**How It Works**

The WebSocket client I've created meets all your requirements:

**WebSocket Client Features**

* **Connection**: Connects to secure WebSocket (WSS) servers, including the specified echo servers
* **Message Sending**: Allows sending text messages to the server
* **Message Receiving**: Continuously listens for and displays messages from the server
* **Command-line Interface**: Simple CLI for user interaction with commands
* **Error Handling**: Proper handling of connection issues and disconnections

**Implementation Details**

* **Core Components**:
* Uses Boost.Beast library for WebSocket protocol implementation
* Implements SSL/TLS support for secure connections
* Uses multiple threads to handle both sending and receiving messages
* **Key Classes**:
* WebSocketClient: Manages the WebSocket connection, sending and receiving messages
* URL parsing functionality to handle different server endpoints
* **GN Build System**:
* Separate build targets for debug and release configurations
* Debug configuration includes debug symbols and optimization disabled
* Release configuration includes optimization flags
* Proper compiler flags and library dependencies

**Using the Client**

* Run the application
* Enter a WebSocket server URL or use the default
* When connected, send messages that will be echoed back by the server
* Use /disconnect to disconnect or /quit to exit

The code is well-structured with error handling throughout to ensure graceful recovery from connection issues.

Would you like me to explain any specific part of the implementation in more detail?