Week 1: To create small CLI project using STL and CMAKE (e.g To-Do List)

Command-Line Interface (CLI)

✅ Step 1: Make sure your dev environment is set up

Run these in terminal:

```
bash
CopyEdit
sudo apt update
sudo apt install build-essential cmake gdb valgrind -y
```

> This installs: g++, make, cmake, gdb, and valgrind.

✅ Step 2: VS Code Setup (Optional Plugins)

Install the following **VS Code extensions**:

1. `C/C++` (by Microsoft) — IntelliSense + Debugging.
2. `CMake Tools` — Easy CMake integration.
3. `CodeLLDB` or `gdb` debugger (if needed).
4. (Optional) `Clang-Format` — for auto-formatting.

✅ Step 3: Folder Structure

Create a folder `todo-cli` like this:

```
bash
todo-cli/
├── CMakeLists.txt
└── src/
    └── main.cpp
```

You can create the desired folder structure in Ubuntu using the terminal with the following command:

```
bash
CopyEdit
mkdir -p todo-cli/src && touch todo-cli/CMakeLists.txt todo-cli/src/main.cpp
```

Explanation:

- `mkdir -p todo-cli/src`: Creates the folder `todo-cli` and the subfolder `src` (with `p` to make parent dirs as needed).
- `touch todo-cli/CMakeLists.txt`: Creates the `CMakeLists.txt` file.
- `touch todo-cli/src/main.cpp`: Creates the `main.cpp` file inside the `src` folder.

To verify:

You can use `tree` to view the structure:

```
bash
CopyEdit
tree todo-cli
```

If `tree` is not installed:

```
bash
CopyEdit
sudo apt install tree
```

What is CMake?

Step - 4 Write Code and logic

Link to Project →
https://github.com/RohitRaj911/Low_Latency_C_plus_plus/tree/master

✅ Step 6: Build & Run

From the project root:

```

mkdir build && cd build
cmake ..
make
./todo
```