ORDER EXECUTION AND MANAGEMENT SYSTEM TO TRADE ON DERIBIT TEST FOR GOQUANT COMPANY



REQUIREMENTS

- Authentication
- Placing orders
- Modifying orders
- Canceling orders
- Getting order book and positions.
- Real-Time Market Data
- WebSocket streaming for live market data

CHALLENGES AND SOLUTIONS

- Problems:

- Most implementations rely solely on CLI tools
- Lack of a user-friendly interface

- Solution:

 Create a web-based UI by enabling static file serving through the webserver

IMPLEMENTATION STEPS

- Build a C++ codebase to interact with the Deribit API
- Create a WebSocket server to serve HTML pages
- Integrate a webserver to handle API requests and serve static files

TECHNOLOGIES USED

Language:

• C++

Libraries:

- OpenSSL (for secure connections).
- NLohmann JSON (for parsing JSON responses).



PROBLEM SOLVING AND OPTIMIZATION

- Epoll for handling concurrent client connections
- Reduced JSON parsing overhead using NLohmann JSON

PROJECT STRUCTURE

File Descriptions:

- main.cpp: Entry point of the program.
- server.cpp: Implements the webserver to handle incoming requests.
- request.cpp: Handles request parsing and validation.
- api.cpp: Manages API requests and responses with Deribit.
- utils.cpp: Provides utility functions for various tasks.

Directories:

- includes: Contains all the header files.
- public: Stores static files to be served by the webserver

THANK YOU!

FOR YOUR ATTENTION