HTTP Observatory Report

ⓘ Report Feedback

Scan summary: medicalcheckin.com → www.medicalcheckin.com

A+

Score: 110 / 100

Scan Time: Just now

Tests Passed: 9 / 10

↗ since last scan

Scan results

Scoring

Test	Score	Reason	Recommendation
Content Security Policy (CSP)	+10 ✓	Content Security Policy (CSP) implemented with `default-src 'none'`, no `'unsafe'` and form-action is set to `'none'` or `'self'`	None
Cookies	-	No cookies detected	None
Cross Origin Resource Sharing (CORS)	0 ✓	Content is not visible via cross-origin resource sharing (CORS) files or headers.	None
Redirection	-5 ✗	Initial redirection from HTTP to HTTPS is to a different host, preventing HSTS.	HSTS headers aren't recognized when set over HTTP, so redirect to the same host on HTTPS first, then redirect to the final host.

Test	Score	Reason	Recommendation
Referrer Policy	-	**Referrer-Policy** header not implemented.	Set to **strict-origin-when-cross-origin** at a minimum.
Strict Transport Security (HSTS)	0 ✓	**Strict-Transport-Security** header set to a minimum of six months (15768000).	Consider preloading: this requires adding the **preload** and **includeSubDomains** directives and setting **max-age** to at least **31536000** (1 year), and submitting your site to https://hstspreload.org/.
Subresource Integrity	-	Subresource Integrity (SRI) not implemented, but all scripts are loaded from a similar origin.	Add SRI for bonus points.
X-Content-Type-Options	0 ✓	**X-Content-Type-Options** header set to **nosniff**.	None
X-Frame-Options	+5 ✓	**X-Frame-Options** (XFO) implemented via the CSP frame-ancestors directive.	None
Cross Origin Resource Policy	-	Cross Origin Resource Policy (CORP) is not implemented (defaults to **cross-origin**).	None

CSP analysis

✓ Content Security Policy (CSP) implemented with **default-src 'none'**, no **'unsafe'** and form-action is set to **'none'** or **'self'**

Test	Result	Info
Blocks execution of inline JavaScript by not allowing **'unsafe-inline'** inside **script-src**	✓	Blocking the execution of inline JavaScript provides CSP's strongest protection against cross-site scripting attacks. Moving JavaScript to external files can also help make your site more maintainable.
Blocks execution of JavaScript's **eval()** function by not allowing **'unsafe-eval'** inside **script-src**	✓	Blocking the use of JavaScript's **eval()** function can help prevent the execution of untrusted code.

Test	Result	Info
Blocks execution of plug-ins, using `object-src` restrictions	✓	Blocking the execution of plug-ins via `object-src 'none'` or as inherited from `default-src` can prevent attackers from loading Flash or Java in the context of your page.
Blocks inline styles by not allowing `'unsafe-inline'` inside `style-src`	✓	Blocking inline styles can help prevent attackers from modifying the contents or appearance of your page. Moving styles to external stylesheets can also help make your site more maintainable.
Blocks loading of active content over HTTP or FTP	✓	Loading JavaScript or plugins can allow a man-in-the-middle to execute arbitrary code or your website. Restricting your policy and changing links to HTTPS can help prevent this.
Blocks loading of passive content over HTTP or FTP	✓	This site's Content Security Policy allows the loading of passive content such as images or videos over insecure protocols such as HTTP or FTP. Consider changing them to load them over HTTPS.
Clickjacking protection, using `frame-ancestors`	✓	The use of CSP's `frame-ancestors` directive offers fine-grained control over who can frame your site.
Deny by default, using `default-src 'none'`	✓	Denying by default using `default-src 'none'`can ensure that your Content Security Policy doesn't allow the loading of resources you didn't intend to allow.
Restricts use of the `<base>` tag by using `base-uri 'none'`, `base-uri 'self'`, or specific origins.	✓	The `<base>` tag can be used to trick your site into loading scripts from untrusted origins.
Restricts where `<form>` contents may be submitted by using `form-action 'none'`, `form-action 'self'`, or specific URIs	✓	Malicious JavaScript or content injection could modify where sensitive form data is submitted to or create additional forms for data exfiltration.
Uses CSP3's `'strict-dynamic'` directive to allow dynamic script loading (optional)	-	`'strict-dynamic'` lets you use a JavaScript shim loader to load all your site's JavaScript dynamically, without having to track `script-src` origins.

Cookies

No cookies detected

Raw server headers

Header	Value
Date	Fri, 29 Aug 2025 16:43:53 GMT
Server	Apache
Upgrade	h2
Connection	Upgrade, close
Content-Type	text/html; charset=UTF-8
Accept-Ranges	bytes
X-Frame-Options	SAMEORIGIN
X-Xss-Protection	1; mode=block
Transfer-Encoding	chunked
X-Content-Type-Options	nosniff
Content-Security-Policy	default-src 'none';font-src 'self';style-src 'self' *.stripe.com;img-src 'self' data: *.stripe.com ;connect-src 'self'; form-action 'self' *.mobilecheckin.net ;object-src 'none';base-uri 'none';script-src 'self' 'nonce-ZSun567s' 'nonce-aLHYyZRkYrFmmphnk2DQXAAAA8A' *.stripe.com; media-src 'self';frame-src 'self' *.stripe.com *.youtube.com; frame-ancestors *.stripe.com *.youtube.com;
Strict-Transport-Security	max-age=63072000; includeSubdomains; preload

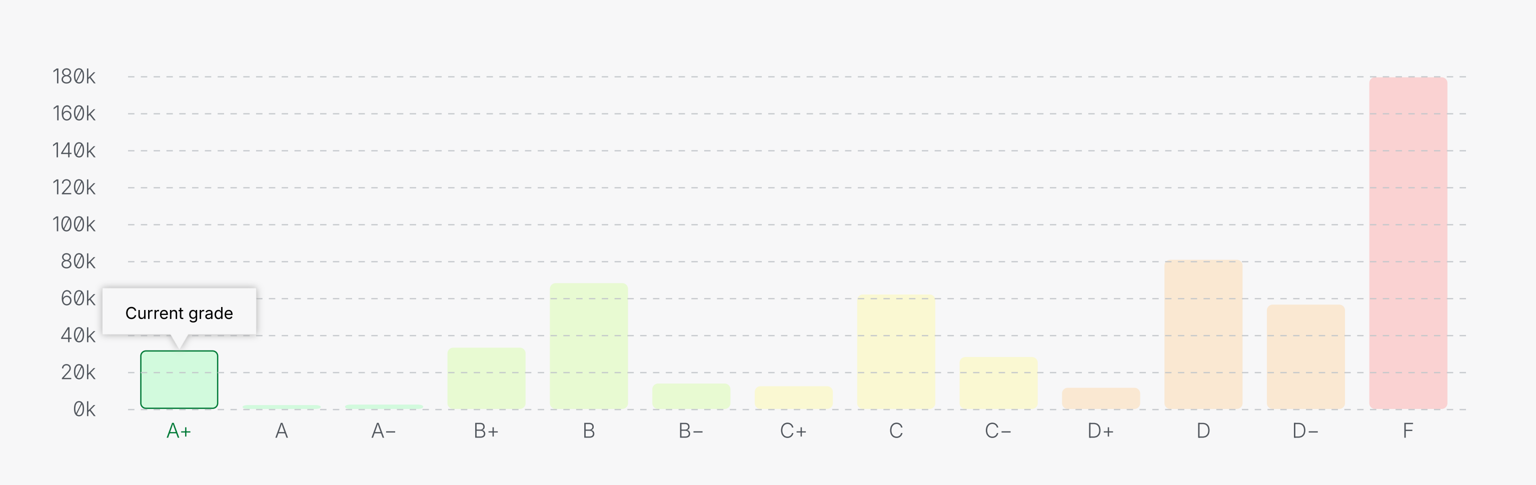
Scan history

Changes in score over time

Date	Score	Grade
Aug 11, 2024, 7:15:52 PM	110	A+
Aug 11, 2024, 7:14:10 PM	30	D
Aug 11, 2024, 7:09:18 PM	45	C-
Jun 24, 2024, 1:43:57 PM	115	A+
Apr 4, 2023, 7:19:06 PM	110	A+
Feb 26, 2020, 10:23:43 PM	75	B
Dec 11, 2018, 4:08:35 PM	65	B-

Benchmark comparison

Performance trends from the past year



Refer to this graph to assess the website's current status. By following the recommendations provided and rescanning, you can expect an improvement in the website's grade.