cmake\_minimum\_required(VERSION 3.18)

project(FileServerProject LANGUAGES CXX)

# C++ standard

set(CMAKE\_CXX\_STANDARD 17)

set(CMAKE\_CXX\_STANDARD\_REQUIRED ON)

set(CMAKE\_CXX\_EXTENSIONS OFF)

# Vcpkg Integration

# Quan trọng: Đảm bảo CMAKE\_TOOLCHAIN\_FILE được thiết lập ĐÚNG CÁCH

# khi bạn chạy lệnh cmake từ dòng lệnh, ví dụ:

# cmake .. -DCMAKE\_TOOLCHAIN\_FILE=/path/to/your/vcpkg/scripts/buildsystems/vcpkg.cmake

# Đoạn code dưới đây chỉ là một nỗ lực "tìm" toolchain file, nhưng không phải lúc nào cũng đủ.

if(NOT DEFINED CMAKE\_TOOLCHAIN\_FILE AND DEFINED ENV{VCPKG\_ROOT})

set(CMAKE\_TOOLCHAIN\_FILE "$ENV{VCPKG\_ROOT}/scripts/buildsystems/vcpkg.cmake"

CACHE STRING "Vcpkg toolchain file")

message(STATUS "Attempting to use VCPKG\_ROOT for toolchain: $ENV{VCPKG\_ROOT}/scripts/buildsystems/vcpkg.cmake")

elseif(NOT DEFINED CMAKE\_TOOLCHAIN\_FILE AND EXISTS "${CMAKE\_CURRENT\_SOURCE\_DIR}/vcpkg/scripts/buildsystems/vcpkg.cmake")

set(CMAKE\_TOOLCHAIN\_FILE "${CMAKE\_CURRENT\_SOURCE\_DIR}/vcpkg/scripts/buildsystems/vcpkg.cmake"

CACHE STRING "Vcpkg toolchain file from submodule")

message(STATUS "Attempting to use vcpkg submodule for toolchain: ${CMAKE\_CURRENT\_SOURCE\_DIR}/vcpkg/scripts/buildsystems/vcpkg.cmake")

endif()

# --- THÊM CÁC LỆNH find\_package() VÀO ĐÂY ---

find\_package(Poco REQUIRED COMPONENTS Net Util Foundation JSON XML)

find\_package(unofficial-sqlite3 CONFIG REQUIRED) # Tên package cho sqlite3 từ vcpkg có thể là unofficial-sqlite3

# Hoặc nếu bạn cài sqlite3 trực tiếp, có thể là SQLite3

find\_package(OpenSSL REQUIRED)

find\_package(nlohmann\_json REQUIRED)

# ---------------------------------------------

# Source files

file(GLOB\_RECURSE SERVER\_SOURCES CONFIGURE\_DEPENDS

src/\*.cpp

)

# Executable

add\_executable(file\_server ${SERVER\_SOURCES})

# Header files

target\_include\_directories(file\_server PRIVATE

${CMAKE\_CURRENT\_SOURCE\_DIR}/include

)

# Link thư viện từ vcpkg

target\_link\_libraries(file\_server PRIVATE

Poco::Net # Imported target từ find\_package(Poco)

Poco::Util

Poco::Foundation

Poco::JSON # Nếu bạn dùng find\_package(Poco ... COMPONENTS JSON)

Poco::XML # Nếu bạn dùng find\_package(Poco ... COMPONENTS XML)

unofficial::sqlite3::sqlite3 # Target cho unofficial-sqlite3

# Hoặc nếu là SQLite3 thì có thể là SQLite::SQLite3

OpenSSL::SSL

OpenSSL::Crypto

nlohmann\_json::nlohmann\_json # Target cho nlohmann\_json

)

# Output paths

set\_target\_properties(file\_server PROPERTIES

RUNTIME\_OUTPUT\_DIRECTORY ${CMAKE\_SOURCE\_DIR}/run

ARCHIVE\_OUTPUT\_DIRECTORY ${CMAKE\_BINARY\_DIR}/lib # Không thực sự cần cho executable

LIBRARY\_OUTPUT\_DIRECTORY ${CMAKE\_BINARY\_DIR}/lib # Không thực sự cần cho executable

)

# Unit tests

enable\_testing()

add\_subdirectory(test)

# Installation rules (tùy chọn)

install(TARGETS file\_server DESTINATION bin)

# Debug messages (để lại hoặc xóa đi sau khi debug xong)

message(STATUS "CMAKE\_TOOLCHAIN\_FILE is: ${CMAKE\_TOOLCHAIN\_FILE}")

message(STATUS "Poco\_FOUND: ${Poco\_FOUND}, Poco\_DIR: ${Poco\_DIR}")

message(STATUS "unofficial-sqlite3\_FOUND: ${unofficial-sqlite3\_FOUND}, unofficial-sqlite3\_DIR: ${unofficial-sqlite3\_DIR}")

# message(STATUS "SQLite3\_FOUND: ${SQLite3\_FOUND}, SQLite3\_DIR: ${SQLite3\_DIR}") # Nếu bạn thử SQLite3

message(STATUS "OpenSSL\_FOUND: ${OpenSSL\_FOUND}, OpenSSL\_DIR: ${OpenSSL\_DIR}")

message(STATUS "nlohmann\_json\_FOUND: ${nlohmann\_json\_FOUND}, nlohmann\_json\_DIR: ${nlohmann\_json\_DIR}")