Content security policy issue document

Addressing the issue:

Chrome recently updated its policy to prevent XSS (cross-site scripting) attacks, which can be injected through resources like images, scripts, frames, etc. This has caused issues on our website. To resolve this, we need to explicitly allow resources from other domains, such as AWS S3. However, while addressing the CSP issue, we encountered additional challenges, which are listed below.

Method 1 -> Plugins

The list of plugins we used to try to resolve the Content Security Policy issue in Chrome.

1. Headers Security Advanced & HSTS WP
2. HTTP Headers
3. Cookies and content security
4. HTTP Security header by monks
5. GD security Headers
6. Content security policy manager

When using the above plugins, we commonly face an issue where images do not load, even though they are hosted on the same domain. These plugins often create new directives, such as those related to images, Facebook, and, most notably, they block resources from **scienceopen.com**. However, the **Headers Security Advanced & HSTS WP** plugin neither causes this issue nor resolves it

Some plugins stop loading CSS and JS, causing layout and functionality issues

Method 2 -> Add function to avoid CSP

This code adds a custom Content Security Policy (CSP) header to your WordPress site. It restricts image sources (`img-src`) to only allow images from the same domain (`'self'`) and from `https://www.google.co.in`, while frames (`frame-src`) can only be loaded from the same domain and `https://td.doubleclick.net`. The header is sent with every page response using the `send_headers` action.

```
function add_custom_csp_header() {
header("Content-Security-Policy: img-src 'self' https://www.google.co.in; frame-src 'self' https://td.doubleclick.net;");
}
```

```
add_action('send_headers', 'add_custom_csp_header');
```

This code blocks content from the same domain. To resolve this, I added the blocked URLs to the CSP header function.

```
function add_custom_csp_header() { header("Content-Security-Policy: img-src 'self' https://cvia-journal.org https://pbs.twimg.com profile_images https://s.w.org/images/core/emoji/15.0.3/svg/ https://www.scienceopen.com https://td.doubleclick.net https://www.facebook.com/privacy_sandbox/pixel/register/trigger/ https://www.facebook.com/tr/ https://www.googletagmanager.com https://www.google.co.in data:; frame-src https://www.googletagmanager.com;"); }
add_action('send_headers', 'add_custom_csp_header');
```

This code fixes the same-domain issue, but other domains, such as **www.scienceopen.com**, are still blocked.

Method 3 -> Changing Apache configurations in the .htaccess file.

If I attempt to explicitly add domains using the .htaccess file with the help of mod_headers.c, it causes the site to crash. This does not fix the issue, and the action is not easily reversible. Once the site crashes, there is no way to revert it back, so it is not recommended to perform this on a live site.

```
<IfModule mod_headers.c>
Header set Content-Security-Policy "img-src 'self' https://cvia-journal.org https://pbs.twimg.com profile_images https://s.w.org/images/core/emoji/15.0.3/svg/ https://www.scienceopen.com https://td.doubleclick.net https://www.facebook.com/privacy_sandbox/pixel/register/trigger/ https://www.facebook.com/tr/ https://www.googletagmanager.com https://www.google.co.in data:; frame-src https://www.googletagmanager.com;"
</IfModule>
```

Note: the action is not easily reversible so not recommended to try on live site

Method 4 -> Meta Tags (Recommended).

We need to add the source URLs in meta tags to indicate that I trust those external domains to be used on my site and avoid CSP violations. However, this approach did not fully resolve the issue. It blocked details related to scienceopen.com on our site, and the CSP violation remains unresolved.

<meta http-equiv="Content-Security-Policy" content="img-src 'self' https://cvia-journal.org https://pbs.twimg.com profile_images https://s.w.org/images/core/emoji/15.0.3/svg/ https://www.scienceopen.com https://td.doubleclick.net https://www.facebook.com/privacy_sandbox/pixel/register/trigger/ https://www.facebook.com/tr/ https://www.googletagmanager.com https://www.google.co.in data:; frame-src https://www.googletagmanager.com;">

Important note: The methods mentioned above are quite similar, but they are implemented in different ways to address the CSP issue. However, the CSP issue remains unresolved.