**RCS C++ Assignment**

MYLAPS Sports Timing offers the best-in-class sports timing systems to measure, publish and analyze race and practice results for all sports. We are extending this portfolio with products for Race Control. In this assignment we ask you to create a simple piece of software that demonstrates how you build resilient, testable code. It does not have to be a fully functioning, polished solution.

Our new product is a client/server system that processes and distributes events occurring around the track. It integrates with multiple other systems (like CCTV and GPS Positioning systems). In this assignment we would like to see how you would implement likewise systems.

**Description**

* A race consists of several drivers racing around the track for a fixed number of laps (driving from the start/finish line to the start/finish line).
* The driver’s number and the time of day are measured by the timing system each time a driver passes the start/finish line of the racing track. The API should accept these lap times in the form of the attached karttimes.csv file.
* The race is finished when one of the drivers completes all laps.
* The winner of a race is the driver who has driven the fastest lap.

**Result**

* Server should be able to return the following information:
  + Ranking of the drivers, including the fastest lap of the winner and a sorted list of time difference from the winning lap and fastest lap of other drivers.
    - For example:
      * 1st: 20s
      * 2nd: +1s
      * 3rd: +3s
  + Average lap time of any requested driver

**Requirements**

* A server application which calculates the result (and some small analytics) of a race based on lap times collected during a race.
* A client that can provide the lap times as csv (see attachment) and show the results calculated by server.
* Server side of the assignment needs to be written in C++.
* On the client side, you can use the front-end technology of your choice, at mylaps we use C++/Qt to develop our desktop app.
* Using external libraries is allowed, just make sure that the project contains a compilation manual or use automation of your choice for compiling or containerizing it.
* Make sure that your project is accessible on a git repository.
* Client and server logic should work asynchronously.
* Keep error handling in mind.

**Notes**

* Your assignment will be evaluated on creativity, technical design, code quality and commit hygiene.
* We’d like to see how you guard functional quality from degrading (test automation).
* Please do not spend more than 8 hours on this assessment.
* You can always reach out and ask questions about the assignment, also feel welcome to leave comments or documentation to explain your thoughts if needed