Performance Report

RAD

Ben Wearn

2021

# Code Optimisation:

Code optimisation is any code modification that can improve code quality and efficiency. A program may be optimised to become smaller in size, consume less memory, executes faster, or performs fewer input/output operations.

One appropriate code optimiser that could be beneficial to the project is DataDog. This code optimiser is a code optimiser for PHP through profiling. This software is used to detect the most resource-consuming methods or classes in PHP code.

This software allows users to have complete visibility into PHP performance, spend less time troubleshooting and, optimise legacy and cloud applications as it can pinpoint CPU, memory and latency bottlenecks, analyse code methods, classes and threads, allows visibility into code levels performance in development, staging and production environments and, identify performance issues such as blocked threads or memory leaks with an automatic analysis system.

Timeline

Description automatically generated

# Performance Tools:

Performance tools support development in tuning the applications performance. They measure performance data during execution and provide analysis of the data to detect performance bottle necks and map optimisation opportunities. These tools uncover inefficient areas in code that take up critical resources which result in higher costs and inefficient performance.

There are many performance tools that could be used throughout the project, these being gProfiler, SonarQube, and another software from DataDog. For this project, the choice of gProfiler was made, this is because it is an open-source profiler for any environment and it is free and purposed for collaboration. gProfiler continuously analyses code performance across the entire environment to optimise the most resource-consuming areas, improve performance and reduce costs and takes on an always-on approach which continuously samples CPU performance.

Graphical user interface, text, application

Description automatically generated