#include <iostream>

#include <fstream>

#include <string>

#include <filesystem>

using namespace std;

namespace fs = filesystem;

class Logger {

public:

Logger(const string& filename) : logFile(filename, ios::app) {}

void log(const string& message) {

cout << message << endl; // Print to console

logFile << message << endl; // Write to log file

}

private:

ofstream logFile;

};

void displayMenu();

string getUserCommand();

void processCommand(const string& command, string& currentPath, Logger& logger);

void changeDirectory(const string& path, string& currentPath, Logger& logger);

int main() {

bool isRunning = true;

string command;

string currentPath = ".";

Logger logger("logs/client\_log.txt");

while (isRunning) {

displayMenu();

command = getUserCommand();

if (command == "exit") {

isRunning = false;

logger.log("Exiting application.");

} else {

processCommand(command, currentPath, logger);

}

}

return 0;

}

void displayMenu() {

cout << "File Explorer Menu:" << endl;

cout << "1. List directory contents (ls)" << endl;

cout << "2. Create a file (createfile)" << endl;

cout << "3. Create a directory (createdir)" << endl;

cout << "4. Rename a file/directory (rename)" << endl;

cout << "5. Delete a file/directory (delete)" << endl;

cout << "6. View file contents (viewfile)" << endl;

cout << "7. Change directory (cd)" << endl;

cout << "Type 'exit' to quit the application." << endl;

}

string getUserCommand() {

string command;

cout << "Enter command: ";

cin >> command;

return command;

}

void processCommand(const string& command, string& currentPath, Logger& logger) {

if (command == "cd") {

string path;

cout << "Enter directory path: ";

cin >> path;

changeDirectory(path, currentPath, logger);

} else {

logger.log("Invalid command!");

}

}

void changeDirectory(const string& path, string& currentPath, Logger& logger) {

fs::path targetPath(path);

if (fs::exists(targetPath) && fs::is\_directory(targetPath)) {

logger.log("Changing directory from " + currentPath + " to " + path);

currentPath = targetPath;

fs::current\_path(targetPath);

logger.log("Directory changed successfully to " + currentPath);

} else {

logger.log("Directory does not exist: " + path);

}

}