File Explorer Application

Objective: Develop a console-based file explorer application in C++ that interfaces with the Linux operating system to manage files and directories.

Day-wise Tasks:
Day 1: Design the application structure and setup the development environment. Start with basic file operations like listing files in a directory.
Day 2: Implement file and directory navigation features.Enable the user to move through directories.
Day 3: Add file manipulation capabilities (copy, move, delete, create).
Day 4: Implement file search functionality within the file explorer.
Day 5: Add file permission management features.

Code:
File_explorer.cpp

```
#include <iostream>
#include <filesystem>
#include <fstream>
#include <string>
#include <cstdio>   // for remove, rename
using namespace std;
namespace fs = std::filesystem;
string currentPath = fs::current_path().string();
// --- Function to List Files ---
void listFiles() {
   cout << "\nFiles and Directories in: " << currentPath << "\n";
   for (const auto& entry : fs::directory_iterator(currentPath)) {
      cout << (entry.is_directory() ? "[DIR]  " : "[FILE] ")
          << entry.path().filename().string() << endl;
   }
}
// --- Function to Change Directory ---
void changeDirectory() {
   string dir;
   cout << "\nEnter directory name (.. for parent): ";
   cin >> dir;
   fs::path newPath = fs::path(currentPath) / dir;
   if (dir == "..") newPath = fs::path(currentPath).parent_path();

   if (fs::exists(newPath) && fs::is_directory(newPath)) {
      currentPath = fs::canonical(newPath).string();
      cout << "Changed directory to: " << currentPath << endl;
   } else {
      cout << "Invalid directory!" << endl;
   }
}
// --- Function to Create File ---
void createFile() {
```

```
    string filename;
    cout << "\nEnter file name to create: ";
    cin >> filename;
    ofstream file(currentPath + "/" + filename);
    if (file) {
        cout << "File created successfully.\n";
    } else {
        cout << "Error creating file.\n";
    }
}
// --- Function to Copy File ---
void copyFile() {
    string src, dest;
    cout << "\nEnter source filename: ";
    cin >> src;
    cout << "Enter destination filename: ";
    cin >> dest;
try {
        fs::copy(currentPath + "/" + src, currentPath + "/" + dest, fs::copy_options::overwrite_existing);
        cout << "File copied successfully.\n";
    } catch (fs::filesystem_error& e) {
        cout << "Error: " << e.what() << endl;
    }
}
// --- Function to Move File ---
void moveFile() {
    string src, dest;
    cout << "\nEnter source filename: ";
    cin >> src;
    cout << "Enter destination filename: ";
    cin >> dest;
try {
        fs::rename(currentPath + "/" + src, currentPath + "/" + dest);
        cout << "File moved successfully.\n";
    } catch (fs::filesystem_error& e) {
        cout << "Error: " << e.what() << endl;
    }
}
// --- Function to Delete File ---
void deleteFile() {
    string filename;
    cout << "\nEnter filename to delete: ";
    cin >> filename;
try {
        fs::remove(currentPath + "/" + filename);
        cout << "File deleted successfully.\n";
    } catch (fs::filesystem_error& e) {
        cout << "Error: " << e.what() << endl;
```

```
 }
}
// --- Function to Search File (Recursive) ---
void searchFile() {
   string filename;
   cout << "\nEnter filename to search: ";
   cin >> filename;

   bool found = false;
   for (const auto& entry : fs::recursive_directory_iterator(currentPath)) {
      if (entry.path().filename() == filename) {
         cout << "Found: " << entry.path().string() << endl;
         found = true;
      }
   }
   if (!found) cout << "File not found.\n";
}
// --- Function to Show File Permissions ---
void showPermissions() {
   string filename;
   cout << "\nEnter filename: ";
   cin >> filename;
   fs::path filePath = currentPath + "/" + filename;

   if (!fs::exists(filePath)) {
      cout << "File not found.\n";
      return;
   }
  auto perms = fs::status(filePath).permissions();
   cout << "Permissions for " << filename << ":\n";
   cout << ((perms & fs::perms::owner_read) != fs::perms::none ? "r" : "-");
   cout << ((perms & fs::perms::owner_write) != fs::perms::none ? "w" : "-");
   cout << ((perms & fs::perms::owner_exec) != fs::perms::none ? "x" : "-");
   cout << endl;
}
// --- Function to Change File Permission (basic) ---
void changePermission() {
   string filename;
   cout << "\nEnter filename: ";
   cin >> filename;
   fs::path filePath = currentPath + "/" + filename;
 if (!fs::exists(filePath)) {
      cout << "File not found.\n";
      return;
   }
cout << "1. Read-only\n2. Writable\nEnter choice: ";
   int ch;
   cin >> ch;
```

```
 try {
      if (ch == 1)
         fs::permissions(filePath, fs::perms::owner_read, fs::perm_options::replace);
      else if (ch == 2)
         fs::permissions(filePath, fs::perms::owner_all, fs::perm_options::replace);
      cout << "Permissions updated successfully.\n";
   } catch (...) {
      cout << "Failed to change permissions.\n";
   }
}

// --- MAIN MENU ---
int main() {
   int choice;
   do {
      cout << "\n========== FILE EXPLORER ==========\n";
      cout << "Current Directory: " << currentPath << "\n";
      cout << "1. List Files\n2. Change Directory\n3. Create File\n4. Copy File\n";
      cout << "5. Move File\n6. Delete File\n7. Search File\n8. Show Permissions\n";
      cout << "9. Change Permissions\n0. Exit\n";
      cout << "Enter your choice: ";
      cin >> choice;
switch (choice) {
         case 1: listFiles(); break;
         case 2: changeDirectory(); break;
         case 3: createFile(); break;
         case 4: copyFile(); break;
         case 5: moveFile(); break;
         case 6: deleteFile(); break;
         case 7: searchFile(); break;
         case 8: showPermissions(); break;
         case 9: changePermission(); break;
         case 0: cout << "Exiting...\n"; break;
         default: cout << "Invalid choice!\n"; break;
      }
   } while (choice != 0);

   return 0;
}
```

OUTPUT:

```
   FILE EXPLORER
   ==============

Current Directory:
C:\Users\Priyanka

1. List Files
2. Change Directory
3. Create File
4. Copy File
5. Move File
6. Delete File
7. Search File
8. Show Permissions
9. Change Permissions
0. Exit

Enter your choice: 1

Files and Directories in:
C:\Users\Priyanka
[DIR] Documents
[FILE notes.txt
```