Task 2: Performance Optimization

This report includes:-

1. Before Optimization

2. Website Optimization

3. After Optimization

(1) Before Optimization:

GTmetrix Desktop Performance Metrics :

GTmetrix Desktop Performance Metrics for https://www.hostinger.com/in before optimization.
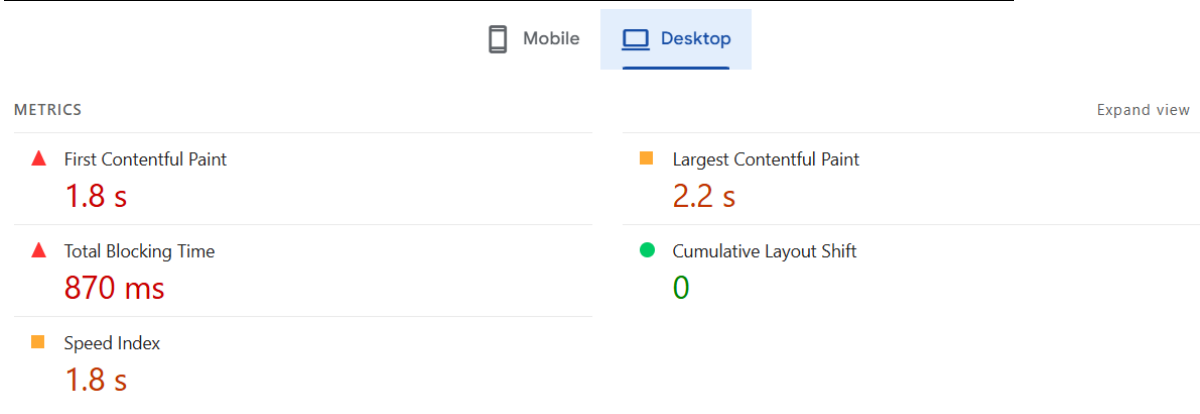
Key metrics included:

- Page Load Time: 3.8s

- Total Page Size: 2.1 MB

- Number of Requests: 48

- Performance Score: 72/100

- First Contentful Paint (FCP): 1.8s

- Largest Contentful Paint (LCP): 2.2s

- Total Blocking Time (TBT): 870ms

- Cumulative Layout Shift (CLS): 0

- Time to Interactive (TTI): 2.8s

This screenshot represents the baseline desktop performance of the website prior to applying optimization steps.
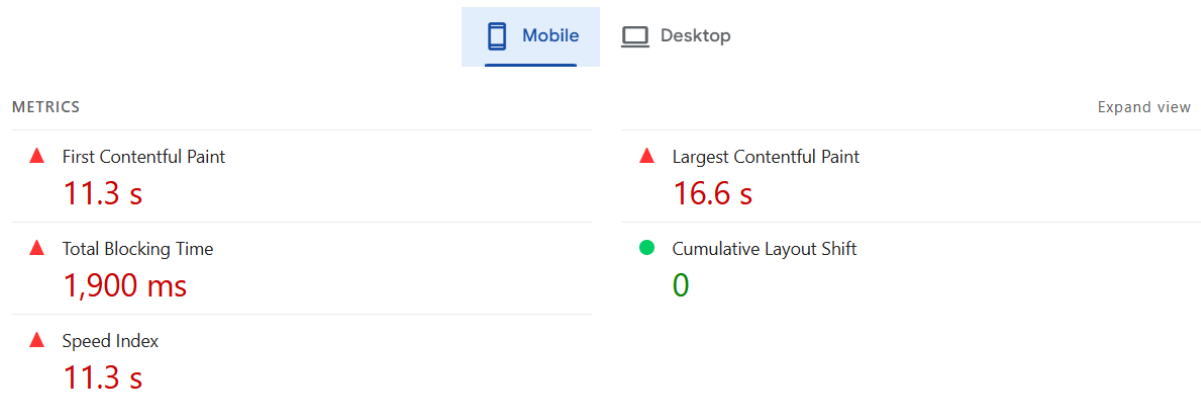
PageSpeed Insights Desktop Performance Metrics :

Mobile		Desktop

METRICS   Expand view

⚠ First Contentful Paint	⬛ Largest Contentful Paint
1.8 s	2.2 s
⚠ Total Blocking Time	⬤ Cumulative Layout Shift
870 ms	0
⬛ Speed Index	
1.8 s	

PageSpeed Insights Desktop Performance Metrics for "https://www.hostinger.com/in" before optimization.

Key metrics included:

- First Contentful Paint (FCP): 1.8s

- Largest Contentful Paint (LCP): 2.2s

- Total Blocking Time (TBT): 870ms

- Cumulative Layout Shift (CLS): 0

- Speed Index: 1.8s

This screenshot represents the baseline desktop performance of the website prior to applying optimization steps.

# PageSpeed Insights Mobile Performance Metrics :

📱 Mobile   🖥 Desktop

METRICS   Expand view

▲ First Contentful Paint   ▲ Largest Contentful Paint
11.3 s   **16.6 s**

▲ Total Blocking Time   ● Cumulative Layout Shift
1,900 ms   **0**

▲ Speed Index
11.3 s

PageSpeed Insights Mobile Performance Metrics for "https://www.hostinger.com/in" before optimization.

Key metrics included:

- First Contentful Paint (FCP): 11.3s

- Largest Contentful Paint (LCP): 16.6s

- Total Blocking Time (TBT): 1,900ms

- Cumulative Layout Shift (CLS): 0

- Speed Index: 11.3s

This screenshot represents the baseline mobile performance of the website prior to applying optimization steps.

(2) Website Optimization:

Images were downloaded from the Hostinger marketing page and compressed using TinyPNG (tinypng.com).

The purpose of compression is to reduce image file size while maintaining acceptable visual quality.

This helps improve page load speed and performance.

1. Image Compression

- The image was compressed using TinyPNG, reducing its size from 144 KB to 118 KB.
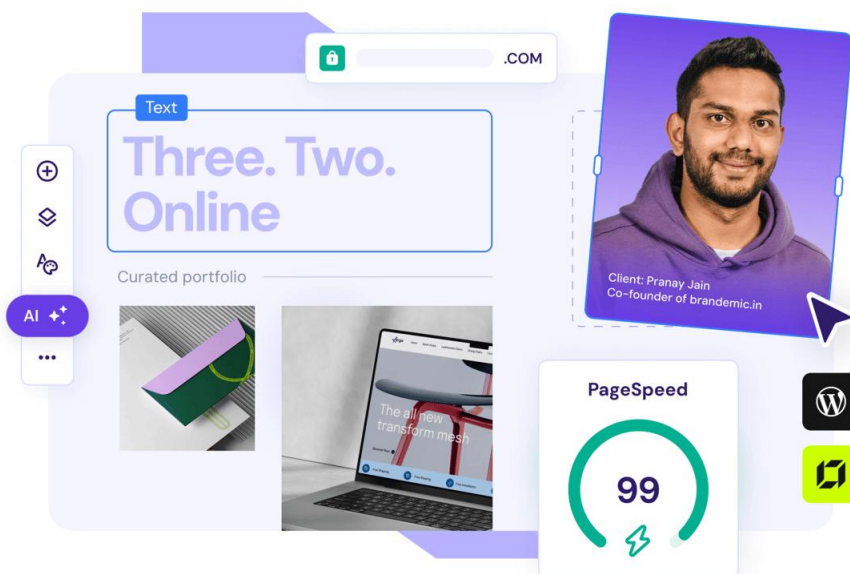
- The image was compressed using TinyPNG, reducing its size from 89 KB to 77 KB.
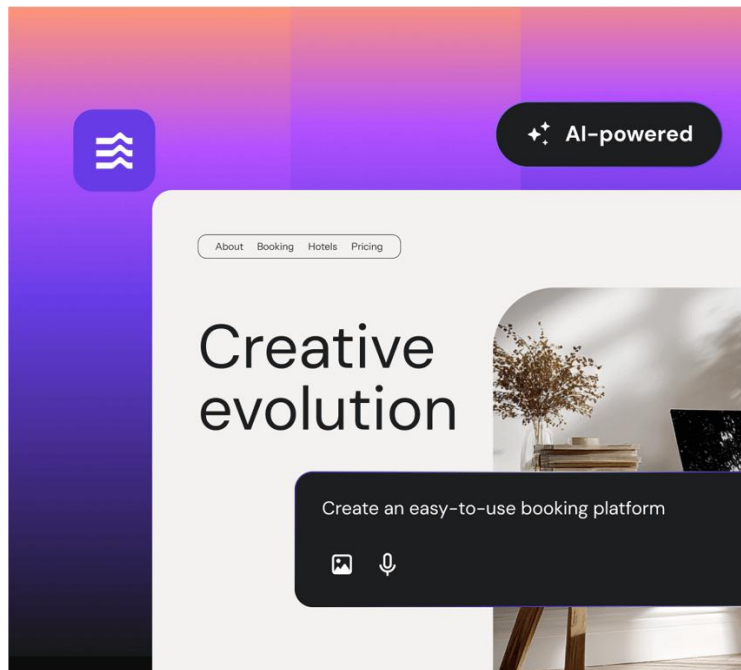
- The image was compressed using TinyPNG, reducing its size from 377 KB to 87 KB.
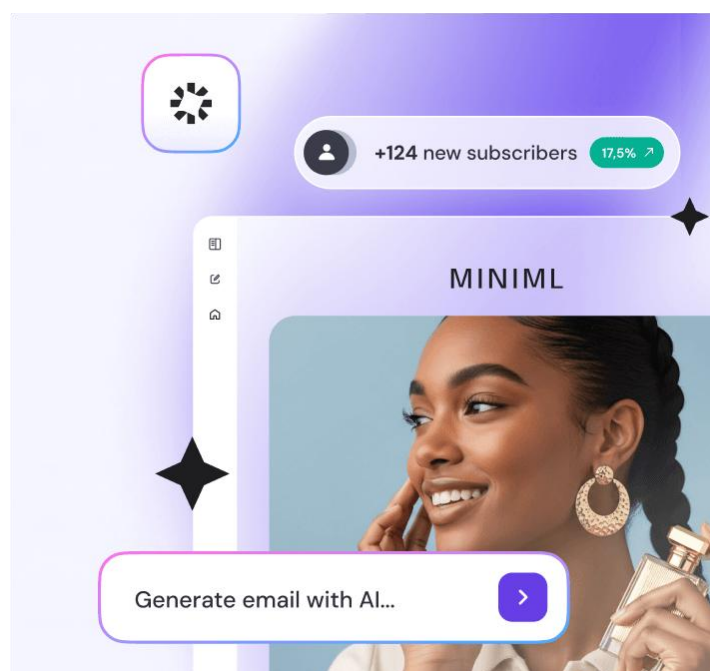
The image was compressed using TinyPNG, reducing its size from 53 KB to 42 KB.
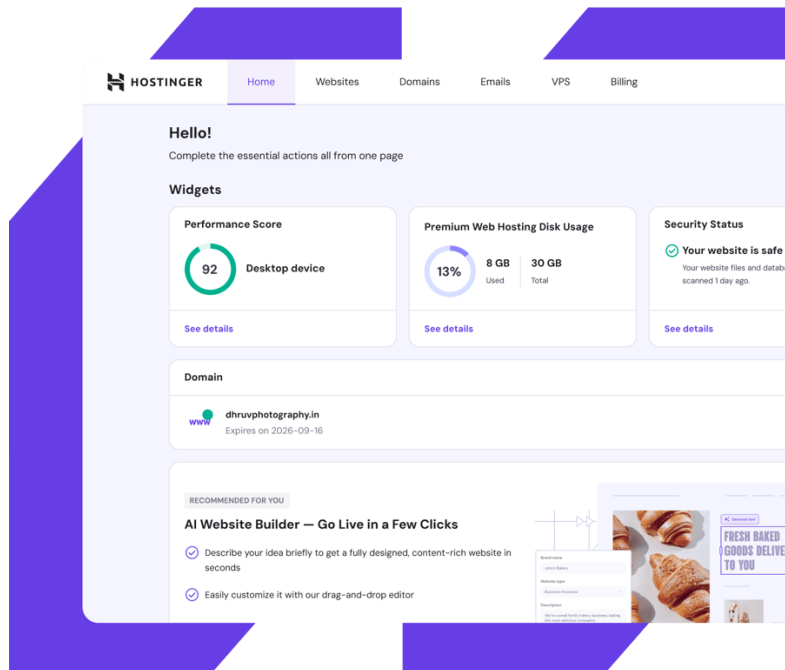
The image was compressed using TinyPNG, reducing its size from 130 KB to 88 KB.
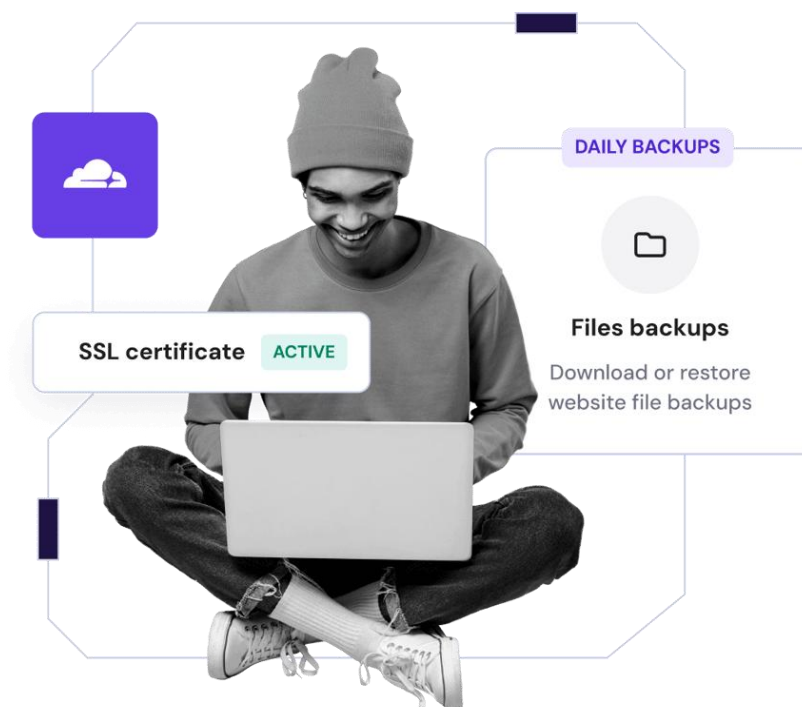
The image was compressed using TinyPNG, reducing its size from 44 KB to 34 KB.
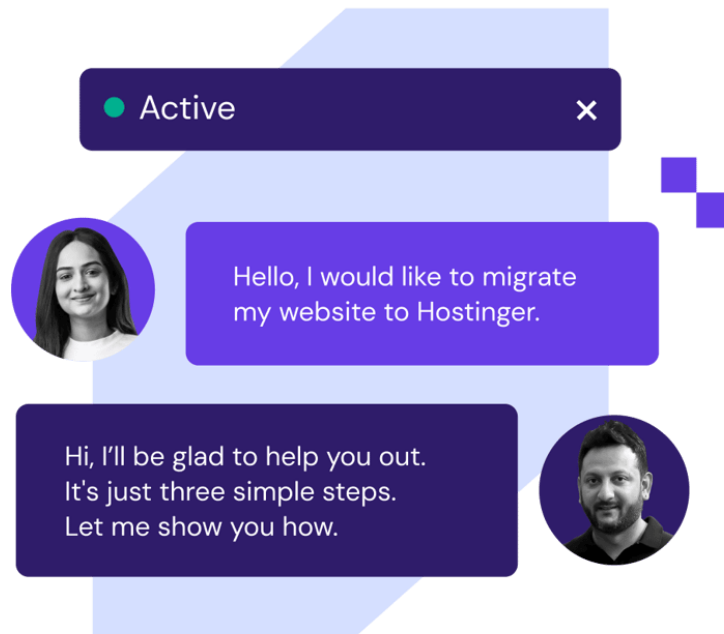
Before vs After Table :

Image No.	Before Size	After size
Image 1	114 KB	118 KB
Image 2	89 KB	77 KB
Image 3	377KB	87KB
Image 4	53 KB	42 KB
Image 5	130 KB	88 KB
Image 6	44 KB	34 KB

2. HTML/ CSS/ JS Minify:

- **HTML/CSS Minification :**
  Description:
  The outerHTML of the Hostinger marketing page was copied and passed through an online HTML minifier tool. Since the original HTML was already minified, no significant change in file size was observed.

  Table:

File Name	Before Size	After Size	Observation
Index.html	734 KB	734 KB	Already Minified

  Before Minification :

  After Minification :

- **JavaScript Minification:**

Description :

Both main.js and main2.js were already compressed and minified by the website.

Attempts to further minify using online tools resulted in syntax errors, such as:

"Unexpected token: punc «{», expected: punc «;» (line: 1, col: 4946)"

This happens because minifiers cannot process already minified or packed JS containing ES6+ syntax and complex structures.

Therefore, these files are considered already optimized.

Table :

File Name	Before Size	After Size	Observation
main.js	553 KB	-	Already minified / Could not minify
Main2.js	114 KB	-	Already minified / Could not minify

Before Minification :

main.js file :

main2.js file:

After Minification :
main.js file :

main2.js file :

JS **JSCompress** - The JavaScript Compression Tool

☰ More Developer Tools

Copy & Paste JavaScript Code   Upload Javascript Files   Output

☐ ECMAScript 2025 (via babel-minify)

Unexpected token: punc «{», expected: punc «;» (line: 197, col: 33)

```
!function i(s, a, o) {
    function n(t, e) {
        if (!a[t]) {
            if (!s[t]) {
                var r = "function" == typeof require && require;
                if (!e && r)
                    return r(t, !0);
                if (l)
                    return l(t, !0);
                throw (r = new Error("Cannot find module '" + t + "'")).code = "MODULE_NOT_FOUND",
                r
            }
            r = a[t] = {
                exports: {}
            },
            s[t][0].call(r.exports, function(e) {
                return n(s[t][1][e] || e)
            }, r, r.exports, i, s, a, o)
        }
    }
```

Compress JavaScript

About JSCompress

JSCompress is an online JavaScript compressor that allows you to compress and minify all of your JS files by up to 80% of their original size. Copy and paste your code or you can upload and combine multiple files and then compress. We use UglifyJS 3 and babel-minify for all JavaScript minification and compression.

Why You Should Compress JavaScript

There are a number of reasons why compressing your JavaScript files is a good idea:

3. After Optimization
Summary of Optimized Resources :

File Name	Before Size	After Size	Observation
Index.html	734 KB	734 KB	Already Minified
main.js	553 KB	-	Already minified / Could not minify
main2.js	114 KB	-	Already minified / Could not minify
Image 1	114 KB	118 KB	Compressed via TinyPNG
Image 2	89 KB	77 KB	Compressed via TinyPNG
Image 3	377KB	87KB	Compressed via TinyPNG
Image 4	53 KB	42 KB	Compressed via TinyPNG
Image 5	130 KB	88 KB	Compressed via TinyPNG
Image 6	44 KB	34 KB	Compressed via TinyPNG

Note: Only local file optimizations were applied. The live website was not modified; hence, "after" performance metrics are estimated.

Observed / Expected Improvements :
- Compressed images → lower page weight → faster loading speed
- Minified JS/HTML → reduced parsing time (where applicable)
- Live testing not performed; expected improvements are theoretical based on applied optimizations.

Conclusion :

- Applied optimizations are expected to improve overall website performance, particularly for mobile users.
- Image compression and minification reduce resource size while maintaining functionality and visual quality.