Тестовое задание: Senior C++ Developer (Game Server)

Это тестовое задание рассчитано примерно на 8 часов выполнения и призвано оценить следующие навыки кандидата:

- Умение писать чистый, поддерживаемый и хорошо продуманный архитектурно код в стиле существующего проекта *(см. пример в файлах)*.

- Знание стандартной библиотеки C++ (STL) и понимание алгоритмической сложности при работе с контейнерами.

- Базовые навыки написания SQL-запросов (SQL, MySQL).

- Общее знание современного C++ (C++23) и библиотек Boost, TBB, libev.

- Понимание принципов работы с сетевыми протоколами (в частности, WebSocket и самописный протокол на базе ULEB).

Часть 1. Код-ревью

Дано: фрагмент исходного кода сервера (файл Client.cpp).

Задание:

- Найти и кратко описать (в 2–3 предложениях) **три участка кода**, которые можно улучшить с точки зрения читаемости, архитектуры или производительности.

- Предложить и реализовать улучшения **в одном** из этих участков кода – в виде патча или отдельного нового файла с исправлениями.

Часть 2. Реализация функциональности

Задание:

- Реализовать мини-модуль TopTracker, который хранит N последних действий игроков с привязкой ко времени.

- В модуле должны быть реализованы операции: добавление нового действия, удаление старых действий по истечении тайм-аута (очистка "просроченных" записей), и получение текущего списка сохранённых действий.

- Использовать стандартные контейнеры (например, std::deque, std::unordered_map) – необходимо объяснить выбор тех или иных структур данных.

- Минимизировать внешние зависимости – желательно использовать только стандартную библиотеку C++ и, при необходимости, Boost (например, Boost.Timer/Chrono) для работы со временем.

- Реализовать unit-тесты для разработанного модуля TopTracker.

- Код модуля и тестов должен быть написан **в том же стиле**, что и существующие файлы проекта (Client.cpp/h), включая соглашения по отступам, именованию и общей архитектуре решения.

Часть 3. SQL-задание

Дано: набор данных в таблице players со структурой полей:
players(id INT, name VARCHAR, login_time DATETIME, device INT)

Задание: сформулировать SQL-запросы для следующих случаев:

- Получить **5 самых активных устройств** – то есть пять значений поля device, с которых произошло наибольшее количество логинов пользователей (упорядочить по количеству логинов).

- Получить **среднее число логинов в день** за последние 7 дней (в расчете от текущей даты).

Формат сдачи

- Исходный код (и при желании сопутствующие материалы) предоставить в виде одного архива либо ссылки на Git-репозиторий. В репозитории должен быть четко выделен код решения каждой части задания (Part 1, Part 2, Part 3).

- Для каждого модуля или части решения добавить файл README с кратким описанием использованной архитектуры, принятых решений и пояснениями по запуску (если применимо), чтобы проверяющие могли легко разобраться в решении.

Пример кода:
https://rockstonedev.com/tz/Client.cpp
https://rockstonedev.com/tz/Client.h