FILE EXPLORER APPLICATION WITH ENHANCED FILE HISTORY & LOG TRACKING

Name: ALIPSA BARAL
Registration number: 2241002027
Batch: 4
Branch: CSE

```
/*
   ------------------------------------------------------------
   SIMPLE CONSOLE FILE EXPLORER (WITH ACTIVITY LOG)
   ------------------------------------------------------------
   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

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)
------------------------------------------------------------*/
void searchFile(const string &pattern, const string &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 fullPath = path + PATH_SEP + name;

        if (name.find(pattern) != string::npos)
            cout << fullPath << endl;

        if (isDirectory(fullPath))
            searchFile(pattern, fullPath);
    }
    closedir(dir);
    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
------------------------------------------------------------*/
```

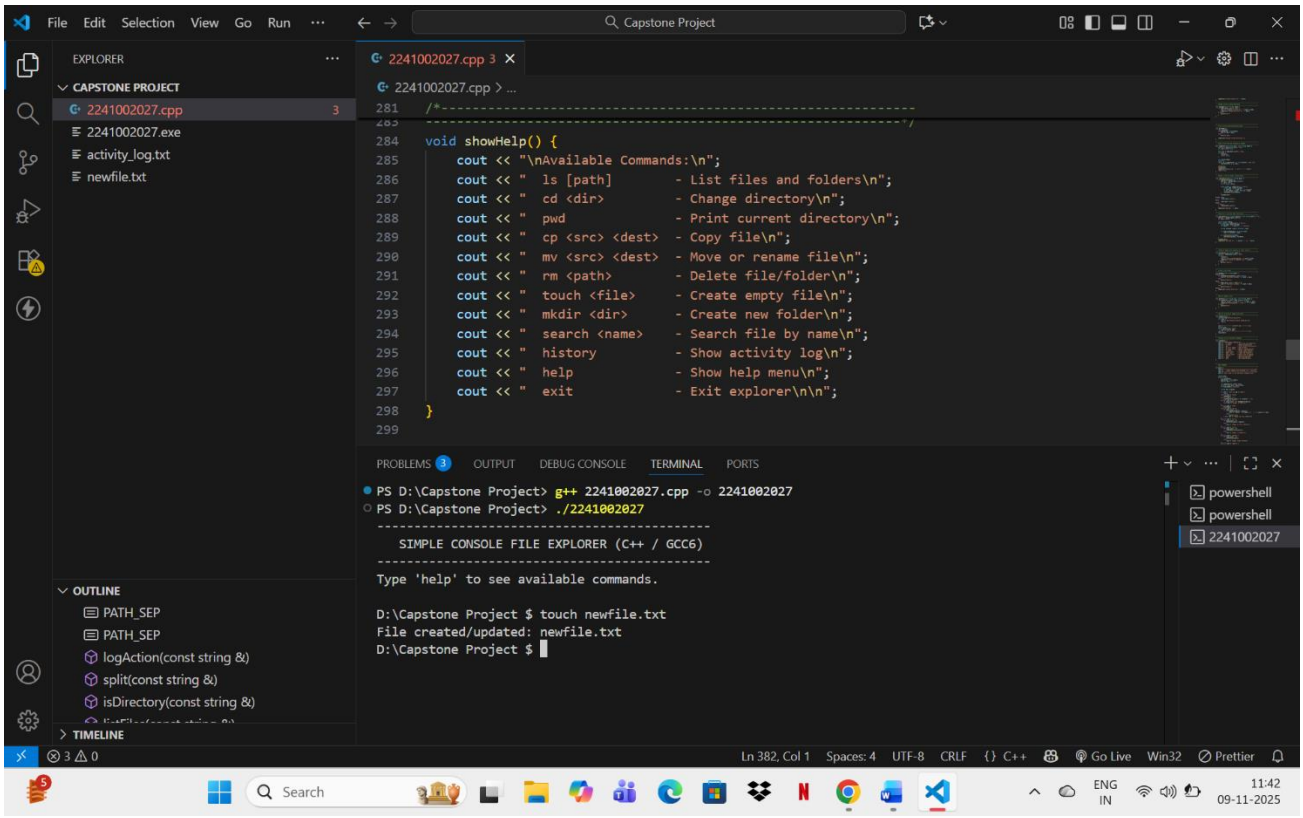
```
void showHelp() {
    cout << "\nAvailable Commands:\n";
    cout << "  ls [path]       - List files and folders\n";
    cout << "  cd <dir>        - Change directory\n";
    cout << "  pwd             - Print current directory\n";
    cout << "  cp <src> <dest>  - Copy file\n";
    cout << "  mv <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 << " $ ";
```

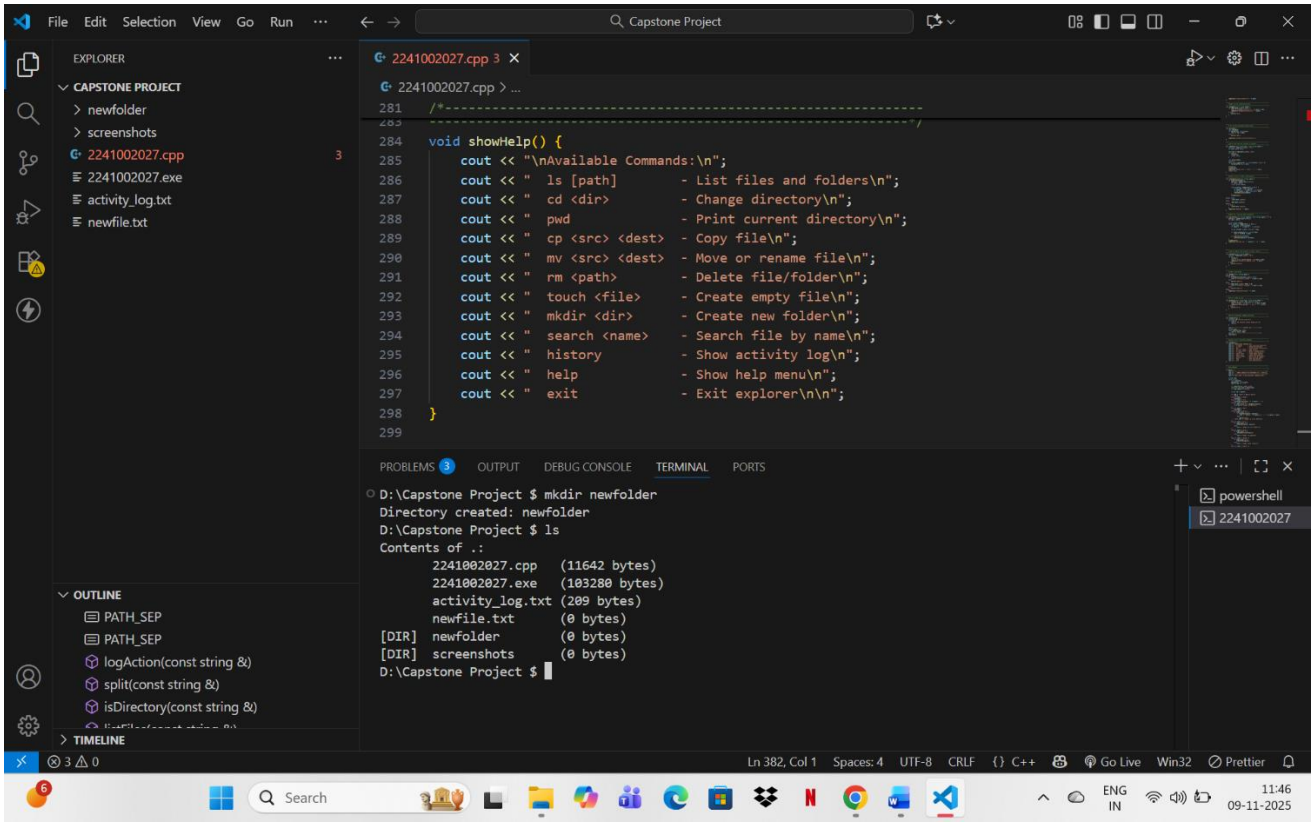
```
if (!getline(cin, line)) break;
vector<string> args = split(line);
if (args.empty()) continue;

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";
```

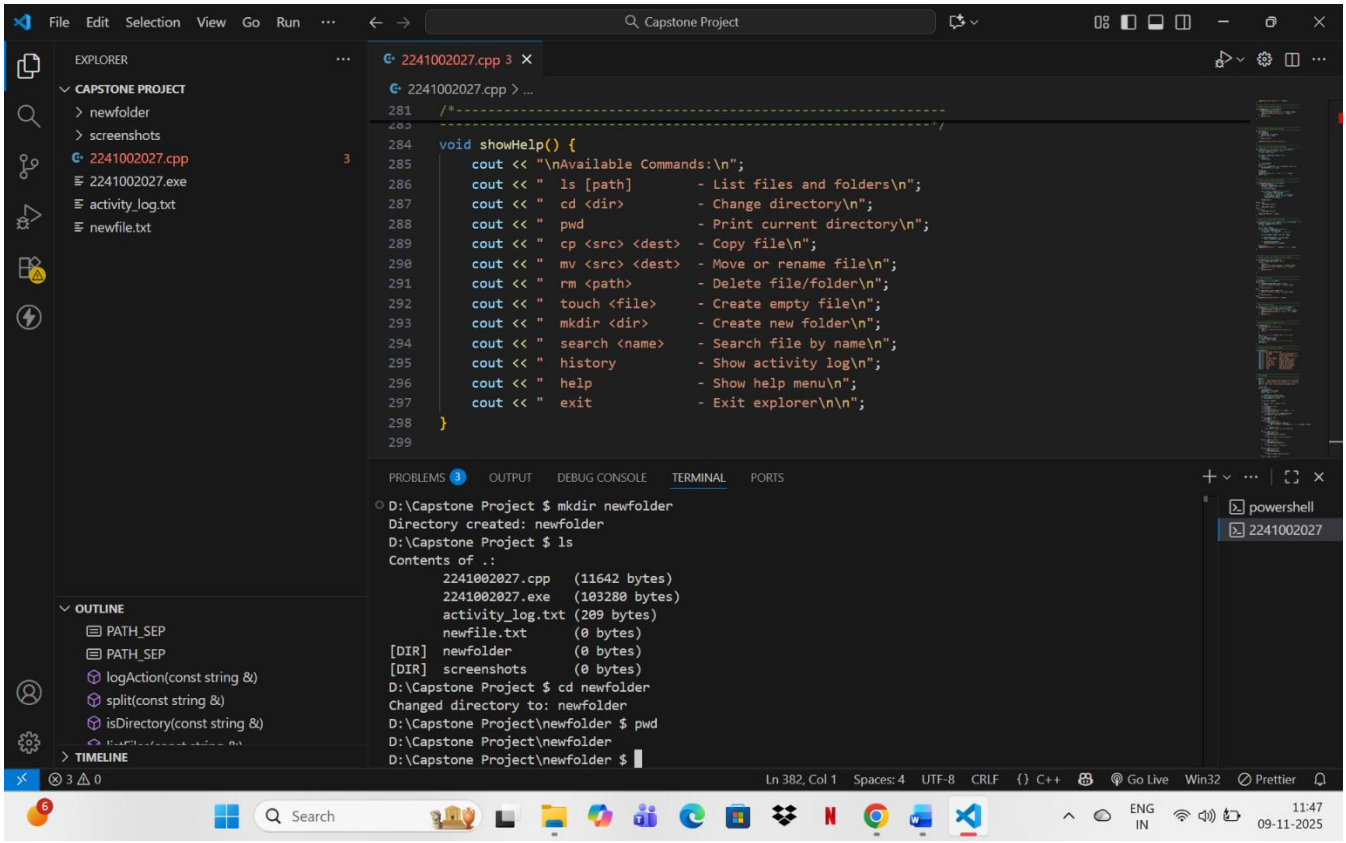
```
    }
    else if (cmd == "rm") {
        if (args.size() > 1)
            removeRecursive(args[1]);
        else
            cout << "Usage: rm <path>\n";
    }
    else if (cmd == "touch") {
        if (args.size() > 1)
            touchFile(args[1]);
        else
            cout << "Usage: touch <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";
}
```

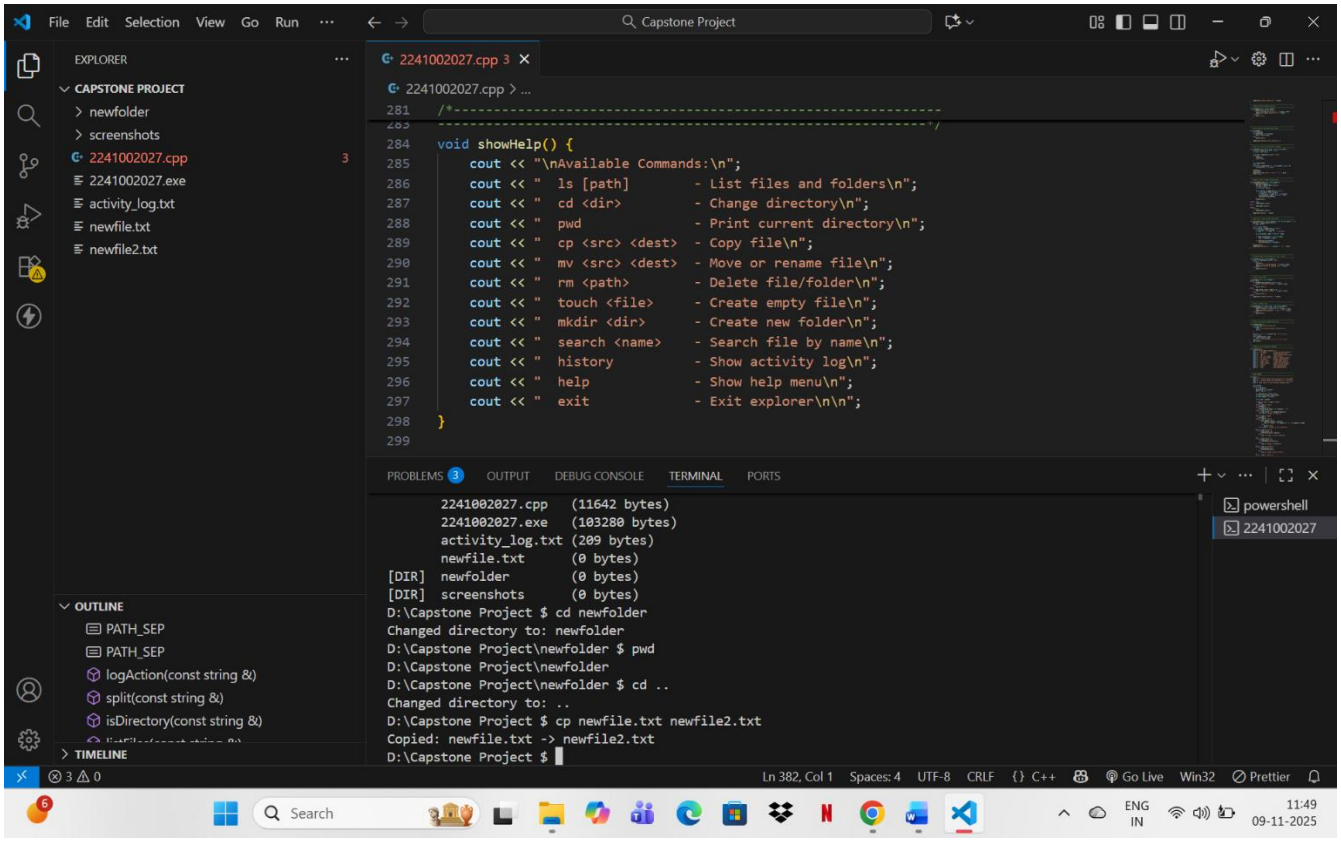
```
    cout << "\nGoodbye! Have a nice day :)\n";
    return 0;
}
```

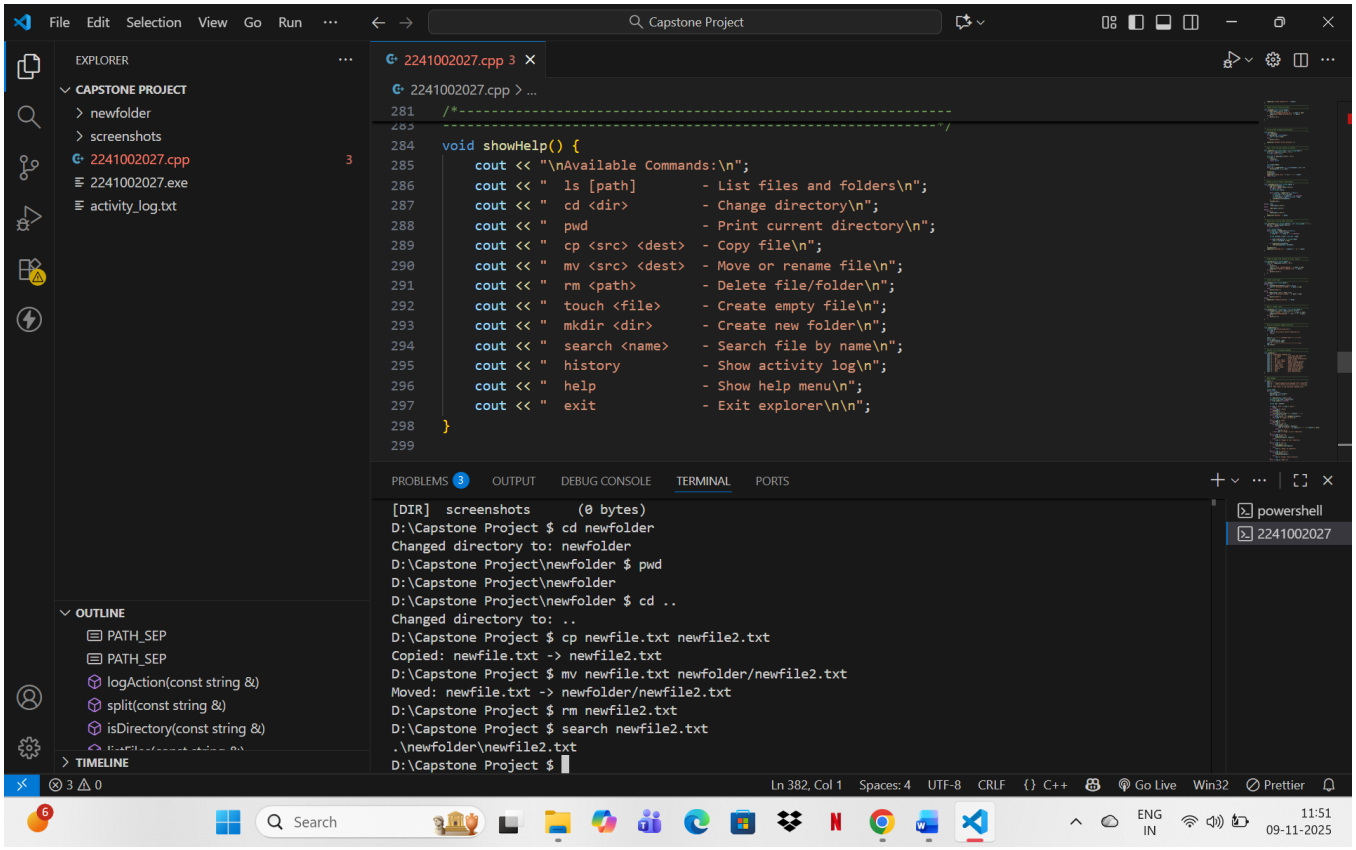
SCREENSHOTS

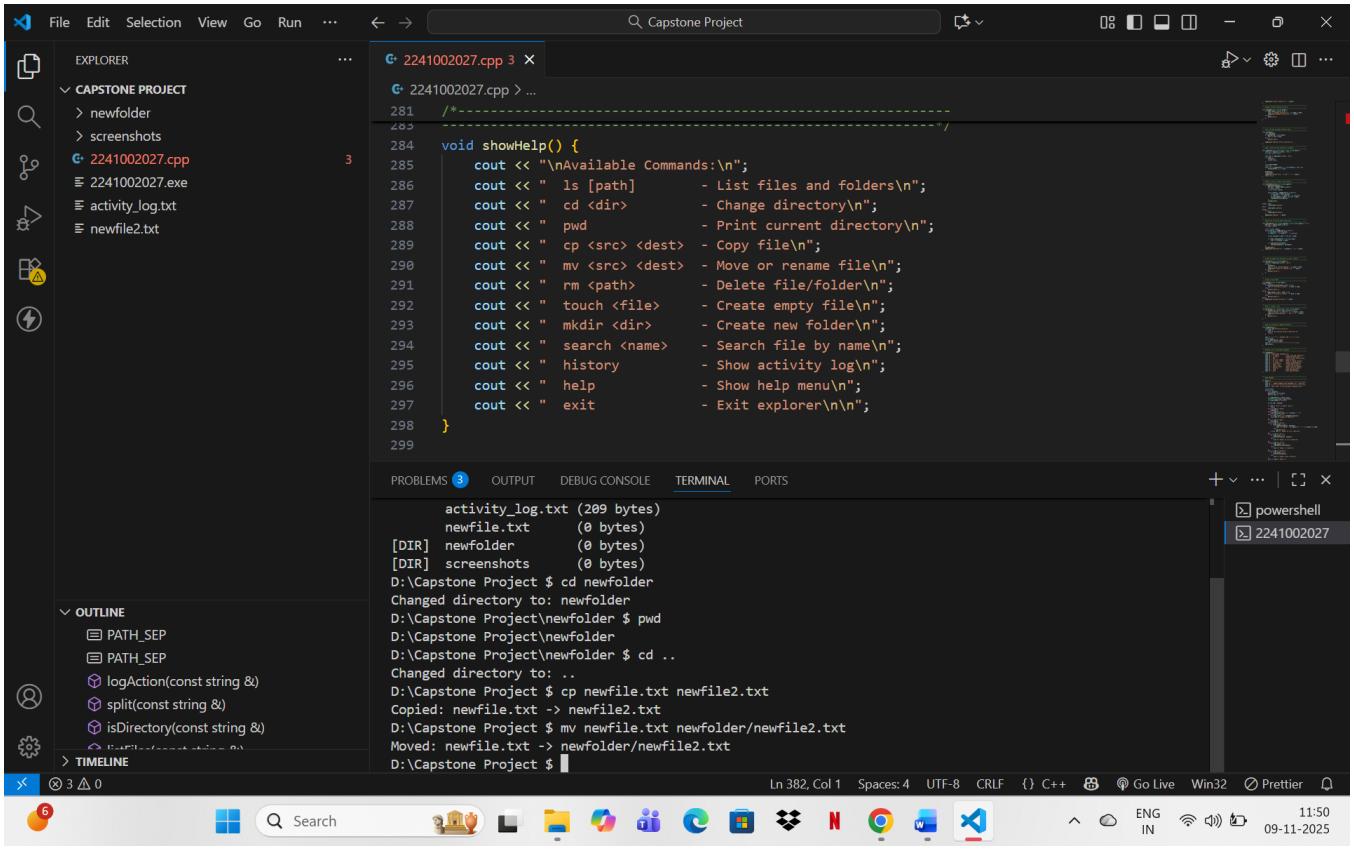


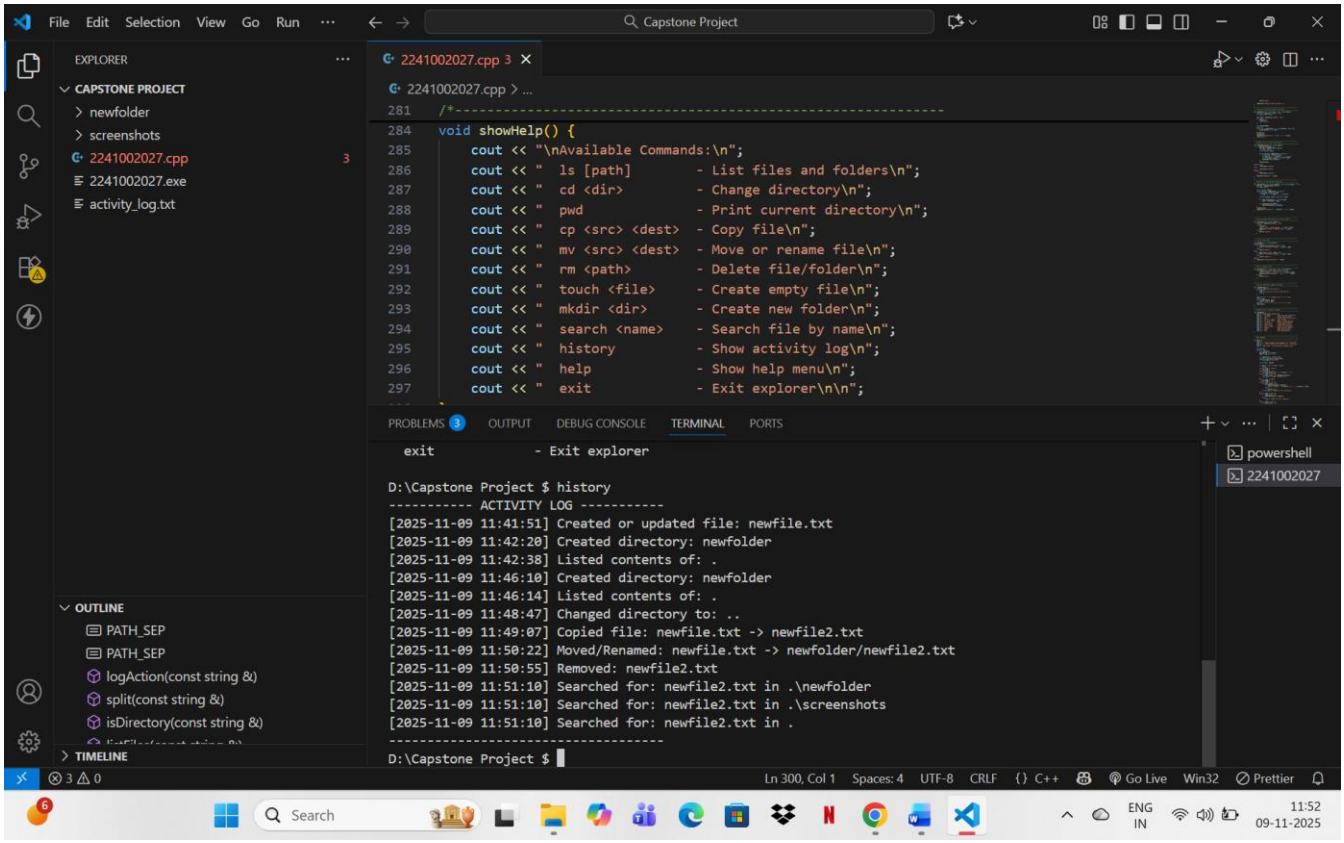


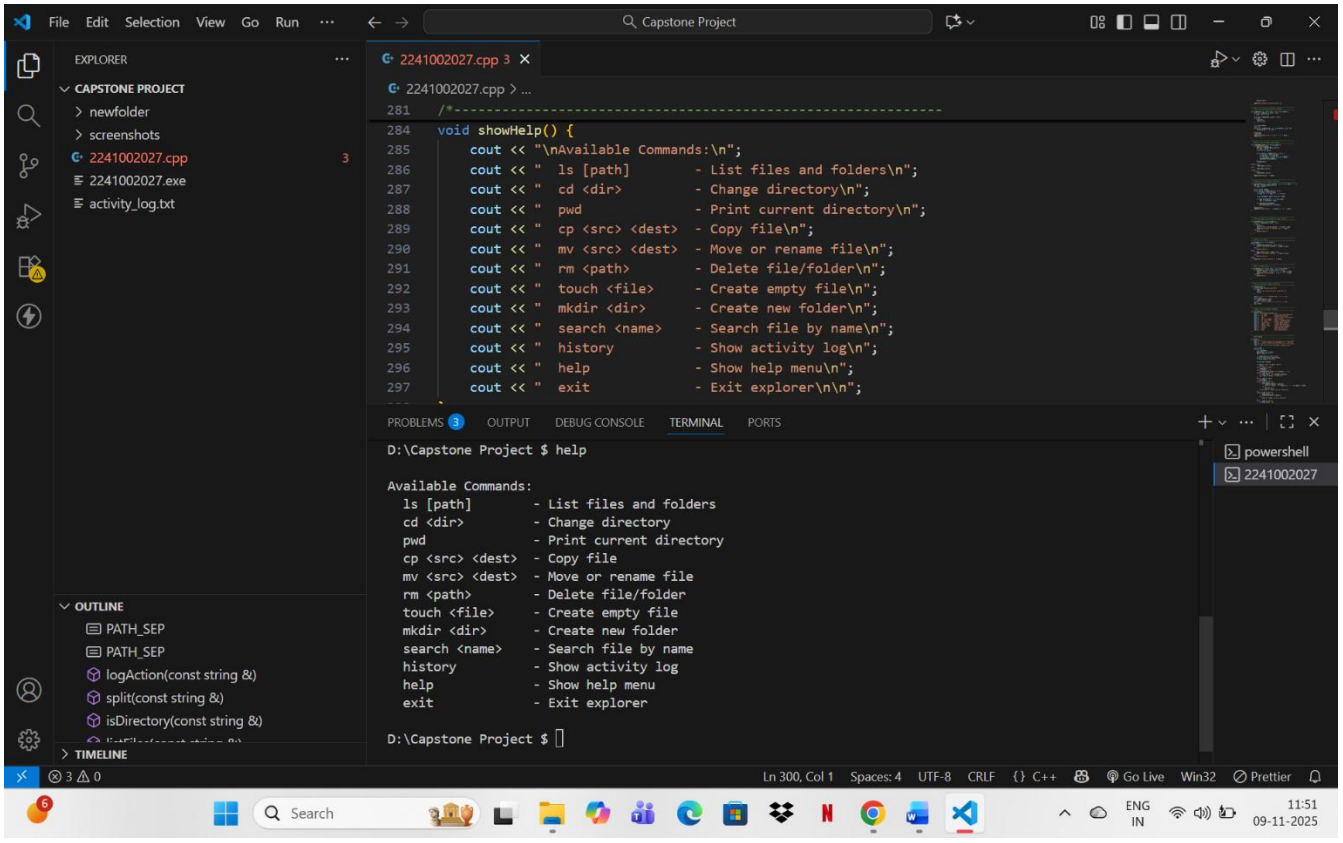

File Edit Selection View Go Run
Capstone Project
EXPLORER
CAPSTONE PROJECT
newfolder
screenshots
2241002027.cpp 3
2241002027.exe
activity_log.txt
newfile.txt
2241002027.cpp 3
2241002027.cpp > ...
281 /*
284 void showHelp() {
285 cout << "\nAvailable Commands:\n";
286 cout << " ls [path] - List files and folders\n";
287 cout << " cd <dir> - Change directory\n";
288 cout << " pwd - Print current directory\n";
289 cout << " cp <src> <dest> - Copy file\n";
290 cout << " mv <src> <dest> - Move or rename file\n";
291 cout << " rm <path> - Delete file/folder\n";
292 cout << " touch <file> - Create empty file\n";
293 cout << " mkdir <dir> - Create new folder\n";
294 cout << " search <name> - Search file by name\n";
295 cout << " history - Show activity log\n";
296 cout << " help - Show help menu\n";
297 cout << " exit - Exit explorer\n\n";
298 }
299
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
D:\Capstone Project $ mkdir newfolder
Directory created: newfolder
D:\Capstone Project $ ls
Contents of .:
2241002027.cpp (11642 bytes)
2241002027.exe (103280 bytes)
activity_log.txt (209 bytes)
newfile.txt (0 bytes)
[DIR] newfolder (0 bytes)
[DIR] screenshots (0 bytes)
D:\Capstone Project $ cd newfolder
Changed directory to: newfolder
D:\Capstone Project\newfolder $ pwd
D:\Capstone Project\newfolder
D:\Capstone Project\newfolder $
powershell
2241002027
OUTLINE
PATH_SEP
PATH_SEP
logAction(const string &)
split(const string &)
isDirectory(const string &)
TIMELINE
Ln 382, Col 1 Spaces: 4 UTF-8 CRLF {} C++ Go Live Win32 Prettier
Search
ENG IN
11:47
09-11-2025

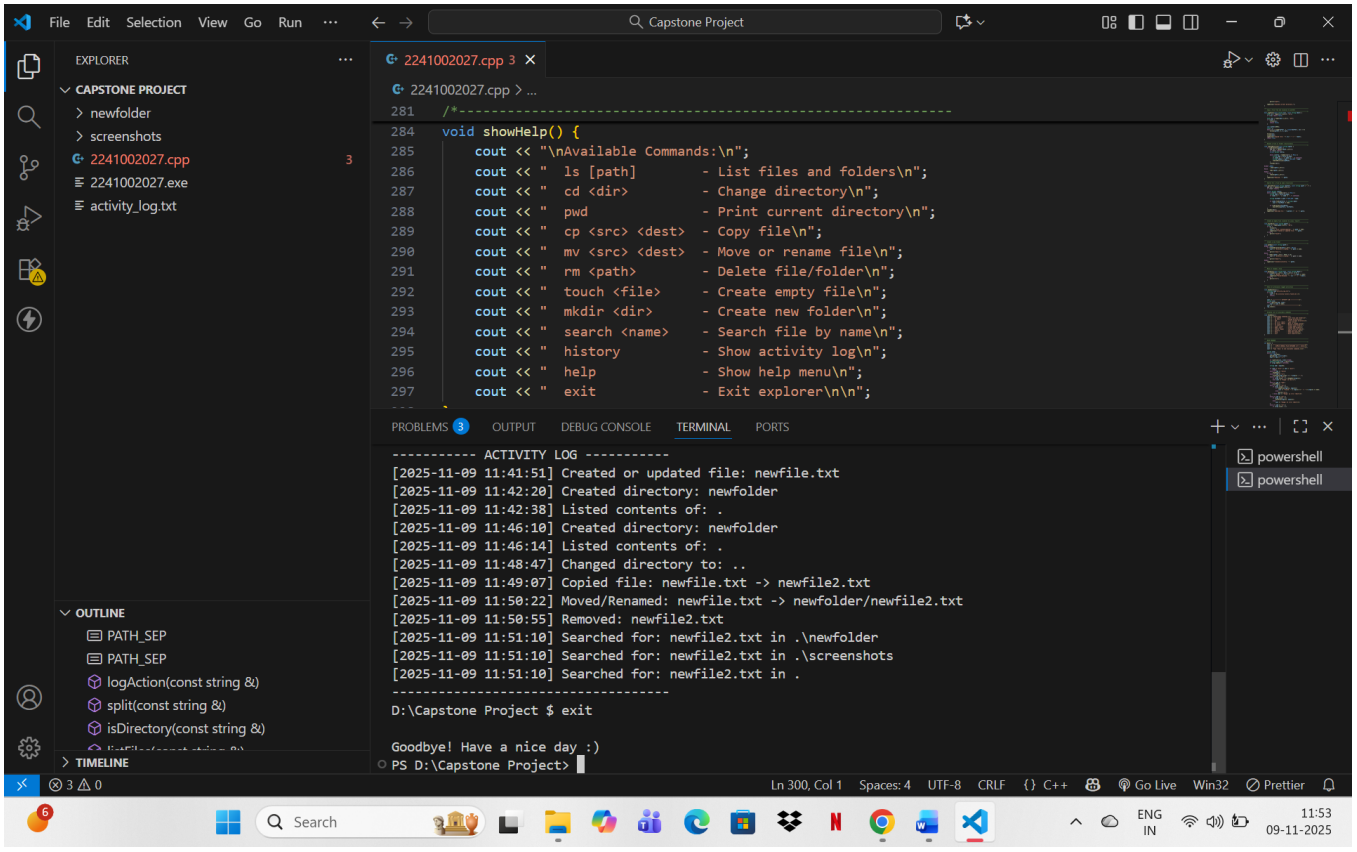

File Edit Selection View Go Run
Capstone Project
EXPLORER
CAPSTONE PROJECT
newfolder
screenshots
2241002027.cpp 3
2241002027.exe
activity_log.txt
newfile.txt
newfile2.txt
2241002027.cpp > ...
281 /*
284 void showHelp() {
285 cout << "\nAvailable Commands:\n";
286 cout << " ls [path] - List files and folders\n";
287 cout << " cd <dir> - Change directory\n";
288 cout << " pwd - Print current directory\n";
289 cout << " cp <src> <dest> - Copy file\n";
290 cout << " mv <src> <dest> - Move or rename file\n";
291 cout << " rm <path> - Delete file/folder\n";
292 cout << " touch <file> - Create empty file\n";
293 cout << " mkdir <dir> - Create new folder\n";
294 cout << " search <name> - Search file by name\n";
295 cout << " history - Show activity log\n";
296 cout << " help - Show help menu\n";
297 cout << " exit - Exit explorer\n\n";
298 }
299
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
2241002027.cpp (11642 bytes)
2241002027.exe (103280 bytes)
activity_log.txt (209 bytes)
newfile.txt (0 bytes)
[DIR] newfolder (0 bytes)
[DIR] screenshots (0 bytes)
D:\Capstone Project $ cd newfolder
Changed directory to: newfolder
D:\Capstone Project\newfolder $ pwd
D:\Capstone Project\newfolder
D:\Capstone Project\newfolder $ cd ..
Changed directory to: ..
D:\Capstone Project $ cp newfile.txt newfile2.txt
Copied: newfile.txt -> newfile2.txt
D:\Capstone Project $
powershell
2241002027
OUTLINE
PATH_SEP
PATH_SEP
logAction(const string &)
split(const string &)
isDirectory(const string &)
TIMELINE
Ln 382, Col 1 Spaces: 4 UTF-8 CRLF {} C++ Go Live Win32 Prettier
Search
ENG IN
11:49
09-11-2025


File Edit Selection View Go Run
Capstone Project
EXPLORER
CAPSTONE PROJECT
newfolder
screenshots
2241002027.cpp 3
2241002027.exe
activity_log.txt
2241002027.cpp 3
2241002027.cpp > ...
281 /*----------
284 void showHelp() {
285 cout << "\nAvailable Commands:\n";
286 cout << " ls [path] - List files and folders\n";
287 cout << " cd <dir> - Change directory\n";
288 cout << " pwd - Print current directory\n";
289 cout << " cp <src> <dest> - Copy file\n";
290 cout << " mv <src> <dest> - Move or rename file\n";
291 cout << " rm <path> - Delete file/folder\n";
292 cout << " touch <file> - Create empty file\n";
293 cout << " mkdir <dir> - Create new folder\n";
294 cout << " search <name> - Search file by name\n";
295 cout << " history - Show activity log\n";
296 cout << " help - Show help menu\n";
297 cout << " exit - Exit explorer\n\n";
298 }
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
[DIR] screenshots (0 bytes)
D:\Capstone Project $ cd newfolder
Changed directory to: newfolder
D:\Capstone Project\newfolder $ pwd
D:\Capstone Project\newfolder
D:\Capstone Project\newfolder $ cd ..
Changed directory to: ..
D:\Capstone Project $ cp newfile.txt newfile2.txt
Copied: newfile.txt -> newfile2.txt
D:\Capstone Project $ mv newfile.txt newfolder/newfile2.txt
Moved: newfile.txt -> newfolder/newfile2.txt
D:\Capstone Project $ rm newfile2.txt
D:\Capstone Project $ search newfile2.txt
.\newfolder\newfile2.txt
D:\Capstone Project $
powershell
2241002027
OUTLINE
PATH_SEP
PATH_SEP
logAction(const string &)
split(const string &)
isDirectory(const string &)
TIMELINE
Ln 382, Col 1 Spaces: 4 UTF-8 CRLF {} C++ Go Live Win32 Prettier
Search
ENG IN
11:51
09-11-2025


File Edit Selection View Go Run
Capstone Project
EXPLORER
CAPSTONE PROJECT
newfolder
screenshots
2241002027.cpp 3
2241002027.exe
activity_log.txt
newfile2.txt
2241002027.cpp 3
2241002027.cpp > ...
281 /*----------
284 void showHelp() {
285 cout << "\nAvailable Commands:\n";
286 cout << " ls [path] - List files and folders\n";
287 cout << " cd <dir> - Change directory\n";
288 cout << " pwd - Print current directory\n";
289 cout << " cp <src> <dest> - Copy file\n";
290 cout << " mv <src> <dest> - Move or rename file\n";
291 cout << " rm <path> - Delete file/folder\n";
292 cout << " touch <file> - Create empty file\n";
293 cout << " mkdir <dir> - Create new folder\n";
294 cout << " search <name> - Search file by name\n";
295 cout << " history - Show activity log\n";
296 cout << " help - Show help menu\n";
297 cout << " exit - Exit explorer\n\n";
298 }
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
activity_log.txt (209 bytes)
newfile.txt (0 bytes)
[DIR] newfolder (0 bytes)
[DIR] screenshots (0 bytes)
D:\Capstone Project $ cd newfolder
Changed directory to: newfolder
D:\Capstone Project\newfolder $ pwd
D:\Capstone Project\newfolder
D:\Capstone Project\newfolder $ cd ..
Changed directory to: ..
D:\Capstone Project $ cp newfile.txt newfile2.txt
Copied: newfile.txt -> newfile2.txt
D:\Capstone Project $ mv newfile.txt newfolder/newfile2.txt
Moved: newfile.txt -> newfolder/newfile2.txt
D:\Capstone Project $
powershell
2241002027
OUTLINE
PATH_SEP
PATH_SEP
logAction(const string &)
split(const string &)
isDirectory(const string &)
TIMELINE
Ln 382, Col 1 Spaces: 4 UTF-8 CRLF {} C++ Go Live Win32 Prettier
Search
ENG IN
11:50
09-11-2025


File Edit Selection View Go Run
Capstone Project
EXPLORER
CAPSTONE PROJECT
newfolder
screenshots
2241002027.cpp
2241002027.exe
activity_log.txt
2241002027.cpp 3
2241002027.cpp > ...
281 /*----------
284 void showHelp() {
285 cout << "\nAvailable Commands:\n";
286 cout << " ls [path] - List files and folders\n";
287 cout << " cd <dir> - Change directory\n";
288 cout << " pwd - Print current directory\n";
289 cout << " cp <src> <dest> - Copy file\n";
290 cout << " mv <src> <dest> - Move or rename file\n";
291 cout << " rm <path> - Delete file/folder\n";
292 cout << " touch <file> - Create empty file\n";
293 cout << " mkdir <dir> - Create new folder\n";
294 cout << " search <name> - Search file by name\n";
295 cout << " history - Show activity log\n";
296 cout << " help - Show help menu\n";
297 cout << " exit - Exit explorer\n\n";
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
exit - Exit explorer
D:\Capstone Project $ history
---------- ACTIVITY LOG ----------
[2025-11-09 11:41:51] Created or updated file: newfile.txt
[2025-11-09 11:42:20] Created directory: newfolder
[2025-11-09 11:42:38] Listed contents of: .
[2025-11-09 11:46:10] Created directory: newfolder
[2025-11-09 11:46:14] Listed contents of: .
[2025-11-09 11:48:47] Changed directory to: ..
[2025-11-09 11:49:07] Copied file: newfile.txt -> newfile2.txt
[2025-11-09 11:50:22] Moved/Renamed: newfile.txt -> newfolder/newfile2.txt
[2025-11-09 11:50:55] Removed: newfile2.txt
[2025-11-09 11:51:10] Searched for: newfile2.txt in .\newfolder
[2025-11-09 11:51:10] Searched for: newfile2.txt in .\screenshots
[2025-11-09 11:51:10] Searched for: newfile2.txt in .
D:\Capstone Project $
powershell
2241002027
OUTLINE
PATH_SEP
PATH_SEP
logAction(const string &)
split(const string &)
isDirectory(const string &)
TIMELINE
Ln 300, Col 1 Spaces: 4 UTF-8 CRLF {} C++ Go Live Win32 Prettier
Search
ENG IN
11:52
09-11-2025


File Edit Selection View Go Run
Capstone Project
EXPLORER
CAPSTONE PROJECT
newfolder
screenshots
2241002027.cpp
2241002027.exe
activity_log.txt
2241002027.cpp 3
2241002027.cpp > ...
281 /*----------
284 void showHelp() {
285 cout << "\nAvailable Commands:\n";
286 cout << " ls [path] - List files and folders\n";
287 cout << " cd <dir> - Change directory\n";
288 cout << " pwd - Print current directory\n";
289 cout << " cp <src> <dest> - Copy file\n";
290 cout << " mv <src> <dest> - Move or rename file\n";
291 cout << " rm <path> - Delete file/folder\n";
292 cout << " touch <file> - Create empty file\n";
293 cout << " mkdir <dir> - Create new folder\n";
294 cout << " search <name> - Search file by name\n";
295 cout << " history - Show activity log\n";
296 cout << " help - Show help menu\n";
297 cout << " exit - Exit explorer\n\n";
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS
D:\Capstone Project $ 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:\Capstone Project $
powershell
2241002027
OUTLINE
PATH_SEP
PATH_SEP
logAction(const string &)
split(const string &)
isDirectory(const string &)
TIMELINE
Ln 300, Col 1 Spaces: 4 UTF-8 CRLF {} C++ Go Live Win32 Prettier
Search
ENG IN
11:51
09-11-2025


File Edit Selection View Go Run
Capstone Project
EXPLORER
CAPSTONE PROJECT
newfolder
screenshots
2241002027.cpp 3
2241002027.exe
activity_log.txt
2241002027.cpp 3
2241002027.cpp > ...
/*--
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
powershell
powershell
---------- ACTIVITY LOG ----------
[2025-11-09 11:41:51] Created or updated file: newfile.txt
[2025-11-09 11:42:20] Created directory: newfolder
[2025-11-09 11:42:38] Listed contents of: .
[2025-11-09 11:46:10] Created directory: newfolder
[2025-11-09 11:46:14] Listed contents of: .
[2025-11-09 11:48:47] Changed directory to: ..
[2025-11-09 11:49:07] Copied file: newfile.txt -> newfile2.txt
[2025-11-09 11:50:22] Moved/Renamed: newfile.txt -> newfolder/newfile2.txt
[2025-11-09 11:50:55] Removed: newfile2.txt
[2025-11-09 11:51:10] Searched for: newfile2.txt in .\newfolder
[2025-11-09 11:51:10] Searched for: newfile2.txt in .\screenshots
[2025-11-09 11:51:10] Searched for: newfile2.txt in .
D:\Capstone Project $ exit
Goodbye! Have a nice day :)
PS D:\Capstone Project>
OUTLINE
PATH_SEP
PATH_SEP
logAction(const string &)
split(const string &)
isDirectory(const string &)
TIMELINE
Ln 300, Col 1
Spaces: 4
UTF-8
CRLF
C++
Go Live
Win32
Prettier
Search
ENG
IN
11:53
09-11-2025