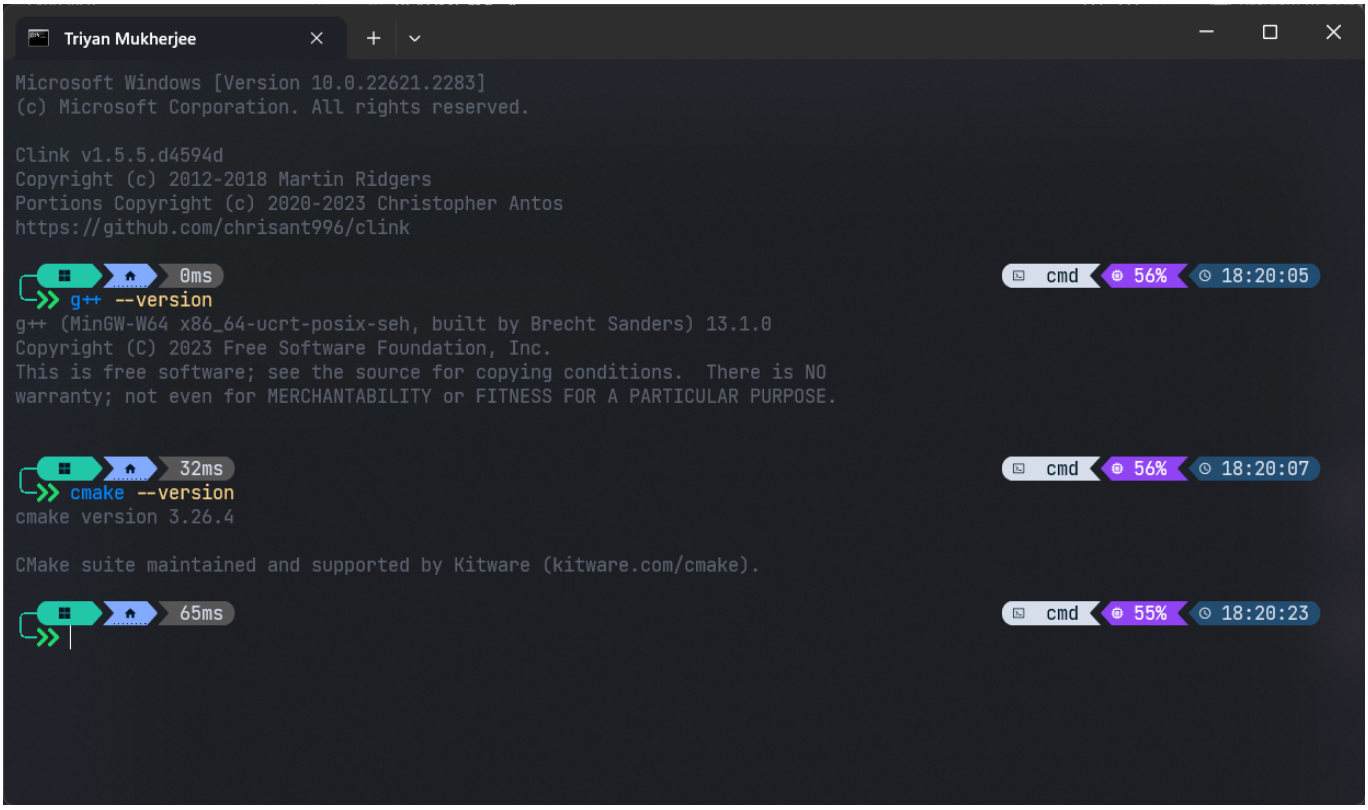
Using CMake with Visual Studio Code

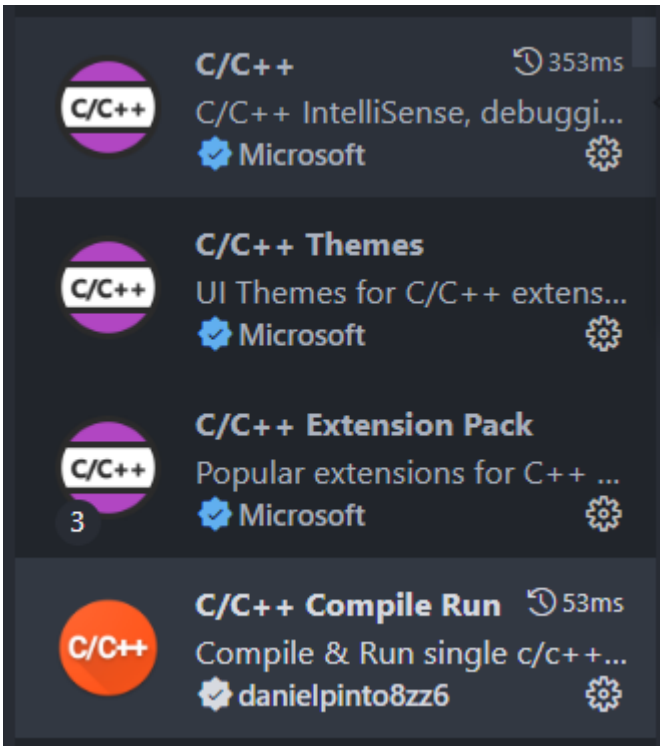
CMake is an open-source build system generator for software projects that allows developers to specify build parameters in a simple, portable, text file format.

CMake makes it easy to build projects with multiple files, multiple build configurations, and multiple platforms. By defining a common build specification, environments can be replicated across development, testing, and production.

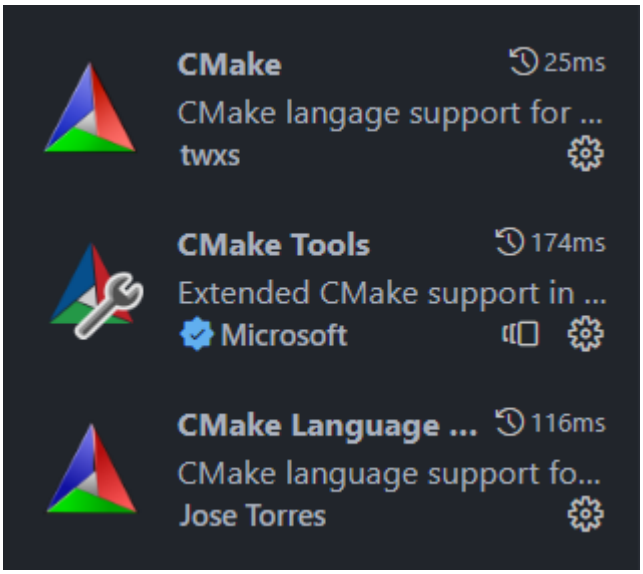
> **Note**
> Make sure you have the `C/C++` compiler installed from the previous tutorial. To verify run `g++ --version` and `cmake --version`.



Installing necessary extensions

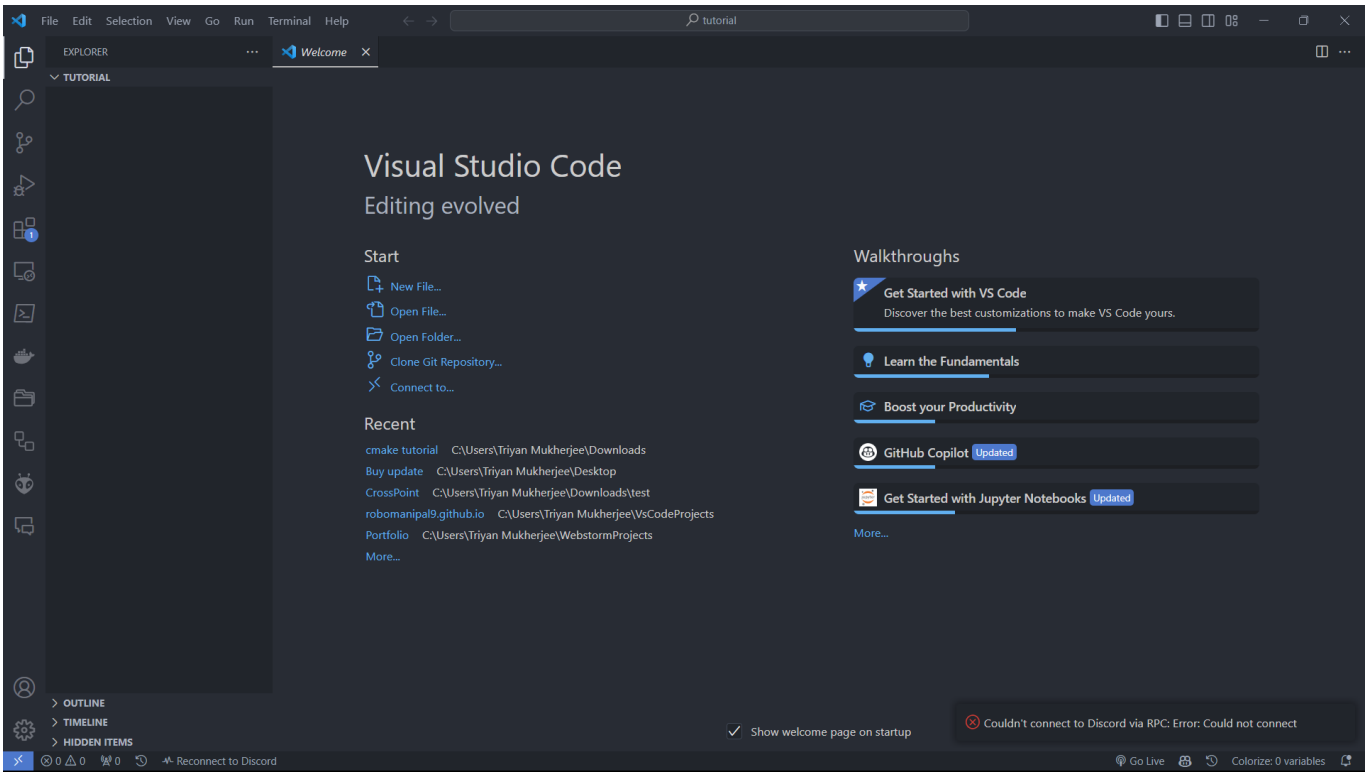


- C/C++ Compile Run extension is an added bonus which allows you to compile and run your code from within VS Code at click of a button.

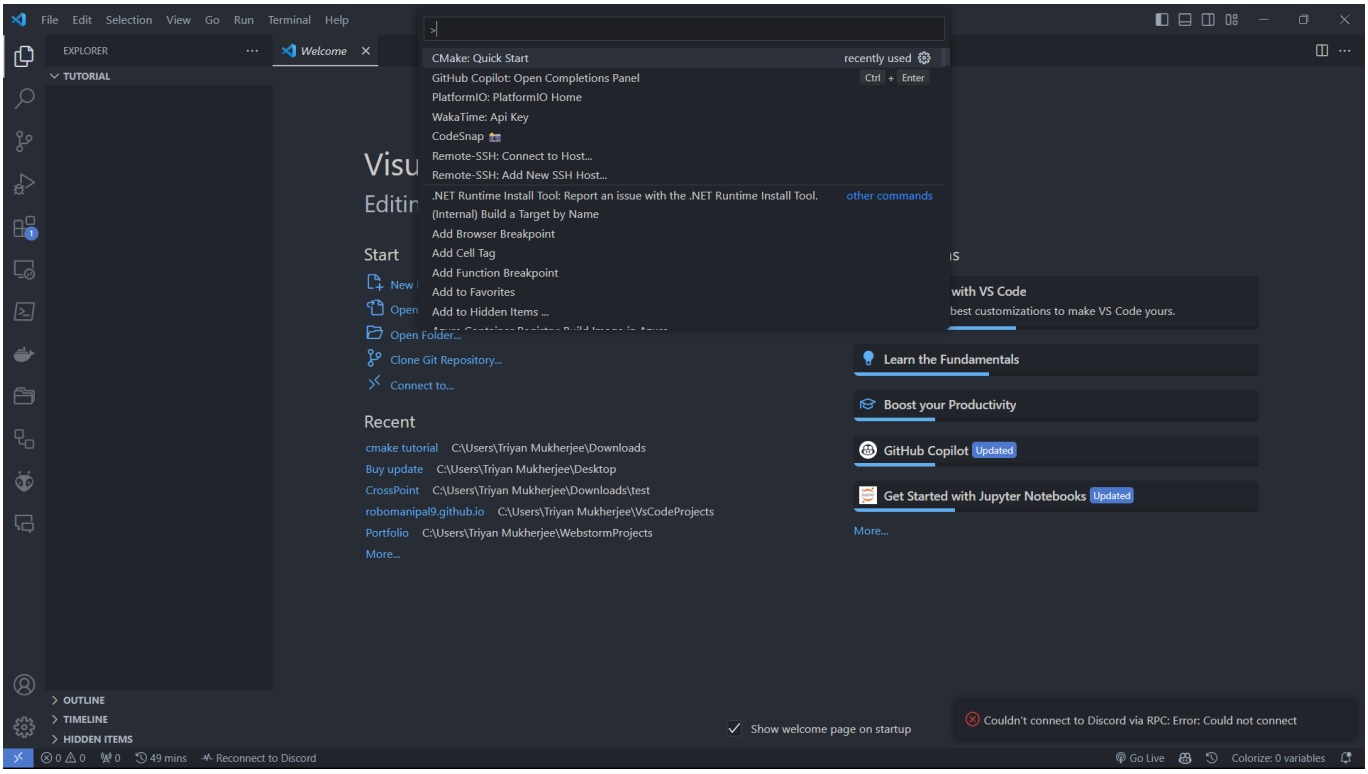


Setting up a Project

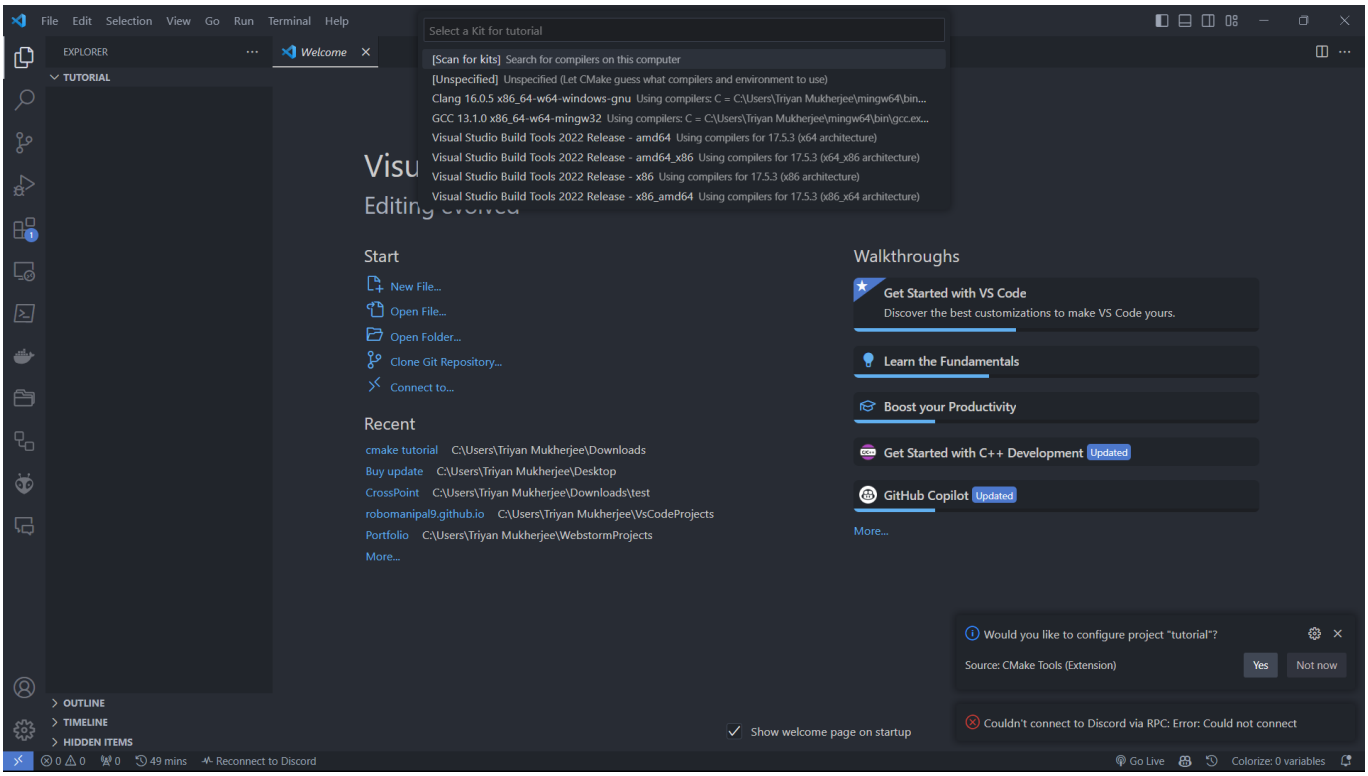
Create a folder for your project and open it in VS Code. In this case we will name the folder `tutorial`.



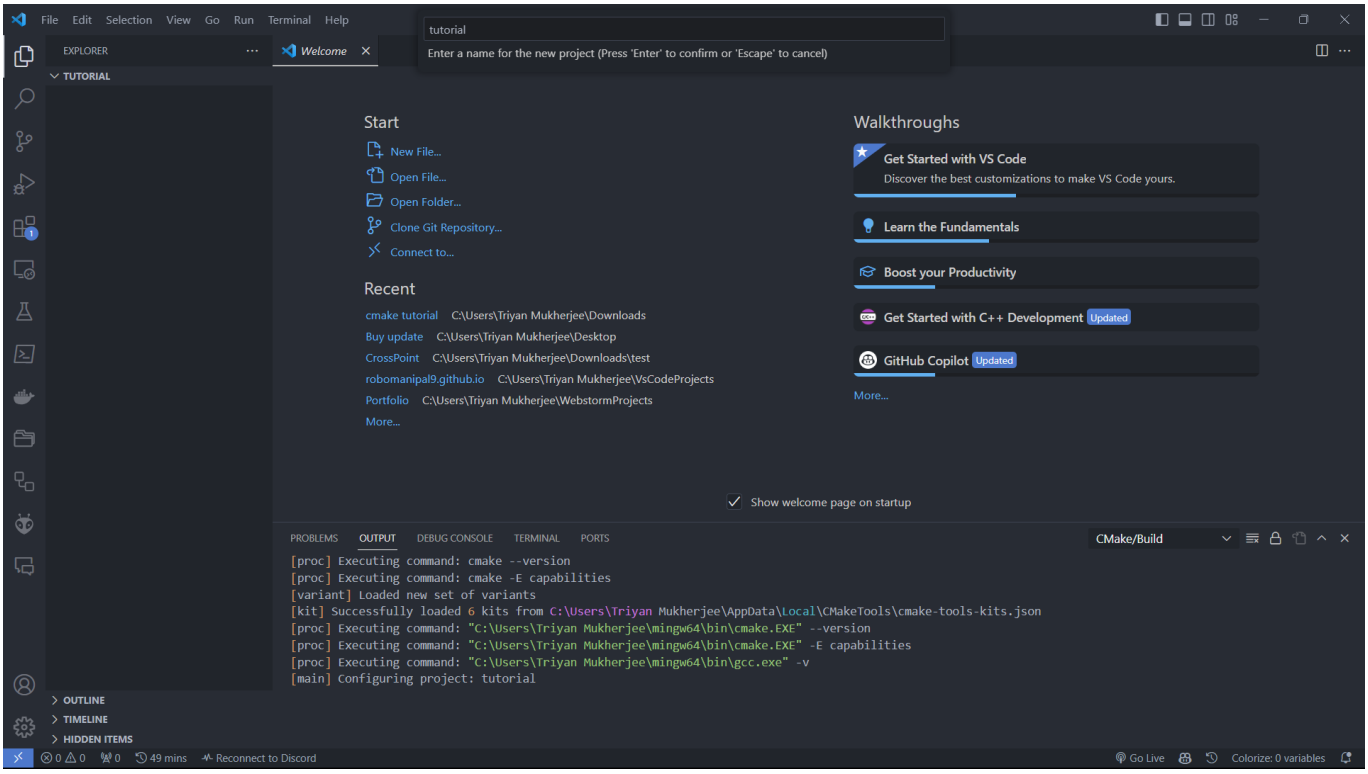
Now we are going to use the `cmake tools` extension to setup the project. Press `Ctrl+Shift+P` and type `cmake` and select `CMake: Quick Start`.



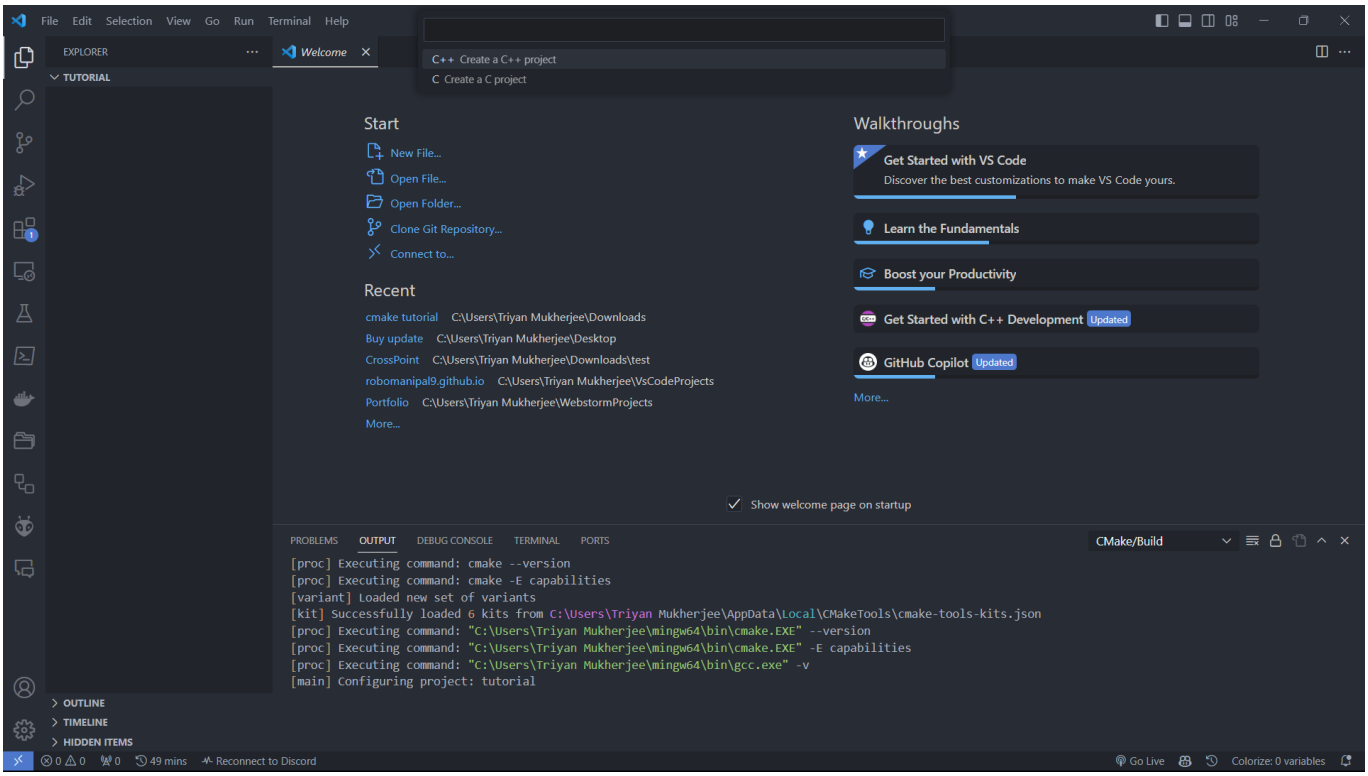
Select `quick start` and press `Enter`. You should see a similar prompt.



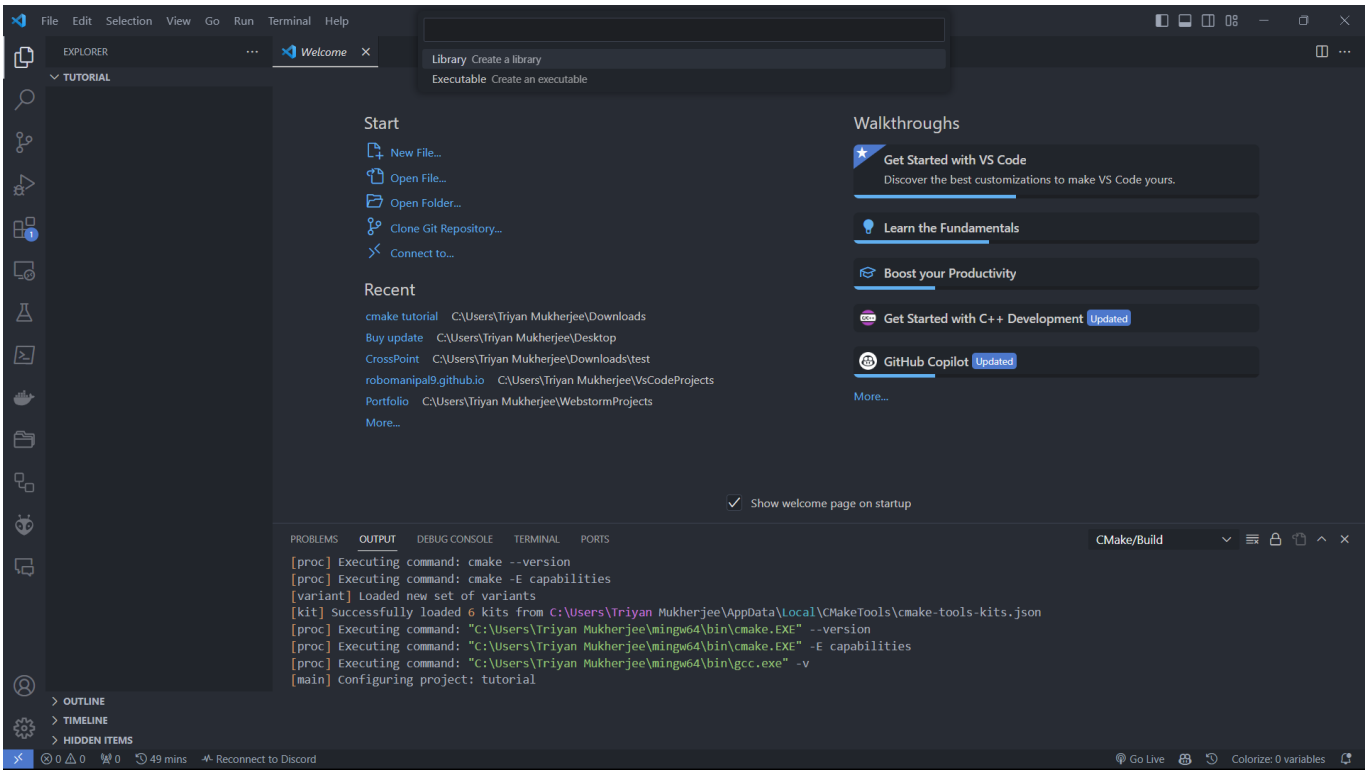
If your installation directory of `mingw` compiler doesn't show up, select `[Scan for kits]` and select the `mingw` compiler.



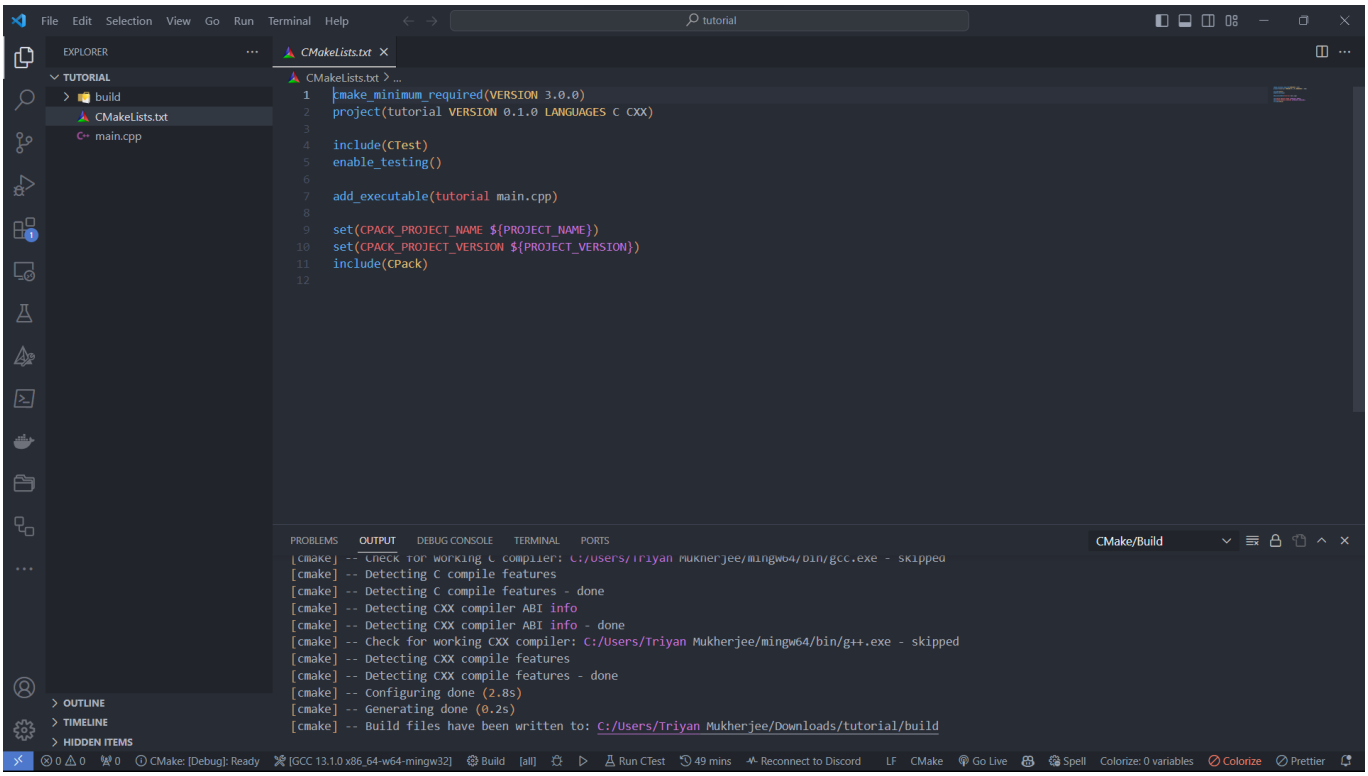
Enter your project name and press `Enter`.



Select a project type in this case we will be going with `C++`.

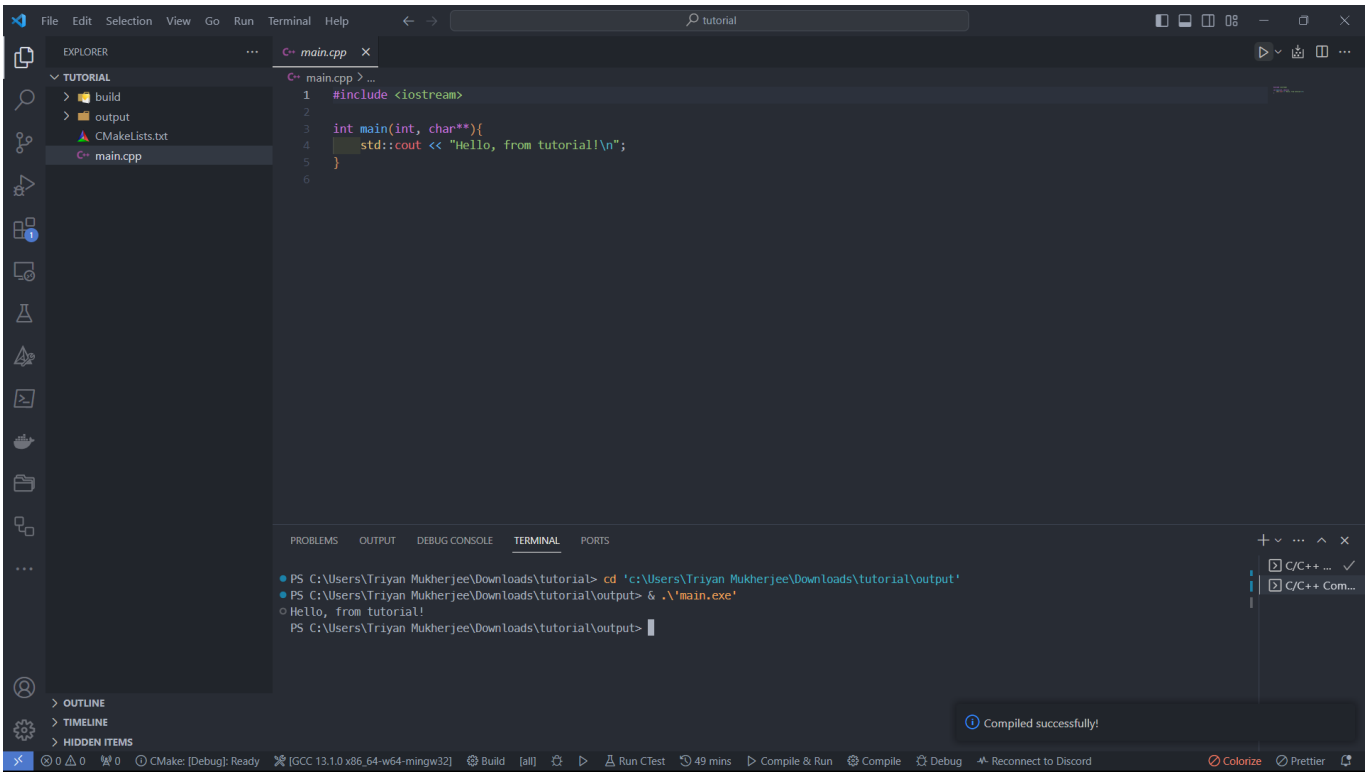


Select a build type in this case we will be going with `Executable`.



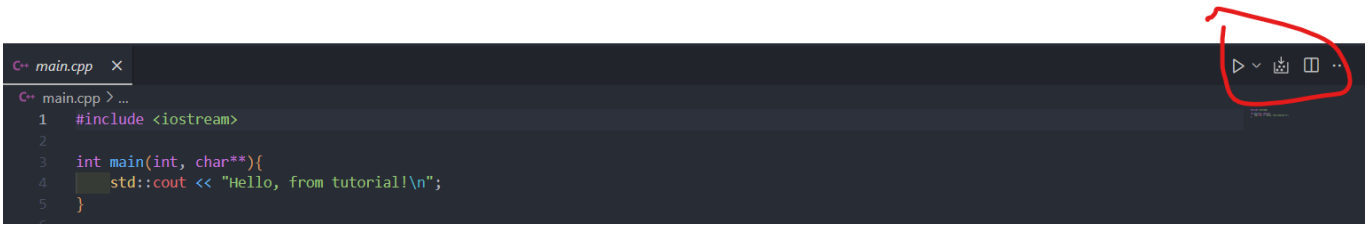
With this a basic setup of the project is completed.

Running the Project

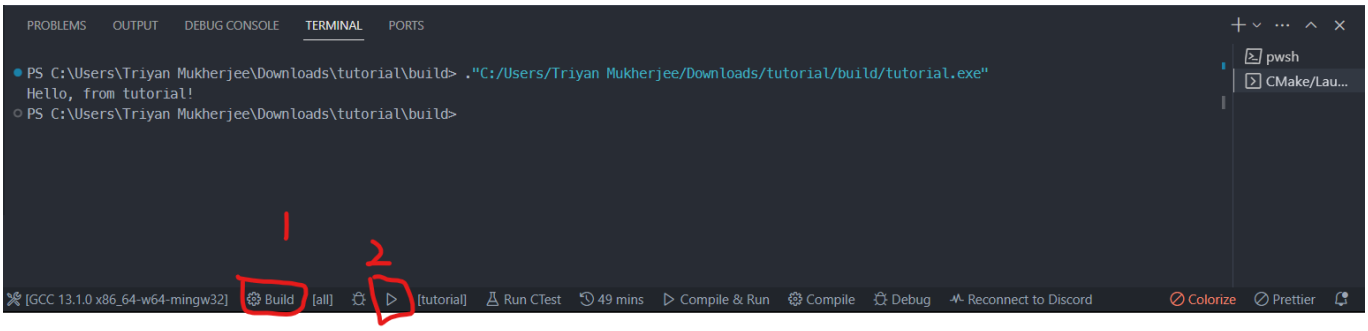


There are two ways to run the project.

- Using the previous runner extension installed.



- Using the cmake tools extension.



Additional Information

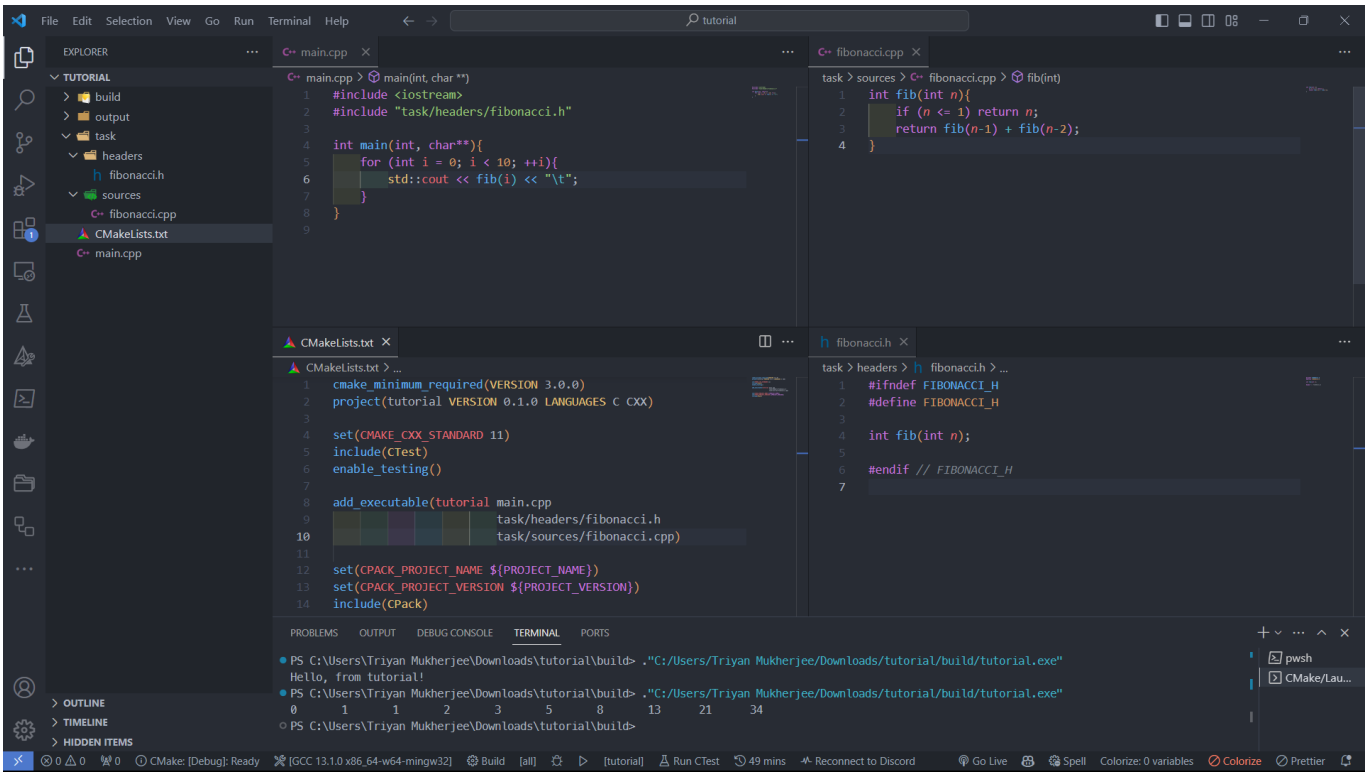
```
# Specify the minimum version for CMake
cmake_minimum_required(VERSION 3.0.0)
# Project's name and language mostly metadata
project(tutorial VERSION 0.1.0 LANGUAGES C CXX)

# Set the C++ standard to C++ 11
set(CMAKE_CXX_STANDARD 11)
# CTest is a testing tool that can be used to test your project.
include(CTest)
enable_testing()

# Add the executable. Add all the source files here.
add_executable(tutorial main.cpp)

set(CPACK_PROJECT_NAME ${PROJECT_NAME})
set(CPACK_PROJECT_VERSION ${PROJECT_VERSION})
include(CPack)
```

Here is an example with a few more files and folders.



> **Note**
> IDE's like `visual studio` and `clion` auto adds files in `CMakeLists.txt` but in VS Code you have to manually add them. PS `visual studio` and `visual studio code` are two different things.