**Final Project description**

**OrderBook.hpp file :**

Obj : Define the structure of an order (‘Order’), and of an orderbook (‘OrderBook’)

The OrderBook is defined as an std::unordered\_map : this provides a fast average-case time complexity for insertion, deletion and lookup operations (uses a hash function to organize elements, unlike std::map that keep elements in order). We use that because in this project we just have a best bid/ask per security. The **Key** of the hash function is the security name (string) and the **value** is a tuple (bid,ask).

In methods, we use the input ‘const std::string& sec\_id’ instead of ‘std::string sec\_id’ to avoid expensive copy (by giving the pointer) and const guarantees a read-only operation

In print\_book, the ‘const auto&’ allows the compiler to deduce the type and the structured binding ‘[sec\_id, prices]’ unpacks the pairs

**StopWatch.hpp:**

**StopWatch** gives you full manual control: start/stop multiple times, accumulate, pause, reset, and optionally label your timer.

**Timer** is minimal and geared for “time this one interval,” with easy conversions to millis, micros, etc.

**ScopedTimer** builds on Timer to eliminate the need to manually call Start() / Stop()—perfect for timing single scopes or functions without cluttering your code.

**Main\_client.cpp**

**Main\_server.cpp**