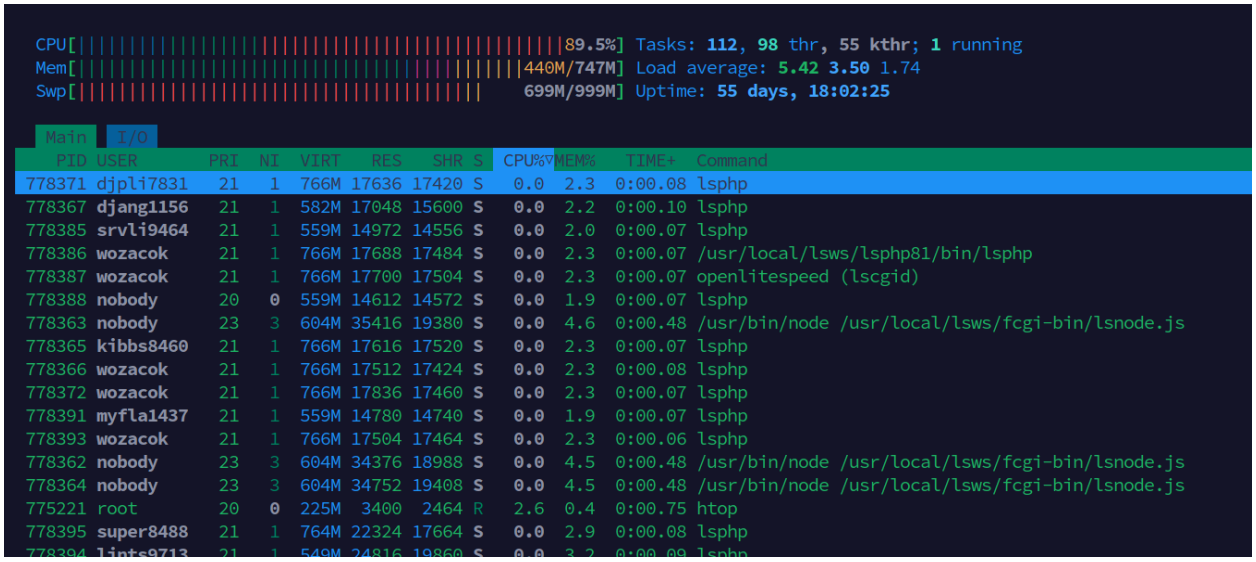
Module 4: Website Performance Troubleshooting

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.

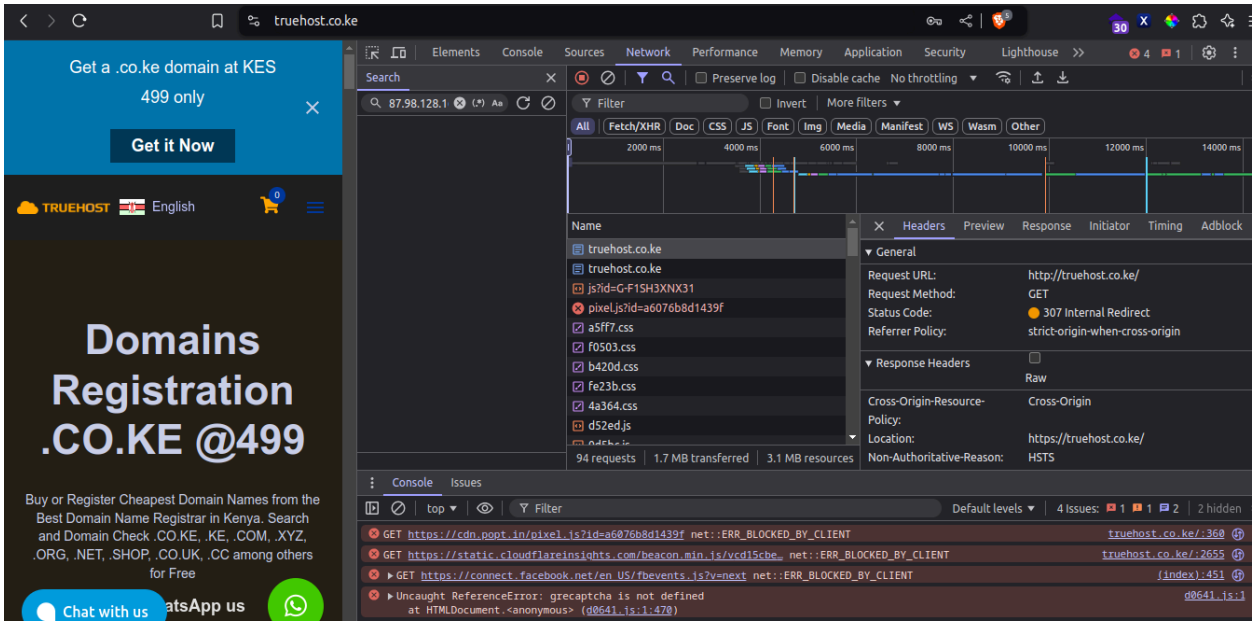


2. Network Issues

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

3. Client-Side Issues

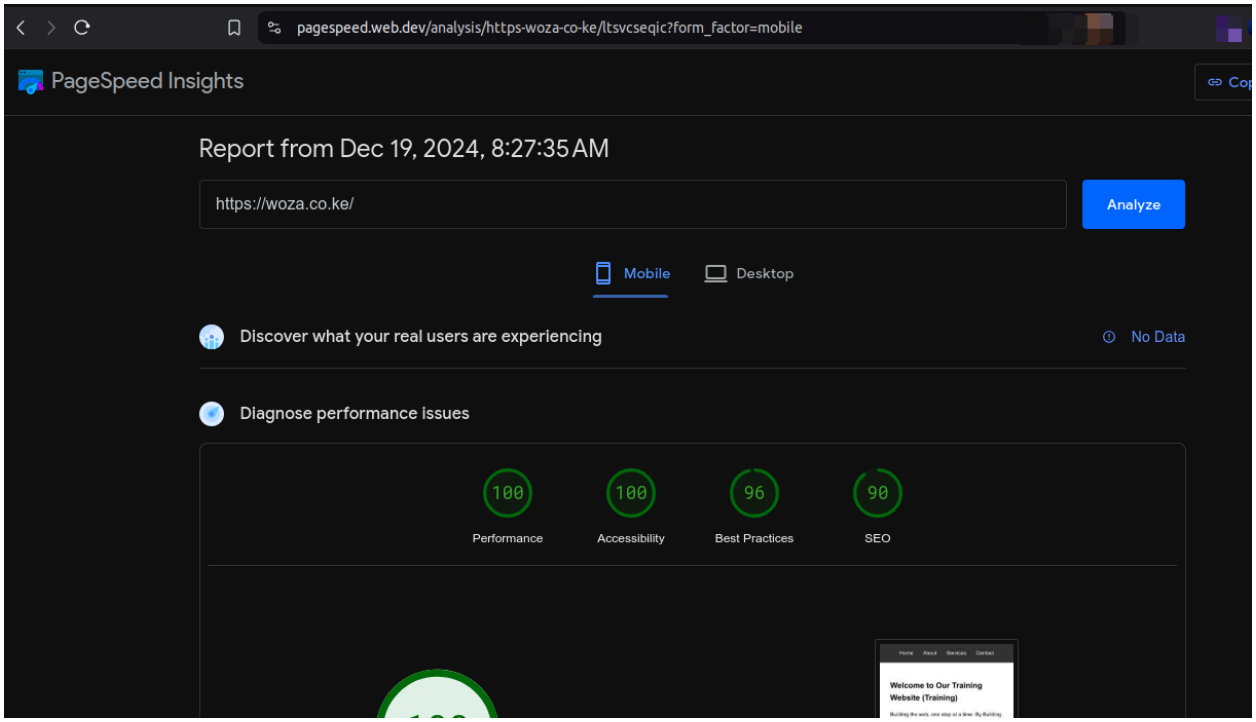
- Browser caching inefficiencies.
- Excessive or poorly optimized JavaScript.
- Tools: Browser developer tools to track network requests and resource loading.



Website Speed Testing Tools

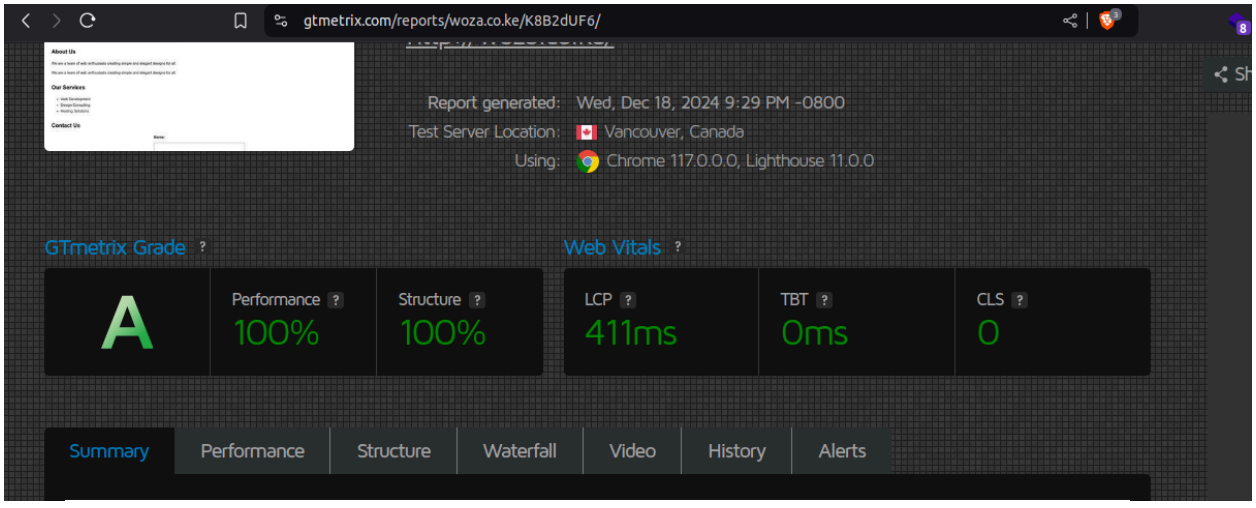
1. Google PageSpeed Insights

- 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.



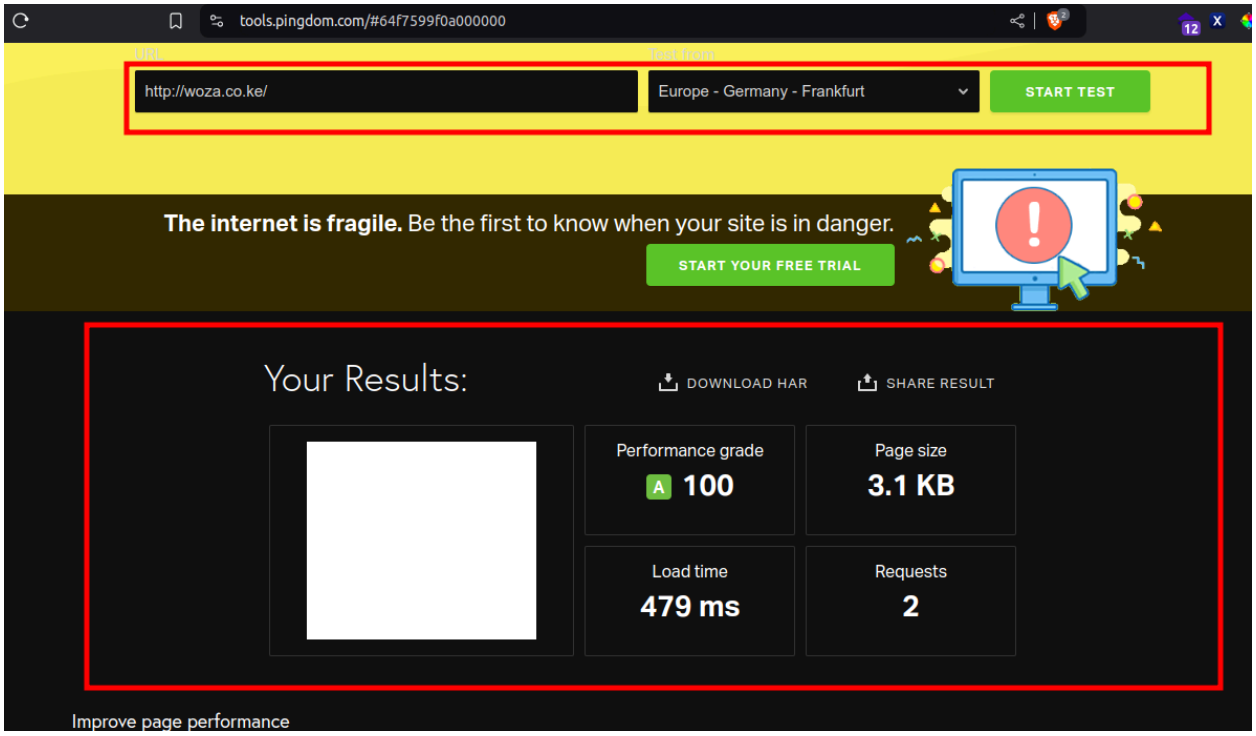
2. GTmetrix

- 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)



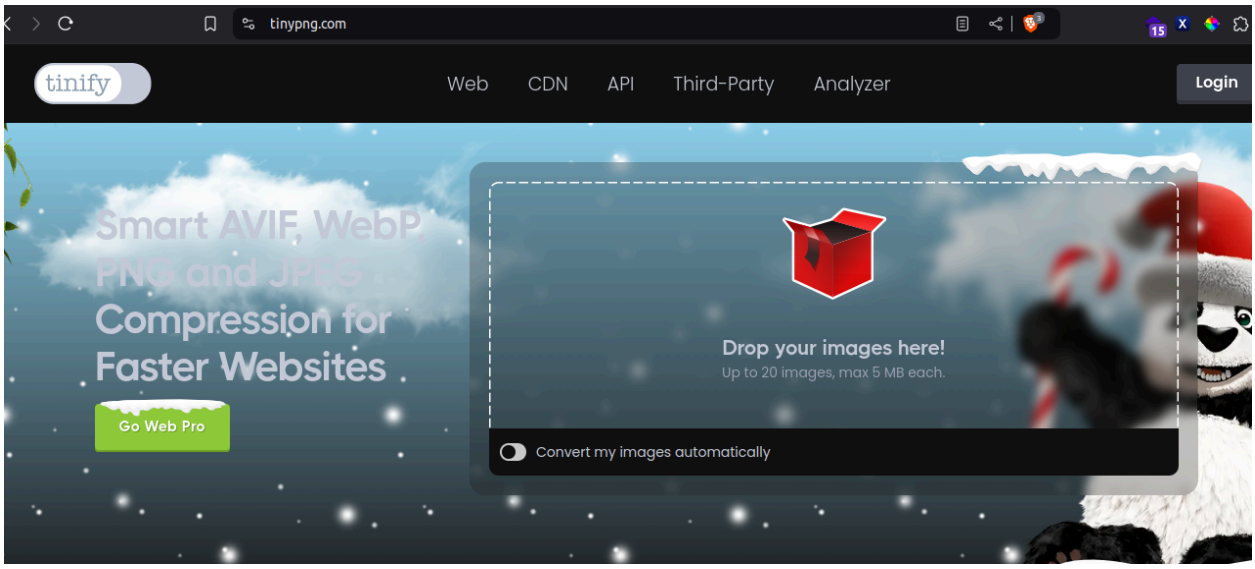
Optimizing Website Performance: Recommendations

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

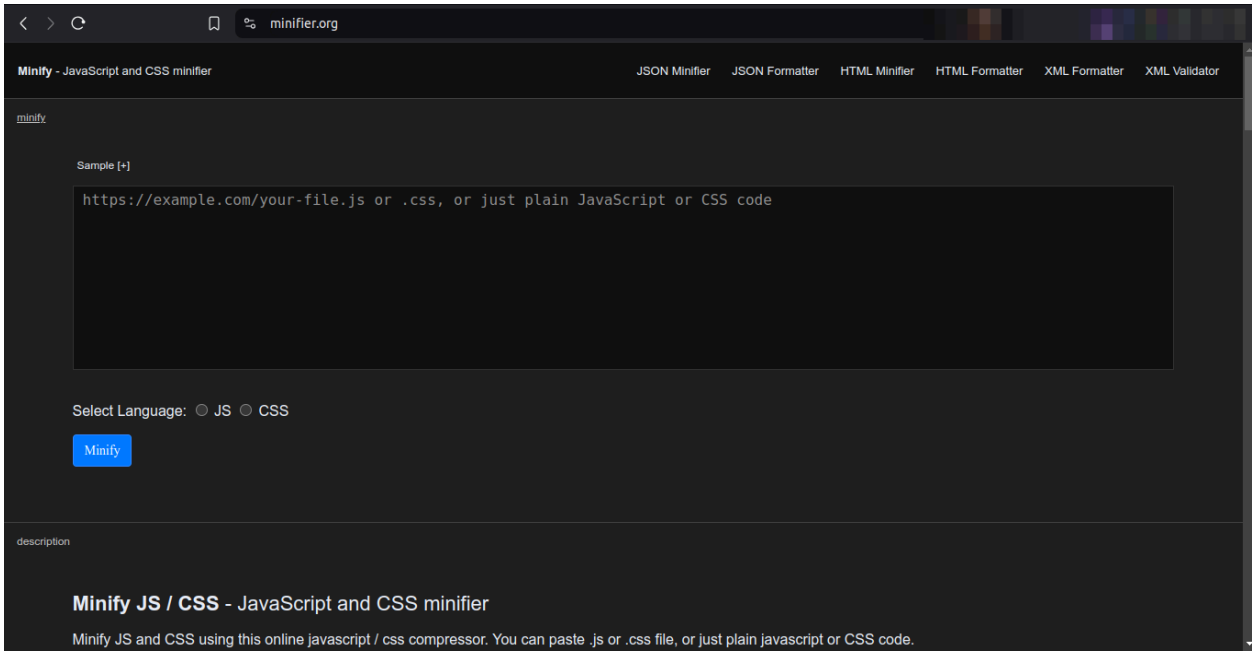
- Compress images using tools like TinyPNG



- 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
```