**Logger Code for this File Explorer Application:**

#include <iostream>

#include <fstream>

#include <string>

#include <ctime>

#include <sstream>

#include <iomanip>

/\*\*

\* @brief Simple Logger class for file and console logging

\*/

class SimpleLogger {

public:

enum Level { DEBUG, INFO, NOTICE, WARNING, ERROR, CRITICAL };

SimpleLogger(const std::string& name) : mName(name) {

mLogFile.open("file\_explorer\_log.txt", std::ios::out | std::ios::app);

}

~SimpleLogger() {

if (mLogFile.is\_open()) {

mLogFile.close();

}

}

void log(Level level, const std::string& message) {

std::string levelStr = getLevelString(level);

std::string logMessage = getCurrentTime() + " [" + levelStr + "] " + mName + ": " + message;

std::cout << logMessage << std::endl; // Print to console

if (mLogFile.is\_open()) {

mLogFile << logMessage << std::endl; // Write to file

}

}

void debug(const std::string& message) { log(DEBUG, message); }

void info(const std::string& message) { log(INFO, message); }

void notice(const std::string& message) { log(NOTICE, message); }

void warning(const std::string& message) { log(WARNING, message); }

void error(const std::string& message) { log(ERROR, message); }

void critical(const std::string& message) { log(CRITICAL, message); }

private:

std::string mName;

std::ofstream mLogFile;

std::string getLevelString(Level level) {

switch (level) {

case DEBUG: return "DEBUG";

case INFO: return "INFO";

case NOTICE: return "NOTICE";

case WARNING: return "WARNING";

case ERROR: return "ERROR";

case CRITICAL: return "CRITICAL";

default: return "UNKNOWN";

}

}

std::string getCurrentTime() {

std::time\_t now = std::time(nullptr);

std::tm\* localTime = std::localtime(&now);

std::ostringstream oss;

oss << std::put\_time(localTime, "%Y-%m-%d %H:%M:%S");

return oss.str();

}

};

/\*\*

\* @brief Example main function demonstrating logging in a File Explorer Application

\*/

int main() {

SimpleLogger logger("FileExplorer");

// Day 1: Logging basic file operations

logger.info("Day 1: Starting basic file operations.");

logger.debug("Listed files in directory '/home/user'");

// Day 2: Logging directory navigation

logger.info("Day 2: Implementing directory navigation.");

logger.debug("Navigated to directory '/home/user/documents'");

// Day 3: Logging file manipulation capabilities

logger.info("Day 3: Adding file manipulation capabilities.");

logger.debug("Copied file 'example.txt' to '/home/user/documents/example\_copy.txt'");

// Day 4: Logging file search functionality

logger.info("Day 4: Implementing file search functionality.");

logger.debug("Searched for 'report.docx' in '/home/user'");

// Day 5: Logging file permission management

logger.info("Day 5: Adding file permission management features.");

logger.debug("Changed permissions of 'example.txt' to 'rwxr-xr-x'");

return 0;

}

**Output:**

