Sample project structure

```
project_root/
├── CMakeLists.txt
├── src/
│   ├── include/
│   │   └── your_project/
│   │       ├── header1.hpp
│   │       └── header2.hpp
│   └── source1.cpp
│   └── source2.cpp
├── tests/
│   ├── CMakeLists.txt
│   ├── unit/
│   │   ├── test_feature1.cpp
│   │   └── test_feature2.cpp
│   └── integration/
│       ├── test_integration1.cpp
│       └── test_integration2.cpp
│
├── build/
└── README.md
```

You'll need two CMakeLists.txt files - one in the root directory and one in the tests directory.

Root `CMakeLists.txt`

```
cmake_minimum_required(VERSION 3.14)
project(your_project_name VERSION 1.0)

# Specify C++ standard
set(CMAKE_CXX_STANDARD 17)
set(CMAKE_CXX_STANDARD_REQUIRED ON)

# Add source files
add_library(${PROJECT_NAME}
    src/source1.cpp
    src/source2.cpp
)

# Add include directories
target_include_directories(${PROJECT_NAME} PUBLIC
    ${CMAKE_CURRENT_SOURCE_DIR}/src/include
)

# Enable testing
enable_testing()

# Add the tests subdirectory
add_subdirectory(tests)
```

Tests `CMakeLists.txt`

```
# Include FetchContent for downloading GoogleTest
include(FetchContent)
FetchContent_Declare(
    googletest
    URL https://github.com/google/googletest/archive/refs/tags/v1.14.0.zip
)
# Configure GoogleTest
FetchContent_MakeAvailable(googletest)
include(GoogleTest)

# Create test executable
add_executable(unit_tests
    unit/test_feature1.cpp
    unit/test_feature2.cpp
)

# Link against GoogleTest and your main library
target_link_libraries(unit_tests
    PRIVATE
    GTest::gtest_main
    ${PROJECT_NAME}
)

# Discover and run tests
gtest_discover_tests(unit_tests)
```

Usage

Create the directory structure:

```
mkdir -p src/include/your_project tests/unit build
```

2. Create the source files:

```
touch src/source1.cpp src/source2.cpp
```

3. Create the test files:

```
touch tests/unit/test_feature1.cpp
```

2. Place your source files in the appropriate directories:

```
mkdir -p src/include/your_project
```

3. Create the header files:

```
touch src/include/your_project/header1.h src/include/your_project/header2.h
```

3. Build the project:

```
cd build
cmake ..
cmake --build .
```

4. Run the tests: ctest

or directly run the test executable:

```
./tests/unit_tests
```