# Optimization Report(S3)

## Description

Performance optimization measures the performance of the server-side scripts using various tool in different environments. It is performed by running the codes several times and observing the average execution time to create a benchmark.

## Need for Optimization

1. To make applications fast & effective.
2. It appeals the consumers: speedy campaigns imply higher selling speed

## Case Studies

* Bing found that searches that were 2 seconds slower resulted in a 4.3% drop in revenue per user.
* When Mozilla shaved 2.2 seconds off their landing page, Firefox downloads increased 15.4%.
* Shopzilla saw conversion rates increase 7-12% because of their web performance optimization efforts.
* Making Barack Obama’s website 60% faster increased donation conversions 14%

## Best Practices

* Using latest PHP – PHP 8.1.0 Alpha 1
* Using JSON data types instead of XML
* Using caching systems
* Closing the DB connection and limit the DB hit.

## Performance Tuning Steps Followed

To keep a check and eliminate slow loading pages, inaccessible pages, unresponsive links, and many other factors that would limit the applications downside and work fast and efficient for ACME Entertainment, below tuning tips were considered to monitor our coding performance:

1. **Bottlenecks** – Identifying impediments and finding the root cause
2. **Profiling** – What fits ACME Entertainment’s needs. Prefix was looked into as a profiling tool
3. **Code Optimization**- code was optimized so that it utilizes memory, executes more rapidly, and also performs fewer input and output operations
4. **Configure Optimization**- optimize the performance of your application, as well as ensure the reliability and cost-effectiveness of your applications system storage
5. **Distributed Computing** - Increase the potential for parallel execution
6. **Catching Strategy** - reduce the number of database operations and code compilation
7. **Load Balancing** - Load balancing is configured in the plugins using various techniques to prioritize by avoiding session objects (stored in local memory)
8. **Avoiding Client-side** - constantly check and remove or lessen the redirection of your pages
9. **Security** - responsible for processing users request in a web page or in the browser
10. **SQL** - dynamic management views (DMVs) which provides provide data about query stats, executing plans, and many more
11. **Content Delivery Network** - reduce latency, reduce bandwidth consumption, and secure apps while it can also block spammers and other that attack your system
12. **Error handling** - allows you to develop and improve error reporting that best suit the needs

## Performance Tuning Steps Followed

* Latest versions used
* Used double Quotes (“) and Single quotes (‘)
* Avoided Relative Paths in File Inclusion
* Releasing all Resources
* Avoided Unnecessary Use of Global Variables
* Never used Count or Any Other Methods in The Condition Section of a Loop
* Latest versions used
* Used double Quotes (“) and Single quotes (‘)
* Avoided Relative Paths in File Inclusion
* Releasing all Resources
* Avoided Unnecessary Use of Global Variables
* Never used Count or Any Other Methods in The Condition Section of a Loop
* Used ISSET
* Used More Static Methods/Properties
* foreach > for > while implementations

## Previous Observations

#### Performance Extracts

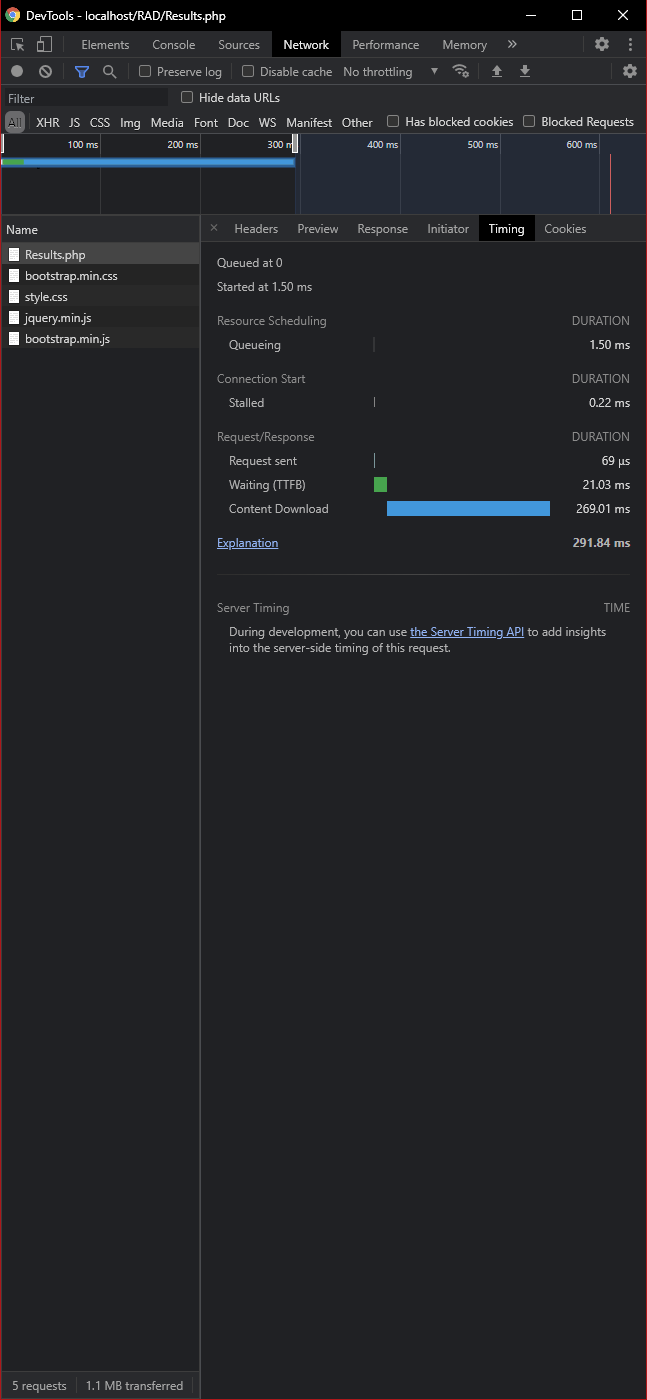
Table

Description automatically generated

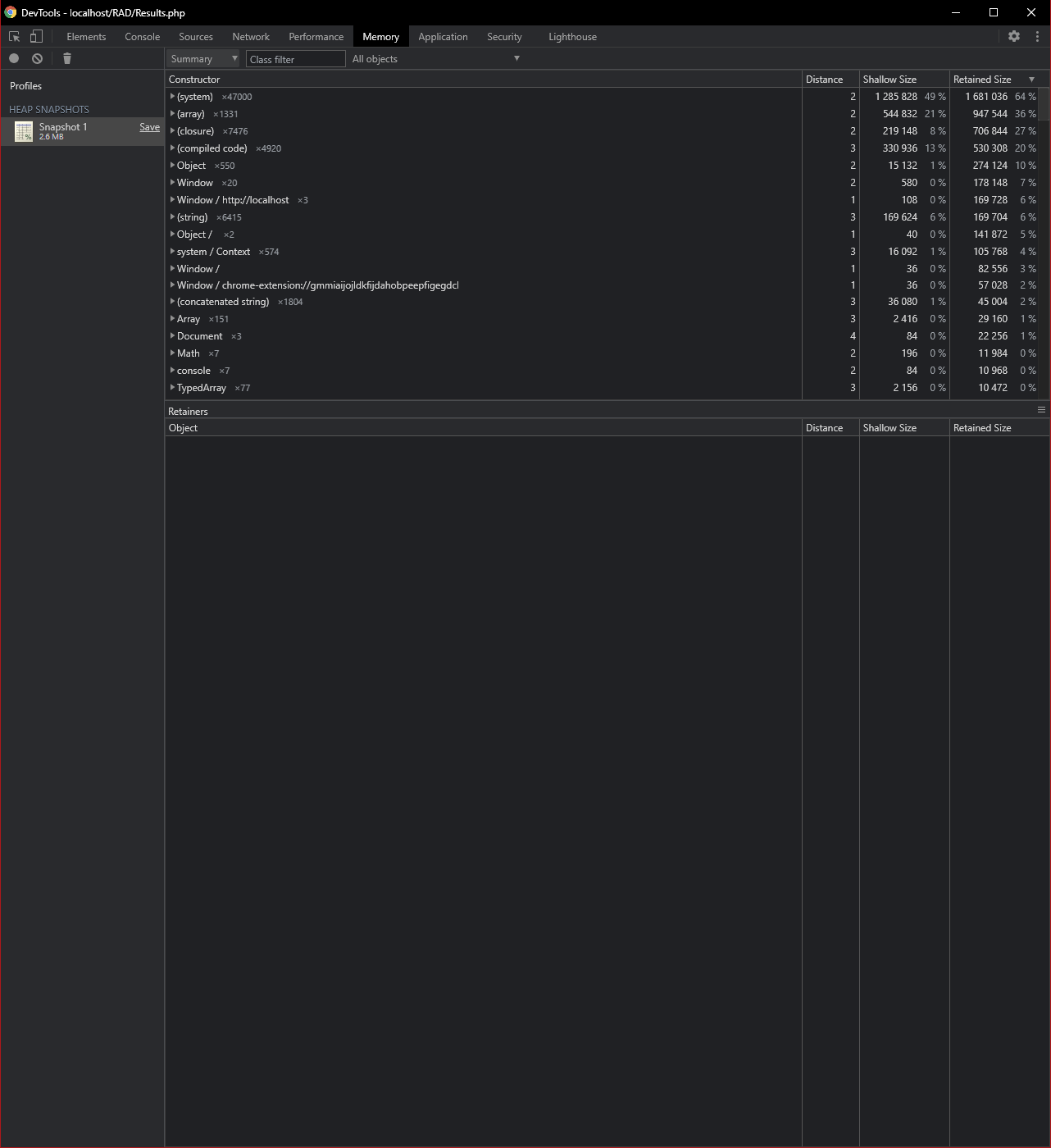
Graphical user interface, application, table

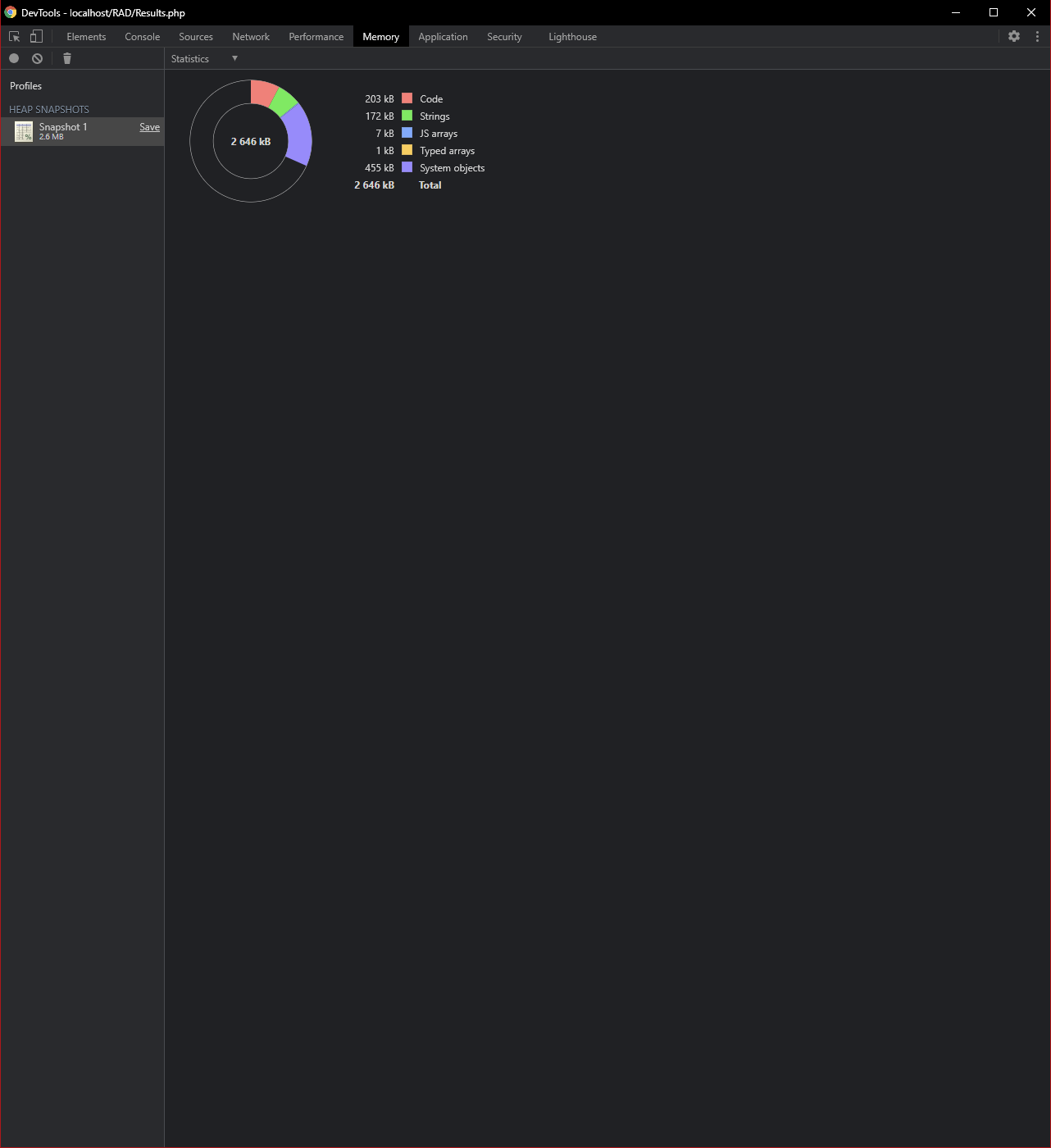
Description automatically generated

## Network Analytics

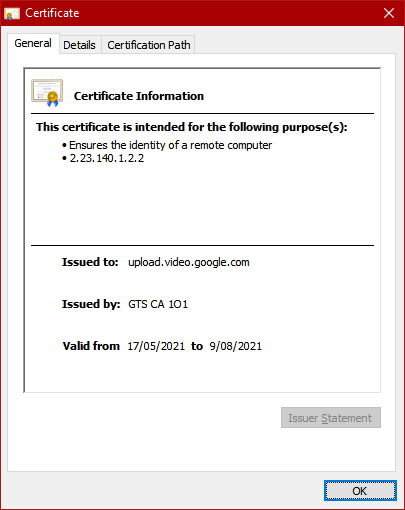


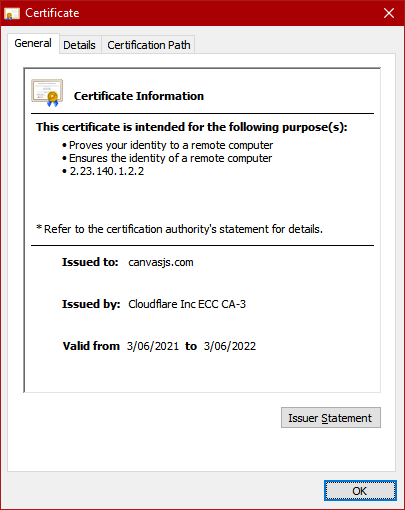
## Memory Status



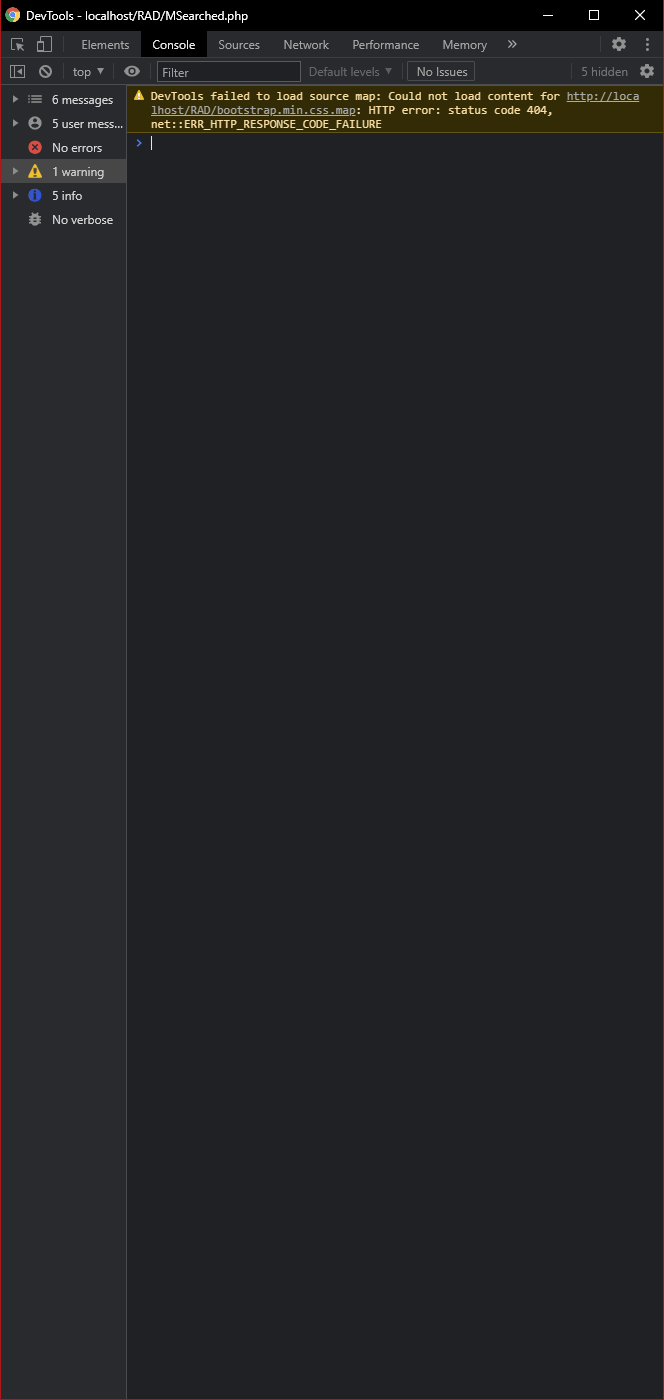


## Security





## Warnings / Error Logging



## Current Performance Extract Report

