


Platform **WebPageTest**

URL: <https://hospital-shortener.onrender.com/>

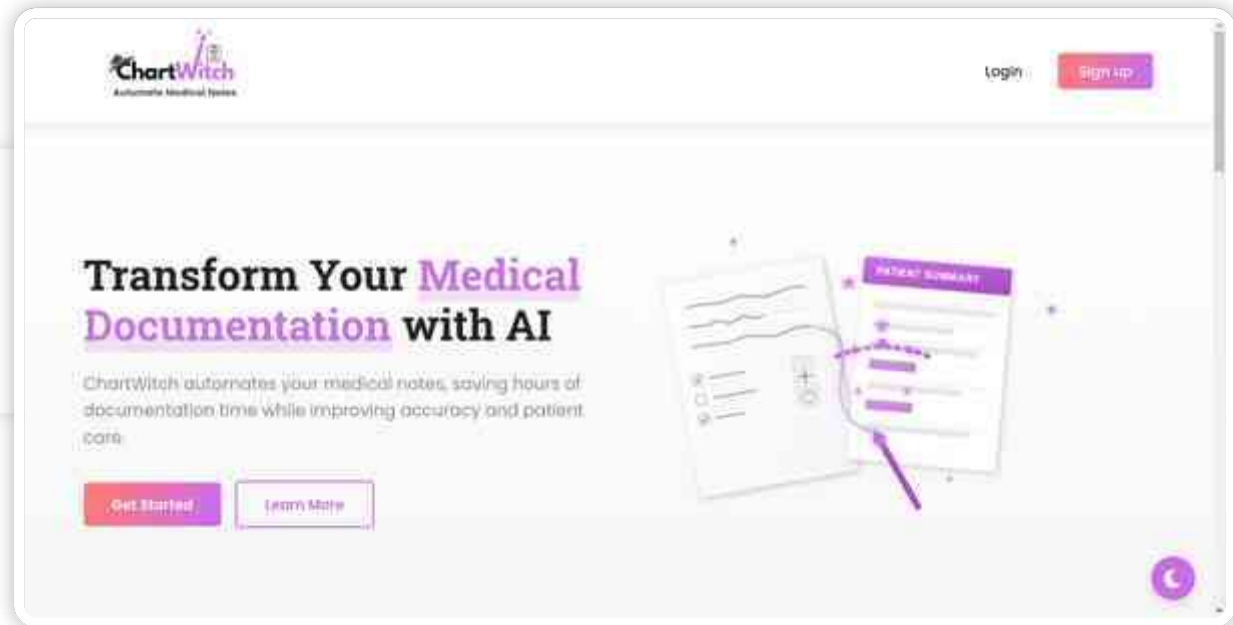
DATE: 3/30/2025, 11:52:51 AM

## Webpage Performance Test Result

SETTINGS: DESKTOP  v132  Cable  Dulles, Virginia USA [More](#) [Share](#)

View: **Opportunities & Experiments**

Tools: [Export](#) [Re-Run Test](#)



# Opportunities & Experiments NEW

WebPageTest helps identify opportunities to improve a site's experience. Select one or more No-Code Experiments below and submit to test their impact.

## Jump to Opportunities by Category:

### Is it Quick?

Not bad...

7 7 Pro 12

### Is it Usable?

Needs Improvement.

3 3 Pro 3

### Is it Resilient?

Not bad...

2 2 Pro 8

## Is it Quick?

**Not bad...** This site was quick to connect and deliver initial code. It began rendering content very quickly. There were 3 render-blocking requests. The largest content rendered quickly.

WebPageTest ran 13 diagnostic checks related to this category and found 7 opportunities.

### ▼ This test had a reasonably quick first-byte time.

A fast time to first byte is essential for delivering assets quickly.

## ▼ 1 JavaScript file is blocking page rendering.

By default, references to external JavaScript files will block the page from rendering while they are fetched and executed. Often, these files can be loaded in a different manner, freeing up the page to visually render sooner.

- /static/landing.js

### Relevant Experiments

#### Defer Render-Blocking Scripts

This experiment adds a `defer` attribute to render-blocking scripts, causing the browser to fetch them in parallel while showing the page. Deferred scripts still execute in the order they are defined in source. Example implementation: `<script src="/static/landing.js" defer></script>`

##### ▼ Assets included in experiment:

- /static/landing.js

☐ **Run This Experiment!**

#### Async Render-Blocking Scripts

This experiment adds an `async` attribute to render-blocking scripts, causing the browser to fetch them in parallel while showing the page. Async scripts are not guaranteed to execute in the order they are defined in source. Example implementation: `<script src="/static/landing.js" async></script>`

##### ▼ Assets included in experiment:

- /static/landing.js

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

### Inline Render-Blocking Scripts

This experiment embeds the contents of specified external scripts directly into the HTML within a `script` element. This increases the size of the HTML, but can often allow page to display sooner by avoiding server round trips. Example implementation:

```
<script>/* contents from /static/landing.js here...*/</script>
```

#### ► Assets included in experiment:

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## ▼ 2 externally-referenced CSS files are blocking page rendering.

By default, references to external CSS files will block the page from rendering while they are fetched and executed. Sometimes these files should block rendering, but can be inlined to avoid additional round-trips while the page is waiting to render. Sometimes, such as with stylesheets that are only used for loading custom fonts, inline or async CSS can greatly improve perceived performance.

- /static/landing.css
- /ajax/libs/font-awesome/6.0.0/css/all.min.css

### Relevant Experiments

#### Inline external CSS

This experiment embeds the contents of specified external stylesheets directly into the HTML within a `style` element. This increases the size of the HTML, but can often allow page to display sooner by avoiding server round trips. **Note:** The inline experiment may break relative references to images and other assets in the CSS. These would need proper path changes in production.

▼ **Assets included in experiment:**

- ☒ /static/landing.css
- ☒ /ajax/libs/font-awesome/6.0.0/css/all.min.css

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

### Load CSS Asynchronously

This experiment loads specified stylesheets in a way that allows the page to begin rendering while they are still loading. Note that this may cause a flash of unstyled content.

▼ **Assets included in experiment:**

- ☒ /static/landing.css
- ☒ /ajax/libs/font-awesome/6.0.0/css/all.min.css

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

▼ **Largest Contentful Paint time was under 2.5 seconds**

Great job. If LCP was higher, WebPageTest would look for ways to speed it up.

▼ **Zero render-critical images are lazy-loaded.**

When images are lazy-loaded using `loading="lazy"`, they will be requested after the layout is established, which is too late for images in the critical window.

## ▼ Images outside the critical viewport can be lazy-loaded.

When images are lazy-loaded using `loading="lazy"`, when they scroll into the viewport, freeing up early load for other tasks.

- `/static/logo-dark.svg`

### Relevant Experiments

#### Add `loading="lazy"` to images

This experiment adds `loading="lazy"` attributes to images that are outside the viewport at load.

##### ► Assets included in experiment:

**Get WebPageTest Pro**  
FOR UNLIMITED EXPERIMENTS.

## ▼ Several fonts are loaded with settings that hide text while they are loading.

When fonts are loaded with default display settings, like `font-display="block"`, browsers will hide text entirely for several seconds instead of showing text with a fallback font.

- Font Awesome 6 Free 900 normal

### Relevant Experiments

#### Add `font-display: swap`

This experiment adds `font-display="swap"` to custom fonts to show text sooner.

## ▼ 4 fonts are hosted on 3rd-party hosts

Fonts on 3rd party domains may take longer to load due to DNS and connection steps that are not necessary when fonts are hosted on the same domain.

- <https://fonts.gstatic.com/s/poppins/v22/pxiByp8kv8JHgFVrLGT9Z1xIFQ.woff2>
- <https://fonts.gstatic.com/s/robotoslab/v34/BngMUXZYTXPivIBgJJSb6ufN5qU.woff2>
- <https://fonts.gstatic.com/s/poppins/v22/pxiEyp8kv8JHgFVrJJfecg.woff2>
- <https://fonts.gstatic.com/s/poppins/v22/pxiByp8kv8JHgFVrLEj6Z1xIFQ.woff2>

### Relevant Experiments

#### Self-Host 3rd Party Files

This experiment will fetch specified files server-side and reference them on the same domain. Note: the overrides happen at the host level, so any requests from that host will now be self-hosted.

##### ▼ Assets included in experiment:

- [fonts.gstatic.com](https://fonts.gstatic.com)

#### Preconnect 3rd Party Hosts

This experiment will add a `link` with `rel="preconnect"` for specified hosts, which saves time for those steps when the resource is later requested.

▼ **Assets included in experiment:**

- `fonts.gstatic.com`

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Preload 3rd Party Files

This experiment will add a `link` with `rel="preload"` for specified hosts, causing the browser to fetch the file early and at a high priority.

▼ **Assets included in experiment:**

- ☒ `https://fonts.gstatic.com/s/poppins/v22/pxiByp8kv8JHgFVrLGT9Z1xIFQ.woff2`
- ☒ `https://fonts.gstatic.com/s/robotoslab/v34/BngMUXZYTXPtVIBgJJSb6ufN5qU.woff2`
- ☒ `https://fonts.gstatic.com/s/poppins/v22/pxiEyp8kv8JHgFVrJjfecg.woff2`
- ☒ `https://fonts.gstatic.com/s/poppins/v22/pxiByp8kv8JHgFVrLEj6Z1xIFQ.woff2`

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Make Fonts Timeout

This experiment directs specified hosts to WebPageTest's blackhole server, which will hang indefinitely until timing out. Use this experiment to test your site's ability to display fallback text if fonts do not load quickly.

▼ **Assets included in experiment:**

- `fonts.gstatic.com`



▼ **Zero unused preloads were found.**

Preloaded resources are fetched at a high priority, delaying the arrival of other resources in the page. In the case where a preloaded resource is never actually used by the page, that means potentially critical requests will be delayed, slowing down the initial loading of your site.

▼ **This site uses a CDN for delivering its files.**

A Content Delivery Network (CDN) distributes a website's files throughout the world, reducing request latency.

▼ **1 static file has inadequate cache settings.**

Cache settings can instruct browsers and intermediaries to store recent versions of a site's static files (JavaScript, CSS, Images, fonts...) for reuse, reducing page weight and latency.

- WARNING (24.0 hours): <https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&family=Roboto+Slab:wght@400;700;900&subset=latin>



▼ **Zero requests were found that resulted in an HTTP redirect.**

HTTP redirects can result in additional DNS resolution, TCP connection and HTTPS negotiation times, making them very costly for performance, particularly on high latency networks.

▼ **Final HTML (DOM) size is significantly larger than initially delivered HTML (0.78kb larger, or 10.22% of total HTML).**

Typically this is due to over-reliance on JavaScript for generating content, but increases can also happen as a result of browsers normalizing HTML structure as well. When critical HTML content is generated with JavaScript in the browser, several performance bottlenecks can arise:

- Before content can be generated client-side, the browser must first parse, evaluate, and sometimes also request JavaScript over the network.
- If the generated HTML contains references to external assets (images for example), the browser will not be able to discover any assets until the HTML is fully rendered.



## Relevant Experiments

### Mimic Pre-rendered HTML

This experiment mimics server-generated HTML by swapping the initial HTML with the fully rendered HTML from this test run. **Note:** this will very likely break site behavior, but is potentially useful for comparing early metrics and assessing whether moving logic to the server is worth the effort.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Is it Usable?

**Needs Improvement.** This site had good layout stability. It took little time to become interactive. It had 1 accessibility issues, 1 serious. Some HTML was generated after delivery, potentially delaying usability.

WebPageTest ran 5 diagnostic checks related to this category and found 3 opportunities.

## ▼ **Layout shifts exist and may be caused by images missing aspect ratio.**

The CLS score is 0.006. 2 layout-critical images are lacking an aspect ratio, meaning the browser has no way of knowing how tall or wide an image is until it loads. This can cause content to shift as the image loads.

- /static/logo-light.svg
- /static/hero-image.svg

### Relevant Experiments

#### **Add Aspect Ratio to Images**

This experiment adds `width="..."` and `height="..."` attributes to specified images, matching their natural width and height, to provide an aspect ratio.

##### ► **Assets included in experiment:**

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## ▼ **The main thread was not blocked for any significant time.**

When files block the main thread, users are unable to interact with the page content. Typically, parsing and executing large JavaScript files, as well as running long JavaScript tasks can block the main thread and should be avoided.

## ▼ **Meta Viewport tag is configured properly.**

A meta viewport tag will help a mobile-friendly site scale and display properly on small screen devices.



## Accessibility Issues were Detected

Axe found 1 accessibility issues: 1 serious,

### Relevant Tips

#### Make the following changes to improve accessibility:

##### **SERIOUS** Elements Must Have Sufficient Color Contrast [More Info](#)

- Fix any of the following: Element has insufficient color contrast of 3.03 (foreground color: #cb6dea, background color: #ffffff, font size: 12.0pt (16px), font weight: normal). Expected contrast ratio of 4.5:1 `<a href="/login" class="primary-btn">Sign Up Now</a>`
- Fix any of the following: Element has insufficient color contrast of 4.48 (foreground color: #888888, background color: #222222, font size: 12.0pt (16px), font weight: normal). Expected contrast ratio of 4.5:1 `<a href="#">Home</a>`
- Fix any of the following: Element has insufficient color contrast of 4.48 (foreground color: #888888, background color: #222222, font size: 12.0pt (16px), font weight: normal). Expected contrast ratio of 4.5:1 `<a href="#features">Features</a>`
- Fix any of the following: Element has insufficient color contrast of 4.48 (foreground color: #888888, background color: #222222, font size: 12.0pt (16px), font weight: normal). Expected contrast ratio of 4.5:1 `<a href="#">Pricing</a>`
- Fix any of the following: Element has insufficient color contrast of 4.48 (foreground color: #888888, background color: #222222, font size: 12.0pt (16px), font weight: normal). Expected contrast ratio of 4.5:1 `<a href="#">Contact</a>`
- Fix any of the following: Element has insufficient color contrast of 4.48 (foreground color: #888888, background color: #222222, font size: 12.0pt (16px), font weight: normal). Expected contrast ratio of 4.5:1 `<a href="#">Privacy Policy</a>`
- Fix any of the following: Element has insufficient color contrast of 4.48 (foreground color: #888888, background color: #222222, font size: 12.0pt (16px), font weight: normal). Expected contrast ratio of 4.5:1 `<a href="#">Terms of Service</a>`
- Fix any of the following: Element has insufficient color contrast of 4.48 (foreground color: #888888, background color: #222222, font size: 12.0pt (16px), font weight: normal). Expected contrast ratio of 4.5:1 `<a href="#">HIPAA Compliance</a>`
- Fix any of the following: Element has insufficient color contrast of 4.48 (foreground color: #888888, background color: #222222, font size: 12.0pt (16px), font weight: normal). Expected contrast ratio of 4.5:1 `<p>Subscribe to our newsletter for the latest updates</p>`
- Fix any of the following: Element has insufficient color contrast of 3.03 (foreground color: #ffffff, background color: #cb6dea, font size: 10.0pt (13.3333px), font weight: normal). Expected contrast ratio of 4.5:1 `<button type="submit">Subscribe</button>`

- Fix any of the following: Element has insufficient color contrast of 4.48 (foreground color: #888888, background color: #222222, font size: 12.0pt (16px), font weight: normal). Expected contrast ratio of 4.5:1 <p>© 2025 Vishnu Thandel. All rights reserved.</p>

## ▼ Final HTML (DOM) size is significantly larger than initially delivered HTML (0.78kb larger, or 10.22% of total HTML).

Typically this is due to over-reliance on JavaScript for generating content, but increases can also happen as a result of browsers normalizing HTML structure as well. When critical HTML content is generated with JavaScript in the browser, it can increase the time it takes for content to be made accessible to assistive technology such as screen readers.

### Relevant Tips

#### Look for ways to deliver more HTML content from the start

Many modern frameworks offer patterns for generating useful HTML on the server.

## Is it Resilient?

**Not bad...** This site had render-blocking 3rd party requests that could be a single point of failure. It had no security issues. Some HTML was generated after delivery, which can cause fragility.

WebPageTest ran 4 diagnostic checks related to this category and found 2 opportunities.

## ▼ Potential SPOF: 1 3rd-party request is blocking page rendering.

By default, references to external JavaScript and CSS files will block the page from rendering. Third-party blocking requests are particularly risky, as your page's access relies on their response time and availability.'

- <https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css>

## Relevant Experiments

### Preconnect 3rd Party Hosts

This experiment will add a `link` with `rel="preconnect"` for specified hosts, which saves time for those steps when the resource is later requested.

#### ▼ Assets included in experiment:

- [cdnjs.cloudflare.com](https://cdnjs.cloudflare.com)

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

### Preload 3rd Party Files

This experiment will add a `link` with `rel="preload"` for specified files, causing the browser to fetch the file early and at a high priority.

#### ▼ Assets included in experiment:

- <https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css>

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

### Self Host 3rd Party Files

This experiment will fetch these files server-side and reference them on the same domain. Note: the overrides happen at the host level, so any requests from that host will now be self-hosted.

▼ **Assets included in experiment:**

- cdnjs.cloudflare.com

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Make Scripts Timeout

This experiment directs specified hosts to WebPageTest's blackhole server, which will hang indefinitely until timing out. Use this experiment to test your site's ability to serve content if these services hang.

▼ **Assets included in experiment:**

- cdnjs.cloudflare.com

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Block Scripts

This experiment causes specified requests to fail immediately. Use this experiment to test your site's ability to serve content if these services are unavailable.

▼ **Assets included in experiment:**

- https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

---

▼ **Zero security vulnerabilities were detected by Snyk**

Snyk has found 0 security vulnerabilities with included packages.

---

▼ **Zero resources were found that were loaded over an insecure connection.**

Loading requests over HTTPS necessary for ensuring data integrity, protecting users personal information, providing core critical security, and providing access to many new browser features.

---

▼ **Final HTML (DOM) size is significantly larger than initially delivered HTML (0.78kb larger, or 10.22% of total HTML).**

Typically this is due to over-reliance on JavaScript for generating content, but increases can also happen as a result of browsers normalizing HTML structure as well. Common issues such as JavaScript errors and third-party network delays and outages can present potential single points of failure.

### Relevant Experiments

#### Disable Scripts

This experiment makes all scripts (inline and external) unrecognizable as javascript by the browser in order to demonstrate whether the site will still be usable if JavaScript fails to properly run.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

#### Make Scripts Timeout



This experiment directs specified requests to WebPageTest's blackhole server, which will hang indefinitely until timing out. Use this experiment to test your site's ability to serve content if these services hang.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Block Script Requests

This experiment causes specified requests to fail immediately. Use this experiment to test your site's ability to serve content if these services are unavailable.

### ▼ Assets included in experiment:

- /static/landing.js

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

# Create Experiments

**Advanced.** Use this section to create custom experiments to add to your test.

## ▼ Add HTML to document

These experiments allow you to add arbitrary html to a page, which can for example, enable to you test the impact of adding scripts, 3rd-party tags, or resource hints.

Relevant Experiments

### Add HTML to start of head

This experiment adds arbitrary HTML text to the start of the head of the tested website.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

### Add HTML to end of head

This experiment adds arbitrary HTML text to the end of the head of the tested website.



**Run this Experiment with...**

### Add HTML to end of body

This experiment adds arbitrary HTML text to the end of the body of the tested website.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.



## Find and Replace Text

This experiment allows you to find and replace arbitrary text or html in a request.

Relevant Experiments

## Find/Replace Text

This experiment will find and replace occurrences of text in the page, with the option of using regular expressions, capturing parentheses, and flags as well

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.



## Delivery Optimizations

These experiments allow you change attributes and manipulate how assets are delivered. Note expected syntax examples when using.

### Relevant Experiments

#### Defer Render-Blocking Scripts

This experiment adds a `defer` attribute to render-blocking scripts, causing the browser to fetch them in parallel while showing the page. Deferred scripts still execute in the order they are defined in source.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

#### Async Blocking Scripts

This experiment adds a `async` attribute to render-blocking scripts, causing the browser to fetch them in parallel while showing the page. Async scripts are not guaranteed to execute in the order they are defined in source.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Disable JavaScript

This experiment effectively disables JavaScript (and enables noscript elements) to test a site's JavaScript dependence. It works by changing `script` elements into `scrypt` elements, and `noscript` elements into `noscrapt` elements so that the page weight remains and behaviors mimic a JS-disabled environment.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Load CSS Async

This experiment loads specified stylesheets in a way that allows the page to begin rendering while they are still loading. Note that this may cause a flash of unstyled content.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Add Image Aspect Ratio

This experiment adds `width="..."` and `height="..."` attributes to specified images, matching their natural width and height, to provide an aspect ratio.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Inline External Files

This experiment embeds the contents of specified external scripts or stylesheets directly into the HTML. This increases the size of the HTML, but can often allow page to display sooner by avoiding server round trips. **Note:** The inline experiment may break relative references to images and other assets inside the file. These would need proper path changes or absolute URLs in production.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Preload Files

This experiment adds a `<link rel="preload" type="image" href="...">` to the `head` of your HTML document, causing browsers to request a file earlier and at a higher priority than they otherwise might.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Remove Preloads

This experiment removes specified unused preloads from the page, allowing other critical resources to be requested earlier.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Preconnect Domains

This experiment will add a `link` with `rel="preconnect"` for specified hosts, which saves time for those steps when the resource is later requested.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Lazy Load Images

This experiment adds `loading="lazy"` attributes to specified images.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Remove Lazy Loading

This experiment removes `loading="lazy"` attributes from specified images.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Minify CSS

This experiment removes whitespace from same-domain CSS files.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Add Priority Hint

*(Chromium-only)* This experiment adds an `fetchpriority="high"` attribute to specified images, causing them to request earlier at a higher priority.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Remove Priority Hint

*(Chrome-only)* This experiment removes an existing `fetchpriority` attribute to specified images, causing them to request at their default priority.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Add `font-display: swap`

This experiment changes the way a custom font loads so that text can appear while fonts are loading

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Make Hosts Timeout

This experiment directs specified hosts to WebPageTest's blackhole server, which will hang indefinitely until timing out. Use this experiment to test your site's ability to serve content if these services hang.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Block Requests

This experiment causes specified requests to fail immediately. Use this experiment to test your site's ability to serve content if these services are unavailable.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.

## Self-Host Requests

This experiment will fetch these requests server-side and reference them on the same domain. Note: the overrides happen at the host level, so any requests from that host will now be self-hosted.

**Get WebPageTest** Pro

FOR UNLIMITED EXPERIMENTS.



**Experiment Runs: 3 (12 total runs)**

Each experiment run uses 2 test runs (1 experiment, 1 control) for each first & repeat view

**Re-Run Test with Experiments**



[Contact Sales](#)

[Terms of Service](#)

[Privacy Policy](#)

© [Catchpoint Systems, Inc.](#) All rights reserved.