

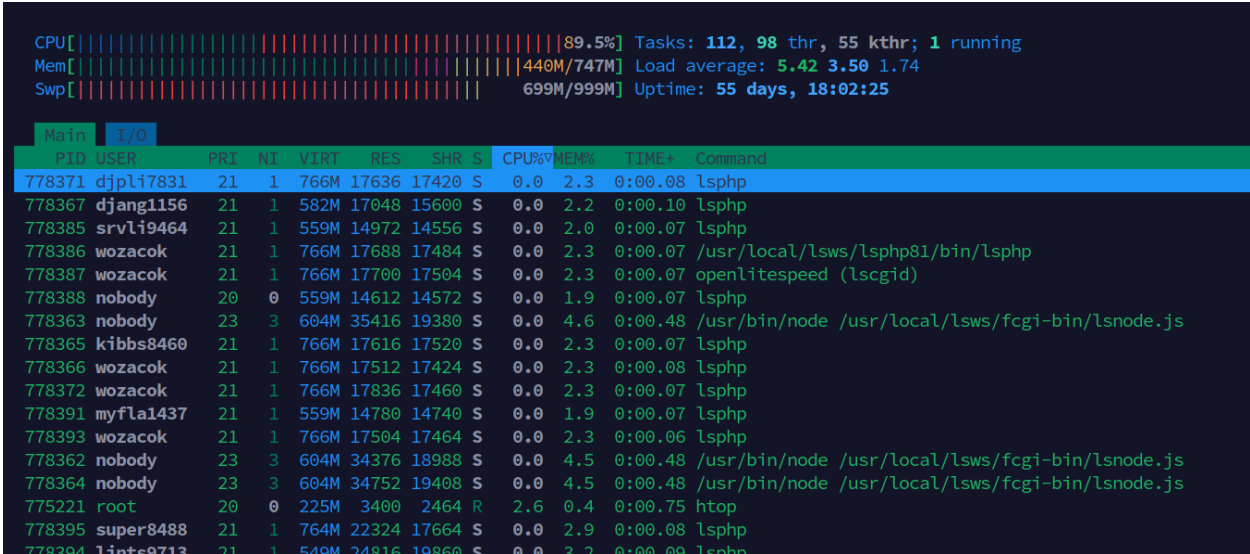
# Module 4: Website Performance Troubleshooting

Website performance directly impacts user experience and search engine rankings. This module explores strategies to identify and resolve performance bottlenecks.

## Identifying Performance Bottlenecks

### 1. Server-Side Issues

- High CPU or memory usage.
- Disk I/O bottlenecks affecting read/write speeds.
- Tools: Use the cPanel Resource Usage tab to monitor resource usage.
- Use the top or htop command to check server RAM, CPU and Load Spikes.



```
CPU[|||||||||||||||||||||||||||||||||||||||||89.5%] Tasks: 112, 98 thr, 55 kthr; 1 running
Mem[|||||||||||||||||||||||||||||||||||||440M/747M] Load average: 5.42 3.50 1.74
Swp[|||||||||||||||||||||||||||||||||||||699M/999M] Uptime: 55 days, 18:02:25

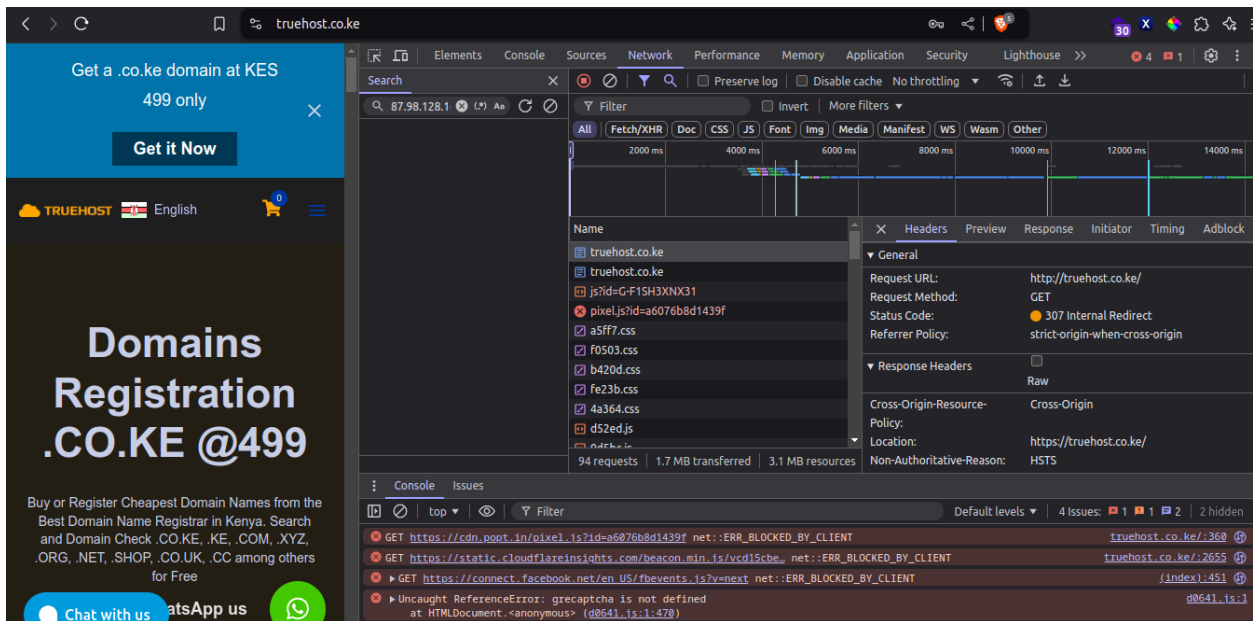
Main I/O
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
778371 djpli17831 21 1 766M 17636 17420 S 0.0 2.3 0:00.08 lsphp
778367 djang1156 21 1 582M 17048 15600 S 0.0 2.2 0:00.10 lsphp
778385 srqli9464 21 1 559M 14972 14556 S 0.0 2.0 0:00.07 lsphp
778386 wozacok 21 1 766M 17688 17484 S 0.0 2.3 0:00.07 /usr/local/lsws/lsphp81/bin/lsphp
778387 wozacok 21 1 766M 17700 17504 S 0.0 2.3 0:00.07 openlitespeed (lscgid)
778388 nobody 20 0 559M 14612 14572 S 0.0 1.9 0:00.07 lsphp
778363 nobody 23 3 604M 35416 19380 S 0.0 4.6 0:00.48 /usr/bin/node /usr/local/lsws/fcgi-bin/lsnode.js
778365 kibbs8460 21 1 766M 17616 17520 S 0.0 2.3 0:00.07 lsphp
778366 wozacok 21 1 766M 17512 17424 S 0.0 2.3 0:00.08 lsphp
778372 wozacok 21 1 766M 17836 17460 S 0.0 2.3 0:00.07 lsphp
778391 myfla1437 21 1 559M 14780 14740 S 0.0 1.9 0:00.07 lsphp
778393 wozacok 21 1 766M 17504 17464 S 0.0 2.3 0:00.06 lsphp
778362 nobody 23 3 604M 34376 18988 S 0.0 4.5 0:00.48 /usr/bin/node /usr/local/lsws/fcgi-bin/lsnode.js
778364 nobody 23 3 604M 34752 19408 S 0.0 4.5 0:00.48 /usr/bin/node /usr/local/lsws/fcgi-bin/lsnode.js
775221 root 20 0 225M 3400 2464 R 2.6 0.4 0:00.75 htop
778395 super8488 21 1 764M 22324 17664 S 0.0 2.9 0:00.08 lsphp
778394 lint9713 21 1 549M 24816 19860 S 0.0 3.2 0:00.09 lsphp
```

### 2. Network Issues

- High latency or bandwidth constraints.
- Tools: Use **traceroute** and **ping** to identify connectivity issues.

### 3. Client-Side Issues

- Browser caching inefficiencies.
- Excessive or poorly optimized JavaScript.
- Tools: Browser developer tools to track network requests and resource loading.

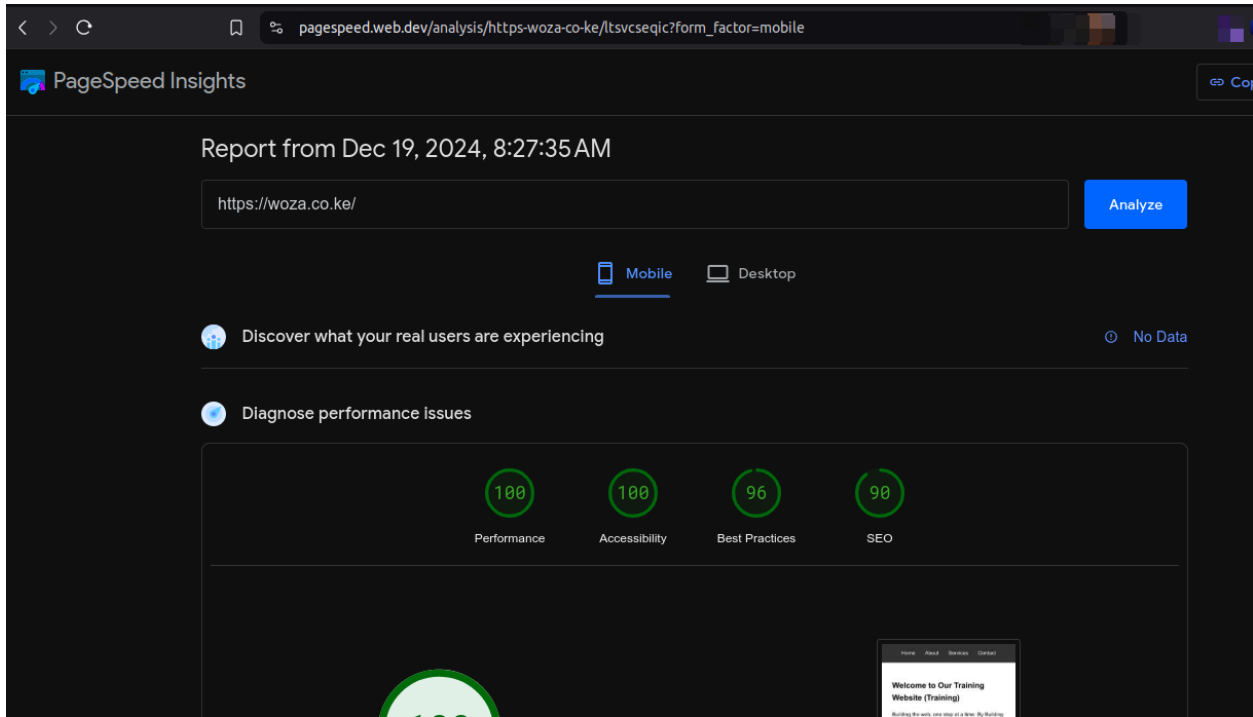


---

## Website Speed Testing Tools

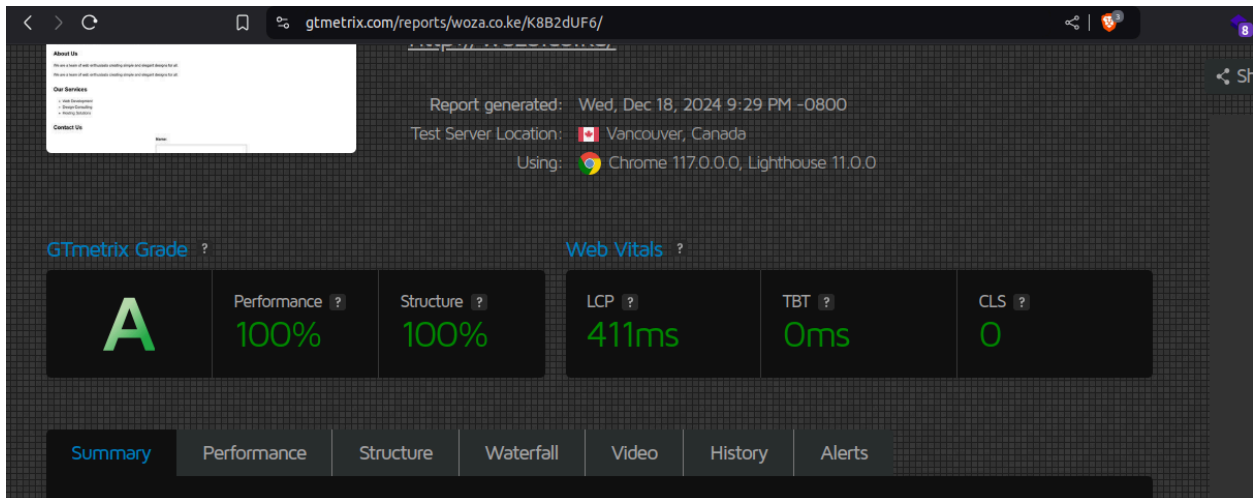
### 1. Google PageSpeed Insights

- Provides actionable insights to improve speed and user experience.
- <https://pagespeed.web.dev/>
- For a website that is well optimized, you should have a report similar to the one shown below.
- At least get a 90% score.



## 2. GTmetrix

- Comprehensive performance analysis with waterfall charts.
- <https://gtmetrix.com/>
- For a website that is well optimized, you should have a report similar to the one shown below.
- At least get an A score.



## 3. Pingdom Website Speed Test

- Helps identify load times for individual resources.
- <https://tools.pingdom.com/>
- For a website that is well optimized, you should have a report similar to the one shown below.
- Performance Grade should be 100 (A)

The screenshot shows the Pingdom website performance test interface. At the top, there is a search bar with the URL `http://woza.co.ke/` and a dropdown menu set to "Europe - Germany - Frankfurt". A green "START TEST" button is visible. Below this, a banner reads "The internet is fragile. Be the first to know when your site is in danger." with a "START YOUR FREE TRIAL" button and a computer icon with a red exclamation mark. The main results section, titled "Your Results:", contains a large white square placeholder on the left and four performance metrics on the right: Performance grade (A 100), Page size (3.1 KB), Load time (479 ms), and Requests (2). At the bottom left, there is a link "Improve page performance".

Metric	Value
Performance grade	A 100
Page size	3.1 KB
Load time	479 ms
Requests	2

---

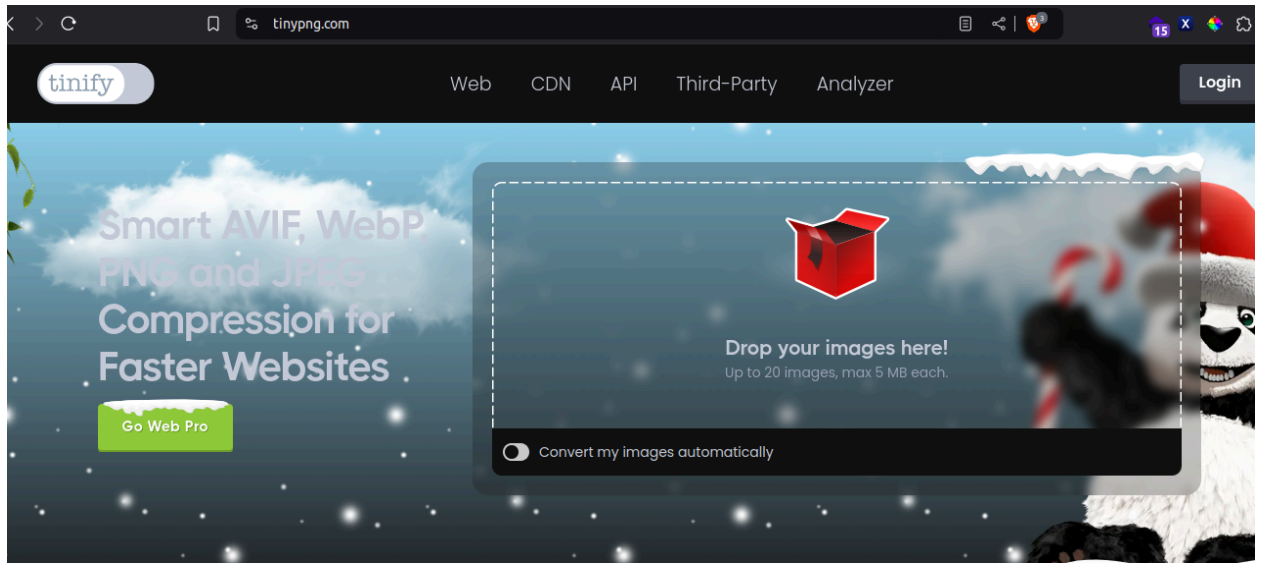
## Optimizing Website Performance: Recommendations

### 1. Caching

- Enable browser caching to store static assets locally.
- Use server-side caching mechanisms like Varnish (on VPS Servers) or LiteSpeed Cache( On servers running Openlitespeed or Litespeed Web Servers)

### 2. Image Optimization

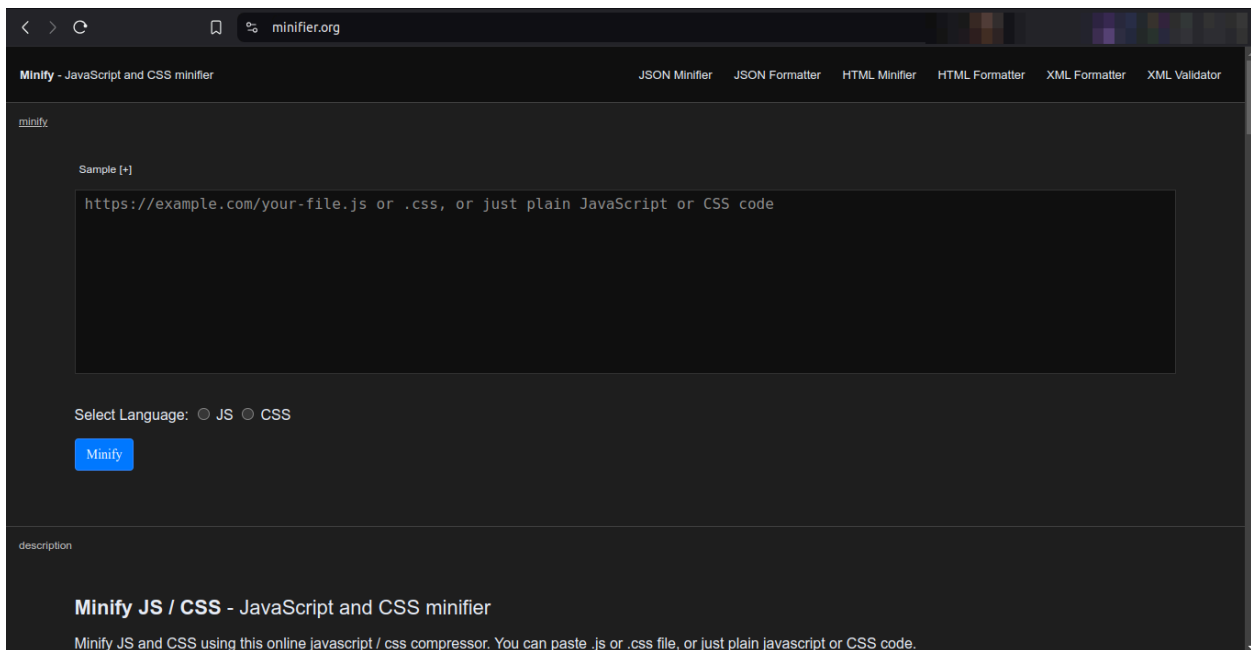
- Compress images using tools like TinyPNG



- Serve images in modern formats like WebP.

### 3. Code Minification and Concatenation

- Remove unnecessary whitespace and comments in CSS, JavaScript, and HTML.
- Combine multiple CSS or JavaScript files to reduce HTTP requests.
- You can use a tool such as <https://www.minifier.org/>



## 4. Database Optimization

- Regularly clean up unused data and optimize tables.
- Implement efficient indexing and query optimization.
- On the terminal, you can use the command “**mysqlcheck**” to optimize mysql databases.

**mysqlcheck -u root -p --optimize --all-databases**

```
[root@srv ~]# mysqlcheck -u root -p --optimize --all-databases
Enter password:
B2hbi8GY1uIxEm.wp_commentmeta
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK
B2hbi8GY1uIxEm.wp_comments
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK
B2hbi8GY1uIxEm.wp_links
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK
B2hbi8GY1uIxEm.wp_litespeed_url
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK
B2hbi8GY1uIxEm.wp_litespeed_url_file
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK
B2hbi8GY1uIxEm.wp_options
note      : Table does not support optimize, doing recreate + analyze instead
status    : OK
B2hbi8GY1uIxEm.wp_postmeta
note      : Table does not support optimize, doing recreate + analyze instead
```

---