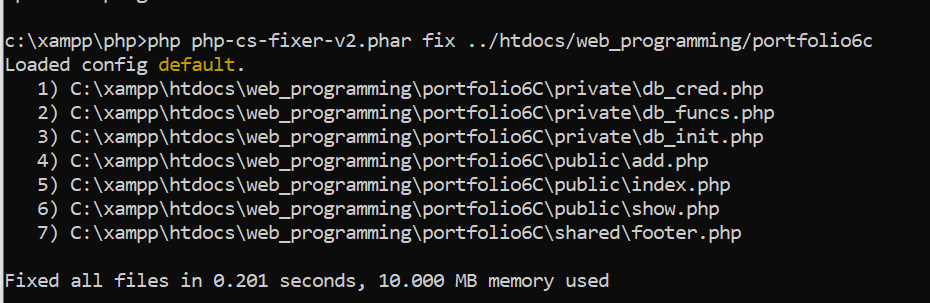
## Performance Report

### Code Formatting:

The PHP Coding Standards Fixer is a good choice for the project for the enforcement of coding standards. This tool is simple to install and use, and allows the development team to find all code formatting errors and apply the necessary fixes to them, depending on what coding standard has been selected.

To install, it can be simply downloaded from <https://cs.symfony.com> and extracted to the desired directory. To use, run the command in the terminal as follows:  
- php php-cs-fixer.phar fix “/path/to/dir”   
Where the part in quotation marks is the directory or individual file where the tool will be run.

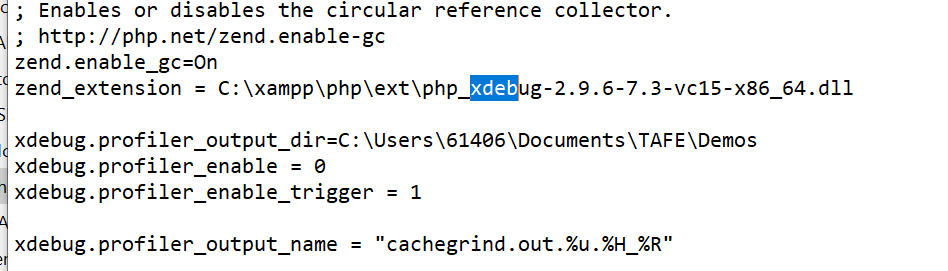
Example of how it is used and the result:



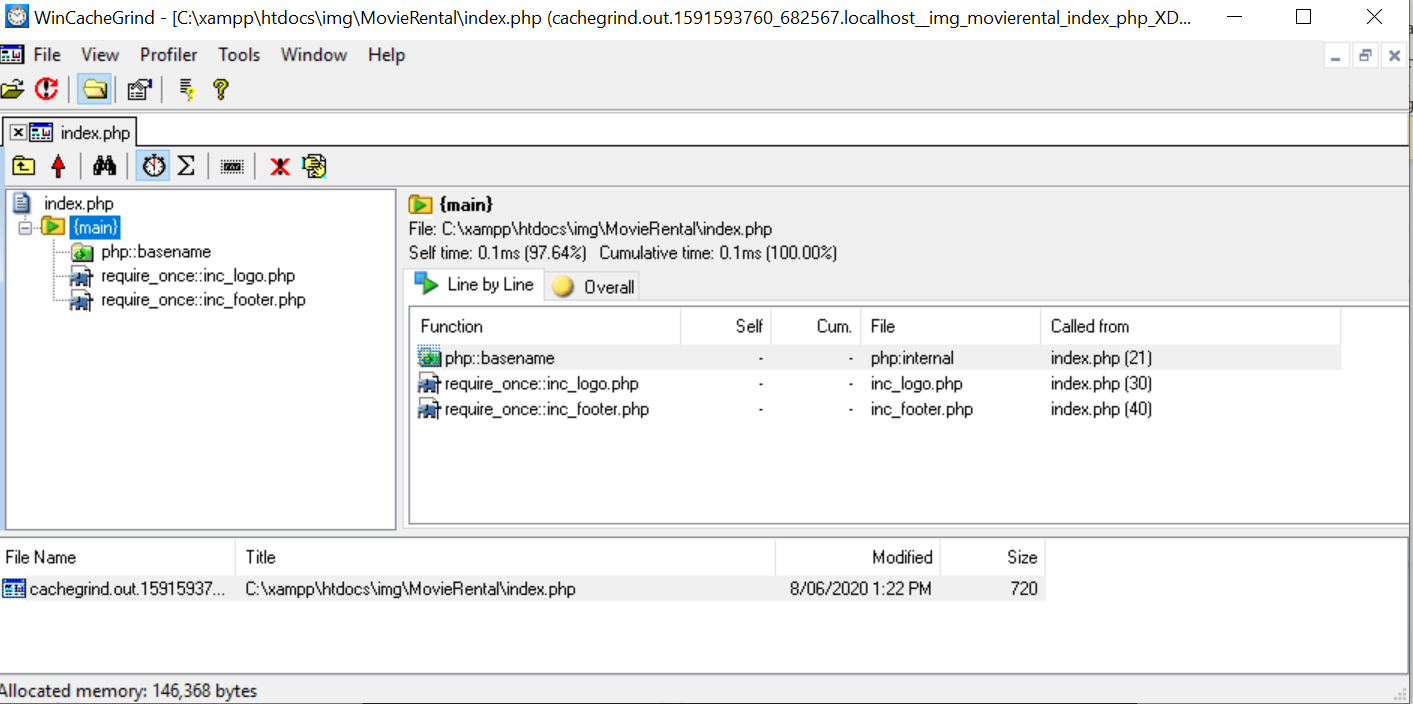
For our project, we will be using the default coding standards because they suitably reformat code to a standard accepted by our team.

### Profiler:

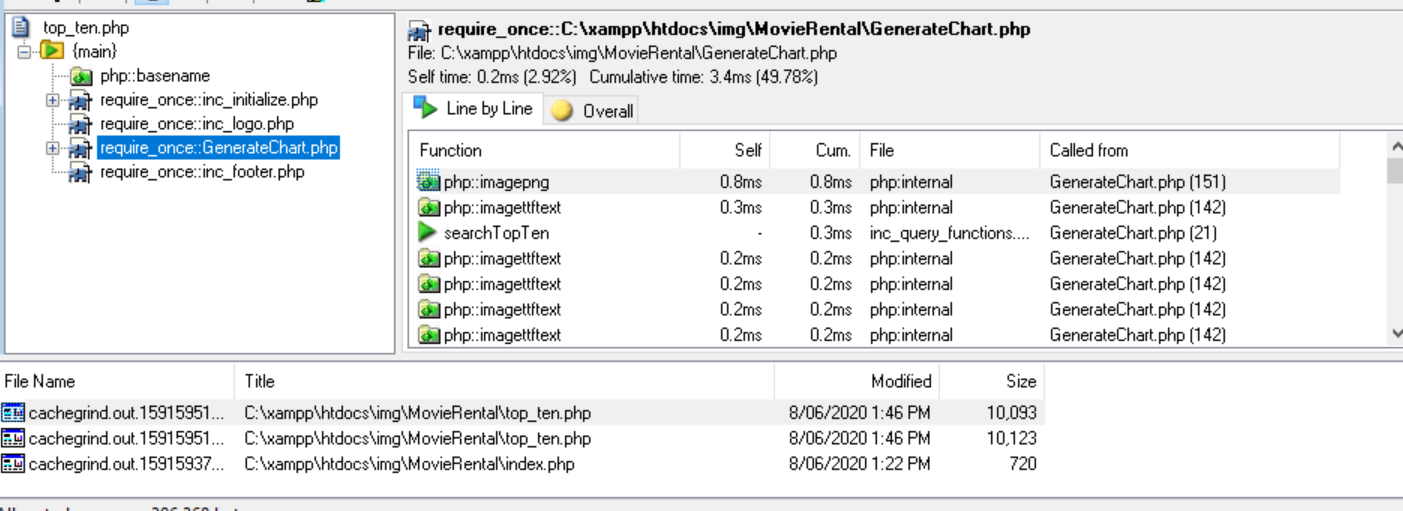
Profiling tools are used to measure the performance of a program by analysing the resources allocated to it, typically per line of code. It is a performance analysis tool which can be used to evaluate a program’s performance and potentially find the root cause of excessive resource usage.

For our project, we used Xdebug, which is an extension PHP debugger which also includes a software profiler. Its installation varies depending on the operating system, but for the machines our team used, which were all Windows, it was required to download the Windows binary corresponding to the installed version of PHP, and configuring the php.ini file to enable use of Xdebug’s profiler like so:  


Running the profiler is dependent on the configuration set in the php.ini file. We set it to trigger on pages where the extension “?XDEBUG\_PROFILE” was appended to the end of the url. The outcome of the profiling session is stored in a specified directory. This file cannot be read conventionally and requires the use of a 3rd party tool, such as WinCacheGrind for Windows machines.



The analysis indicates that the server is running as expected and there are no performance issues.



Here when the loading of the graph was profiled, we noticed no issues with performance and everything was running as expected.

### Conclusion:

The tools outlined in this report are useful and can be used to clean up code and detect performance issues. In our project, the code optimiser helped format the code so that it looks neat, presentable and organised. The profiler indicated to us that there were presently no performance issues.