# E - Commerce : Project Proposal

Saurabh Yadav

Divesh Bhagat

MCS-205

February 22, 2019

Our objective is to build a spectacles (including different type of glasses) selling e-commerce website similar to existing website such as “lenskart.com”. We chose this website after thorough analysis of lenskart.com. Due to some unknown reasons, we were unable to find existing site’s web stack.

## Analysis :

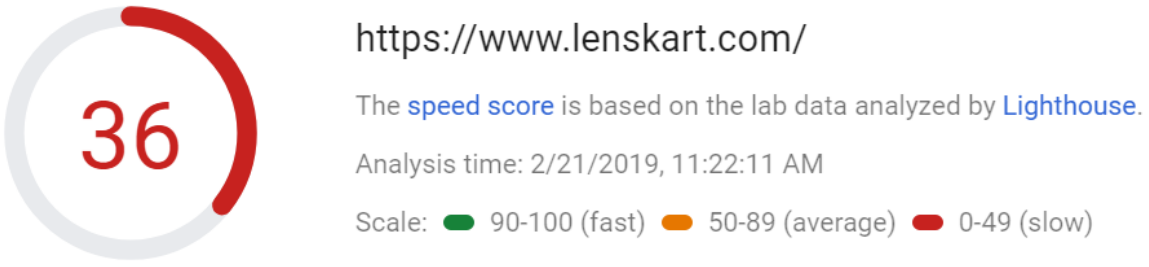
We tried to observe lenskart’s performance with the help of :

https://developers.google.com/speed/pagespeed/insights/

https://tools.pingdom.com/

https://www.similarweb.com/website/lenskart.com

Following diagram shows how well lenskart.com performed on the standard speed score test.




This score is for desktop site. Mobile version had even worse result with score of only 25.

Lenskart.com has many interactive features and images. Following table shows how much data is occupied by respective categories.

Content size by content type

| CONTENT TYPE | PERCENT | SIZE |
|---|---|---|
| Image | 68.58% | 4.6 MB |
| Script | 17.87% | 1.2 MB |
| Font | 6.36% | 424.6 KB |
| HTML | 4.03% | 269.4 KB |
| XHR | 1.72% | 115.0 KB |
| CSS | 1.12% | 74.6 KB |
| Redirect | 0.32% | 21.1 KB |
| Total | 100.00% | 6.7 MB |

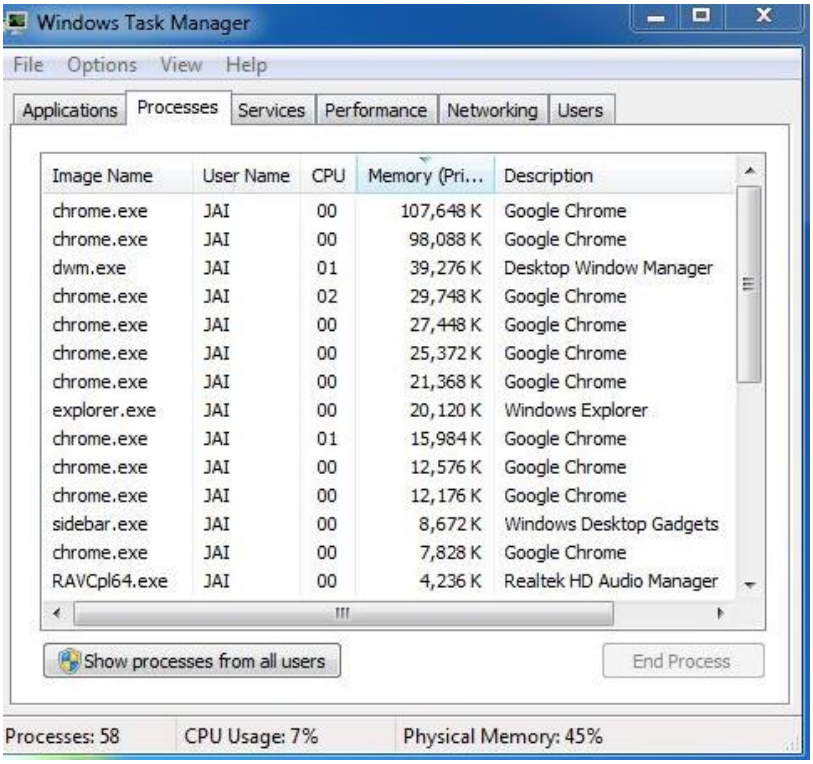
This site also seems to have longer loading time. Following diagram shows the time taken by different parts of site to load.

Lighthouse analysis of the current page on an emulated mobile network. Values are estimated and may vary.

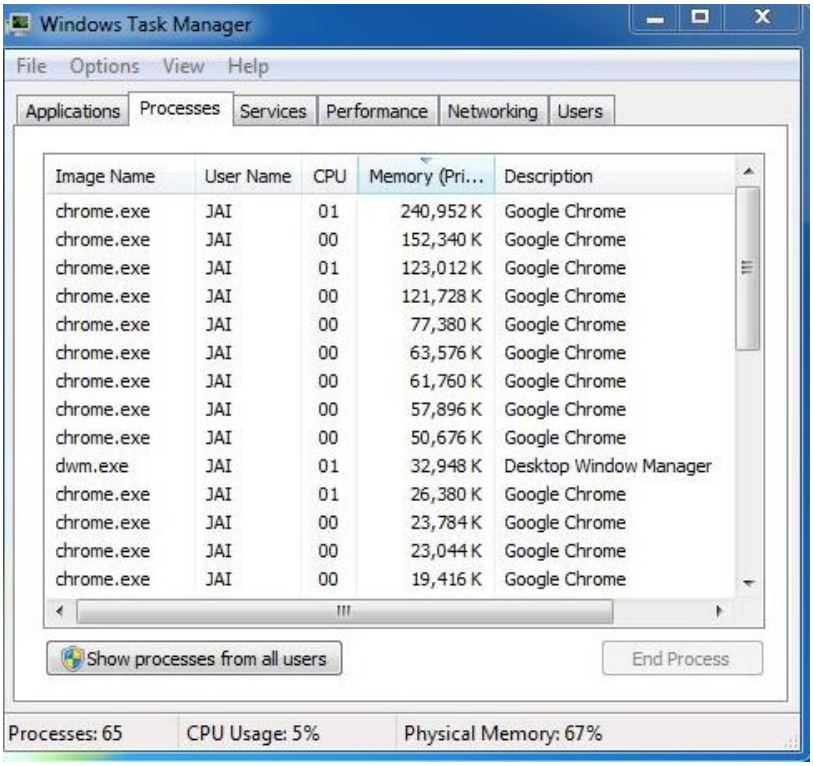
| | | | |
|---|---|---|---|
| First Contentful Paint | 1.0 s | First Meaningful Paint | 1.4 s |
| Speed Index | 4.3 s | First CPU Idle | 5.1 s |
| Time to Interactive | 6.5 s | Estimated Input Latency | 130 ms |

This data is for desktop version of site which we are more focused on.

Now here are some test results for primary memory consumption when using lenskart.com



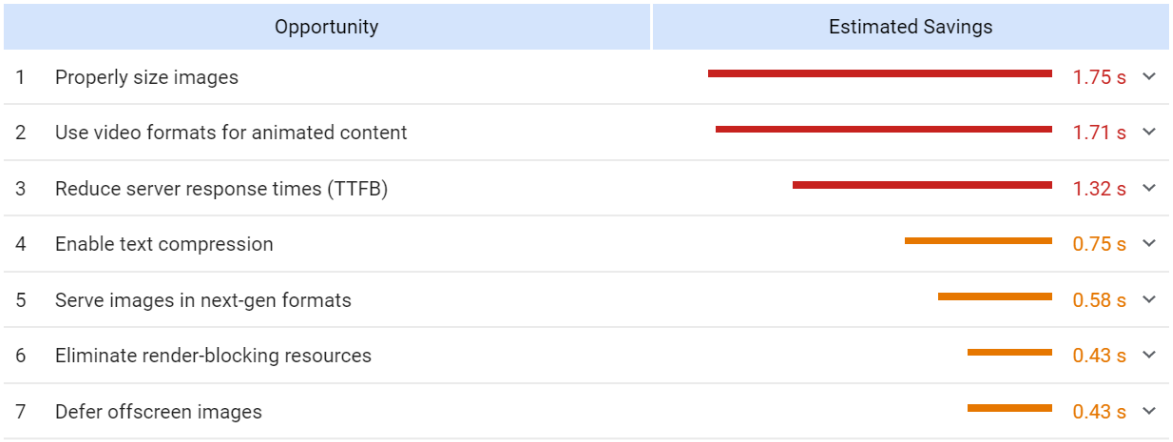

Before using lenskart.com in Chrome




After opening 2 tabs of lenskart.com in Chrome

## Changes proposed :

Based on above observations, we propose to have different optimization techniques to improve performance of lenskart.com that we will be building under this project.



| | Opportunity | Estimated Savings |
|---|---|---|
| 1 | Properly size images | 1.75 s |
| 2 | Use video formats for animated content | 1.71 s |
| 3 | Reduce server response times (TTFB) | 1.32 s |
| 4 | Enable text compression | 0.75 s |
| 5 | Serve images in next-gen formats | 0.58 s |
| 6 | Eliminate render-blocking resources | 0.43 s |
| 7 | Defer offscreen images | 0.43 s |

These all changes are proposed on the basis of what has been observed through different tests. Our goal is to optimize this result as much as possible.

## Frameworks and Technologies

- MongoDB : For maintaining database as it is schema-less database, easily scalable, support of sharding and it is good for maintaining e-commerce product catalog. Since, it is a NOSQL database, then it is obviously secure because no sql injection can be made.
- Django Framework : Helps in developing web application faster, scalable. Support for features like user authentication, content administration, site maps and many more. It security features helps in preventing common secure mistakes like SQL injection, cross-site request forgery, clickjacking and cross-site scripting.