolder
[2025-11-09 12:06:50] Searched for: newfile.txt in .
[2025-11-09 12:07:31] Changed directory to: ..
[2025-11-09 12:07:35] Searched for: newfile.txt in .\newfolder
[2025-11-09 12:07:35] Searched for: newfile.txt in .

D:\Uncharted 4\MY_ASSIGNMENT-1 $ exit
Goodbye! Have a nice day :)
D:\Uncharted 4\MY_ASSIGNMENT-1>
cmd
MY_PROJECT
cmd
OUTLINE
TIMELINE
Ln 204, Col 22 Spaces: 4 UTF-8 LF {} C++ Go Live Win32