**THE MISSPRINTS – CATHERINE BURNS, BENJAMIN ROYANS & JAI ANANDA**

DPLOMA IN SOFTWARE DEVELOPENT 2020 | SM TAFE | ROCKINGHAM

OPTIMIZATION REPORT: Sprint 3

RAPID APPLICATION DEVELOPMENT

Contents

[Definition 1](#_Toc43832151)

[Need for Optimization 1](#_Toc43832152)

[Case Studies 1](#_Toc43832153)

[Best practices 1](#_Toc43832154)

[Performance Tuning steps followed 2](#_Toc43832155)

[Performance Optimization Techniques followed 2](#_Toc43832156)

[PREVIOUS OBSERVATIONS 4](#_Toc43832157)

[Performance Extracts of Admin Page 4](#_Toc43832158)

[Network Analytics 5](#_Toc43832159)

[Memory Stats 6](#_Toc43832160)

[Security Overview 6](#_Toc43832161)

[Performance Extracts of Index Page – Initial Historical Observation 7](#_Toc43832162)

[Network Analytics 8](#_Toc43832163)

[Memory Stats 9](#_Toc43832164)

[Security 9](#_Toc43832165)

[Warnings/Error Logging 9](#_Toc43832166)

[Current performance Extracts of Index Page – Under Shared memory 10](#_Toc43832167)

[Current performance Extracts of Index Page – Under Unshared memory 14](#_Toc43832168)

OPTIMIZATION REPORT

# Definition

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 and efficient
2. It appeals the consumers: speedy campaigns implies 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% as a result of their web performance optimization efforts.
* Making Barack Obama’s website 60% faster increased donation conversions 14%. (stevesouders, 2013)

# Best practices

* + Using latest PHP – PHP 7.x versions
  + Using JSON data types instead of XML
  + Using caching systems
  + Closing the DB connection and limit the DB hits

# 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 Optimization Techniques 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
* Used ISSET
* Used More Static Methods/Properties
* foreach > for > while implementations

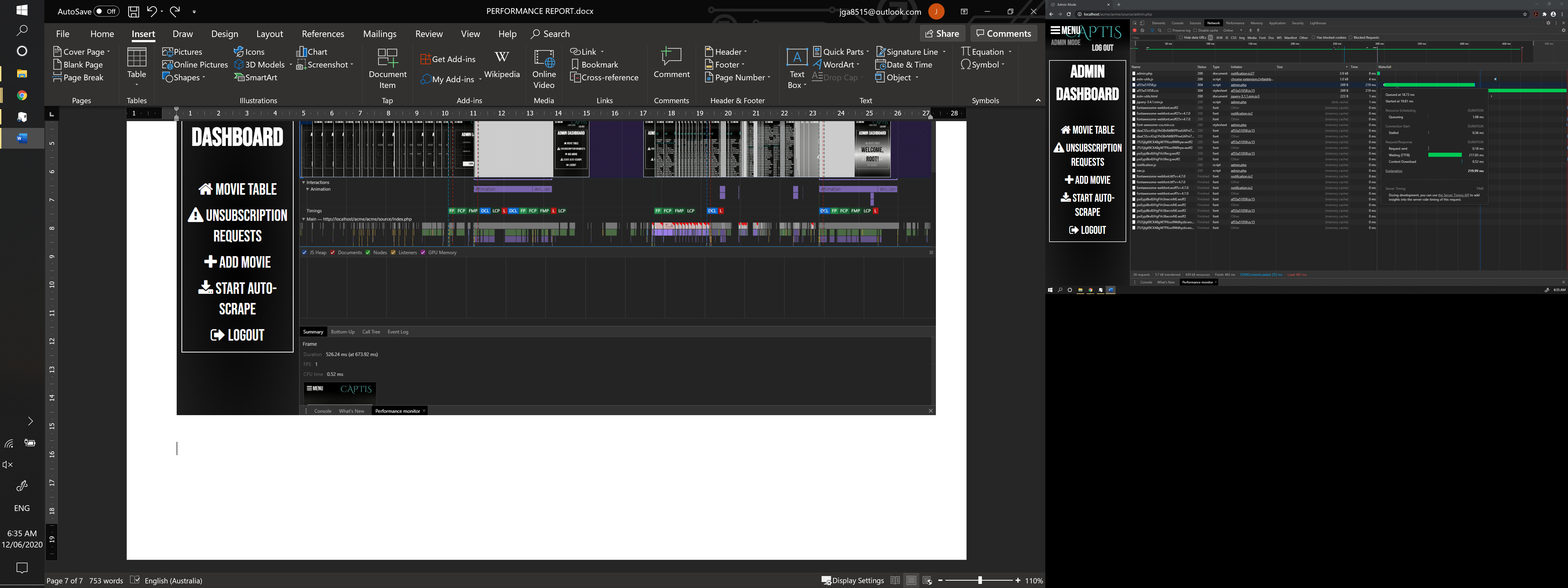
# PREVIOUS OBSERVATIONS

# Performance Extracts of Admin Page

A screen shot of a video game

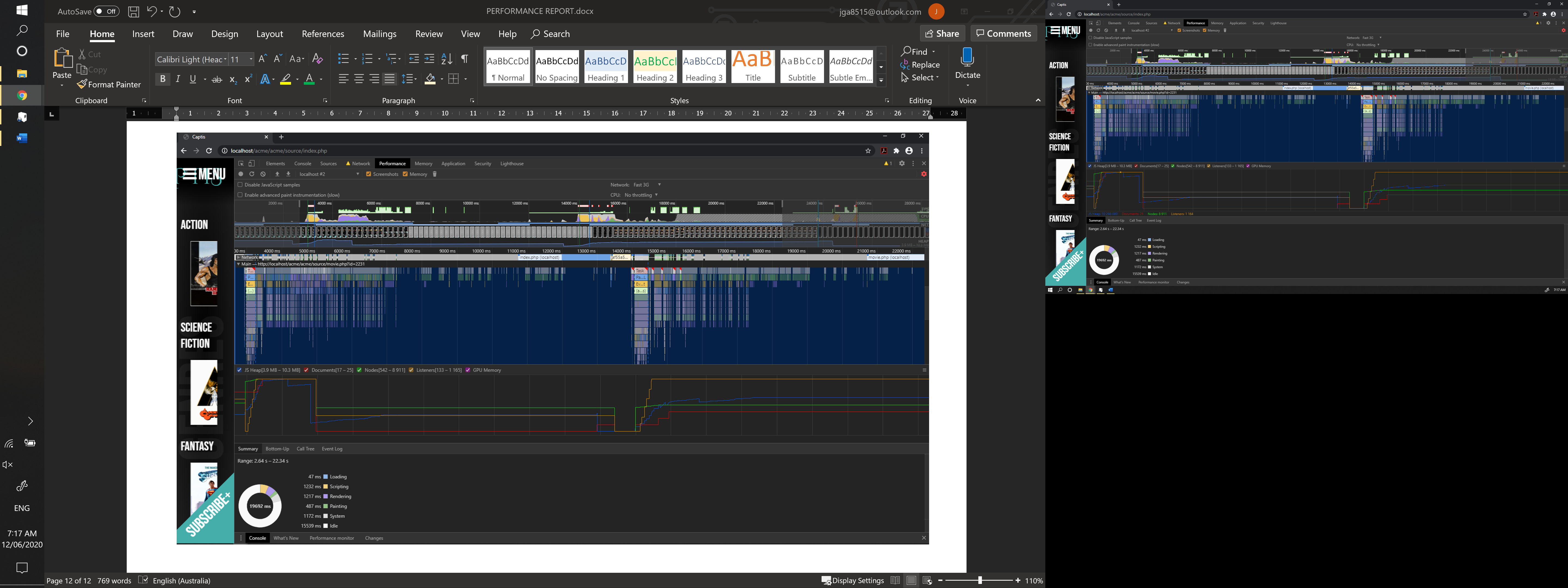
Description automatically generated

## Network Analytics



|  |  |
| --- | --- |
| Memory Stats A screenshot of a cell phone  Description automatically generated | Security Overview A screenshot of a cell phone  Description automatically generated |

# Performance Extracts of Index Page – Initial Historical Observation



## Network Analytics



|  |  |
| --- | --- |
| Memory Stats A screenshot of a cell phone  Description automatically generated | Security A screenshot of a cell phone  Description automatically generated  A screenshot of a computer  Description automatically generated |

## Warnings/Error Logging

A screenshot of a cell phone

Description automatically generated

# Current performance Extracts of Index Page – Under Shared memory

A screen shot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer screen

Description automatically generated

# Current performance Extracts of Index Page – Under Unshared memory

A screenshot of a computer screen

Description automatically generated