## Performance Report

### Performance Tools

The performance of a website is a critical factor; it should not be overlooked in the development process. A site could look good, have all the functionality required, but have excessive load and performance times which would be detrimental to the user’s experience. A website performing poorly could be all it takes for the user to become frustrated and look for a better option. If the purpose of the site is to sell a product, this could lead to missing sales, and reduce site traffic.

Like the graphics of a website, less can be more when keeping the site at optimal performance. But the best way to ensure functionality is not lost due to performance, it is to use tools to perform testing on the speed and determine what functions perform poorly. Some tools for testing include Pingdom, GTmetrix and Google Chrome inspect element. These tools provide a simple waterfall report of the tasks and their speed.

Using Googles Chrome’s built in tools would be the most appropriate for this project as other tools need a proper URL to be input, rather than using the localhost link. The tools are easily accessible by right clicking on the page and navigating to the network and performance tabs which offer some good insight. The duration of a function is recording in milliseconds and useful charts display the data.

Figure 1: Using Google's inspect element to display a waterfall performance view

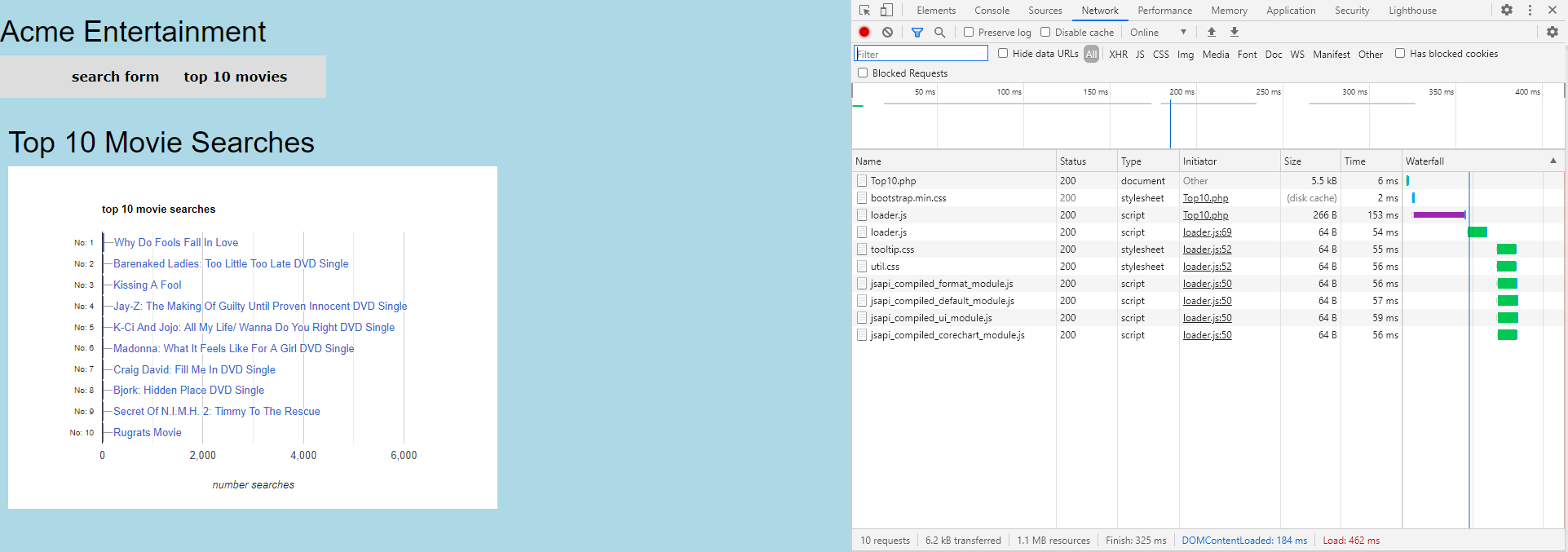
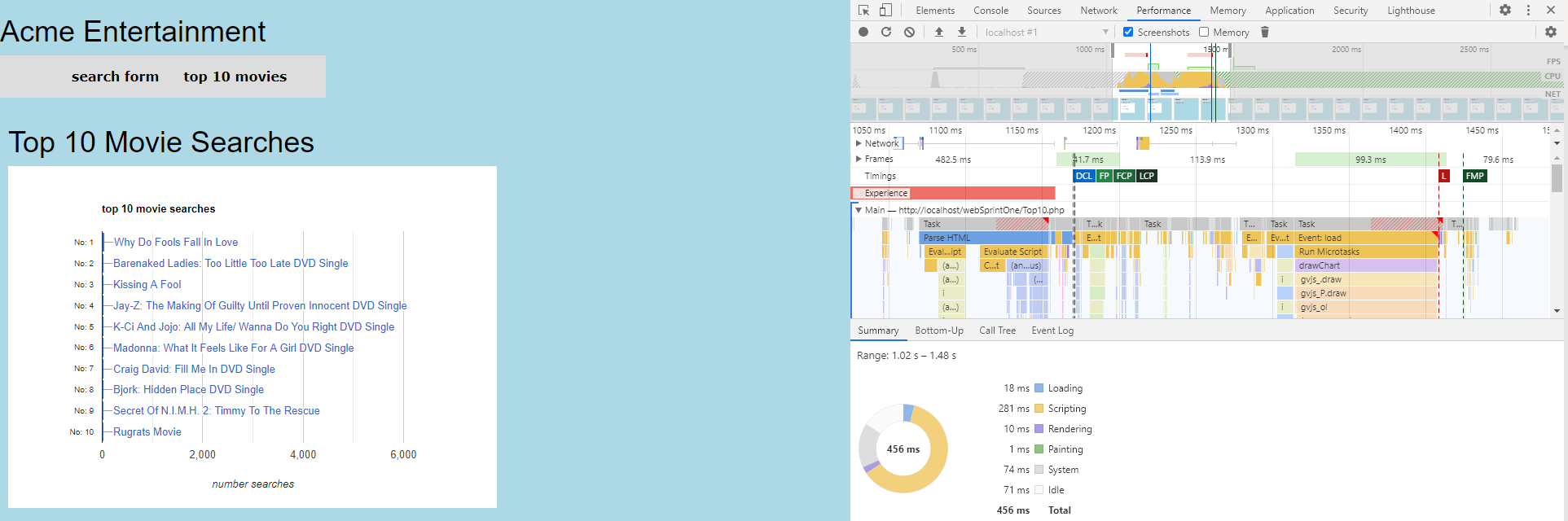


Figure 2: Using Google's inspect element lets you take a closer look at duration of each function



The tools mentioned which capture the performance are good, but there are tools to offer a more detailed insight and can suggest how to improve the performance too. Google Page Speed Insights shows a score out of 100 based on several factors that are known to cause slow load times. It also lists the suggestions of making the page faster with the estimated saving time per task. Google insights is available an extension of the Chrome browser so this tool could be used in the development of the project.

### Code Optimizers

Code optimizers check the code for common flaws such as unused variables, unnecessary object creation, incorrect white space, and lines of code with too many characters. The optimizer tools also allow code standards to be followed which can be useful for large development teams. Having different developers, with different backgrounds can result in inconsistent code which is harder to read and maintain. Running a code optimizer can save a lot of time which would be wasted by the developers trying to follow a style every step along the way. The following tools are available for PHP.

* PHP CS Fixer – Automatically fixes coding standards. Can follow defined coding standards or your teams’ style through configuration.
* PHPCD – A copy/paste detector which scans for duplicated code. The idea is that when you copy and paste code, there is the possibility for bugs to be introduced when logic is updated at one place but not another.
* PHP Mess Detector - Lets you examine code with a combination of rules. Typically used for code size rules and locating overly complicated code and bug sources.
* PHP Code Sniffer – Works by sniffing the code with a preconfigured standard or custom set of rules. Identifies missing comments, incorrect white space. Automatically fixes the errors with the code beautifier following standards such as PEAR.

The PHP Code Sniffer tool could be a good optimizer on the project. Since there are multiple developers writing code, the scripts can be run in the sniffer and a standard can be checked against. It runs in the command window and is an easy tool to work with. You can customize the standards if too many unwanted errors are being produced. The errors are customizable and can be changed to warnings or be removed if they are irrelevant.