# Best “FlyingPress Settings” for A SuperFast WordPress Website

**By Kuldeep Rathore**

FlyingPress is undoubtedly the Best WordPress Speed Optimization plugin.

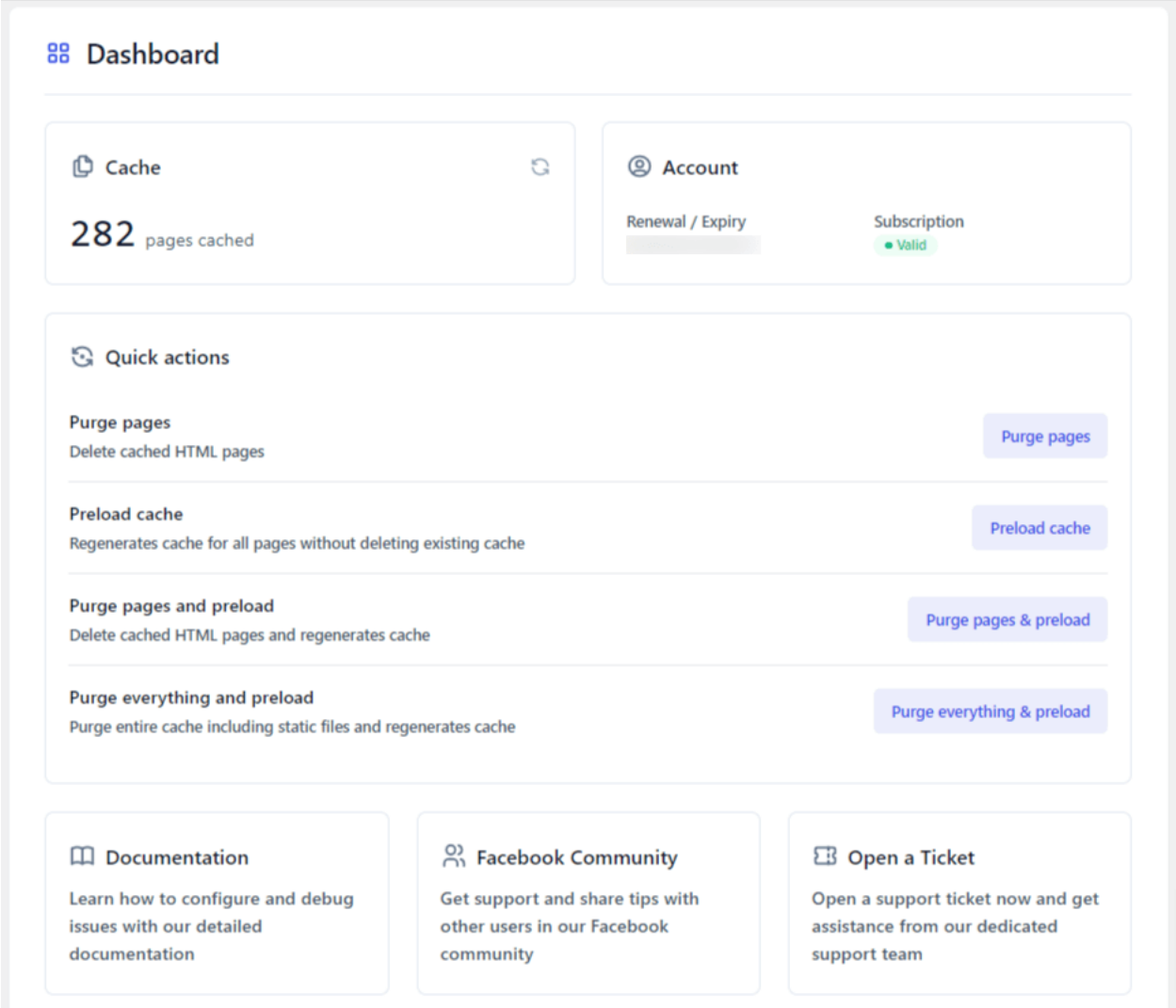
After going through documentation & reading various guides, I’ve prepared this FlyingPress setting (guide) to help you configure the plugin the right way for the best performance by which I & many others have got great results.

In case you don’t have a license yet, I encourage you to get one as FlyingPress is the #1 WordPress Speed Optimization plugin.

👉Click here to Get FlyingPress at 30% OFF (Renewal discount)

If you already have the plugin, here are the “Best FlyingPress settings” that you need to configure to make your site fly...

# 1. Dashboard




➔ **Purge pages:** HTML pages are purged from the cache
➔ **Preload cache:** cached pages are overwritten instead of purging the entire cache.
➔ **Purge pages and preload:** HTML pages are purged, then they are preloaded.
➔ **Purge everything and preload:** Purge all (HTML pages, CSS, JS, fonts) & preload the entire cache.
➔ **Documentation** – leads to FlyingPress documentation page.
➔ **Facebook community** – leads to the FlyingPress Facebook Group which is a great place to browse questions, ask new questions,

give Gijo feedback, see upcoming features/updates & connect with like-minded people.

➔ **Open a ticket** – leads to the contact page where you can get support (directly from the developer)

## 2. Cache Settings

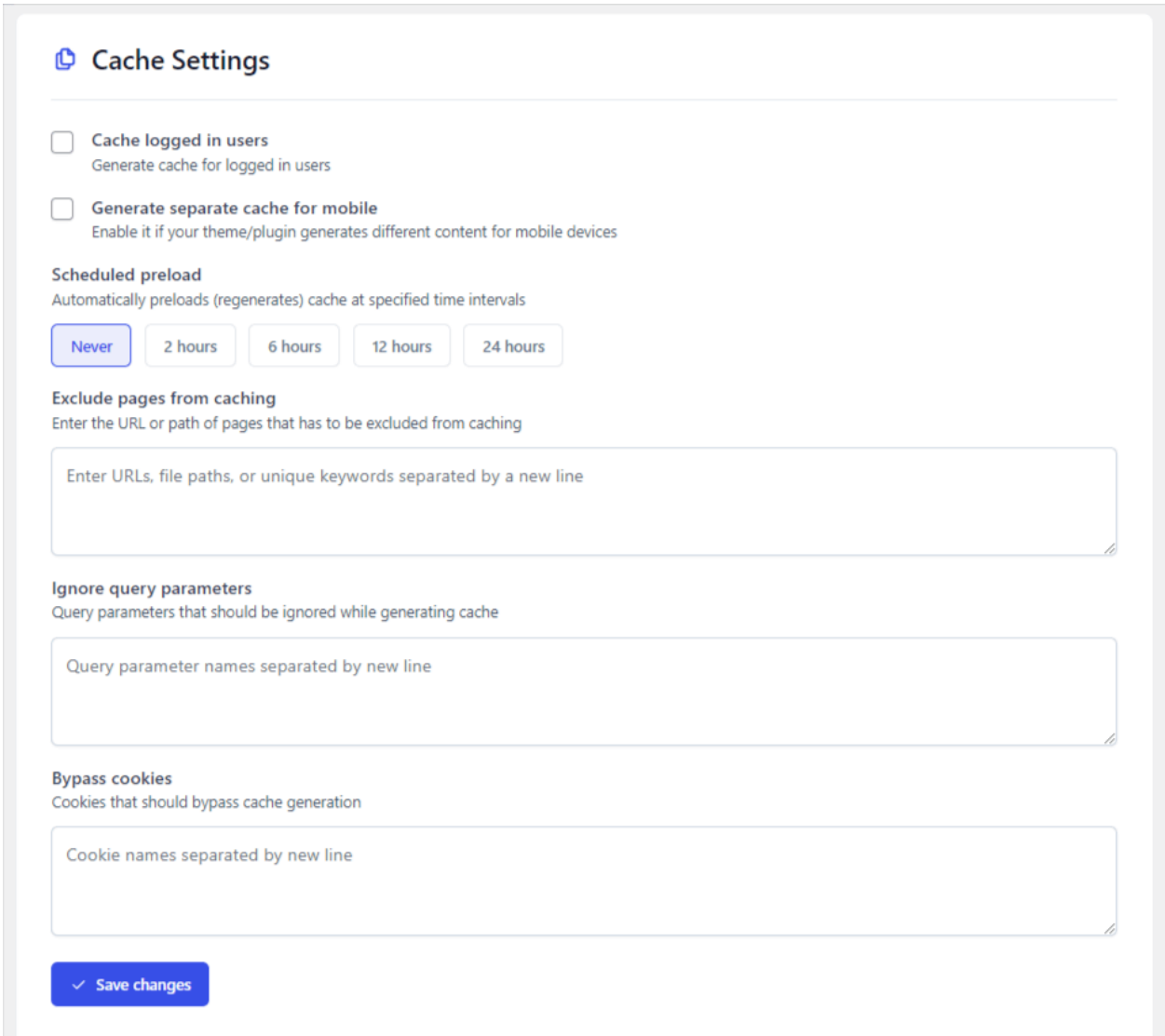
**Note: FlyingPress cache by default is always on. If you're using other caching layers like Varnish, Nginx, etc, FlyingPress will serve as a fallback cache (it improves cache hit ratio)*

➔ **Cache logged-in users: Disabled** – only enable if you're running a membership site (or similar) and have users logging in that need their own cache, otherwise, it requires server resources, so it's better to turn off.

➔ **Generate Separate Cache for Mobile: Disabled** – If your website is fully responsive, no need to enable it.

➔ **Scheduled preload: Never** – Gijo (developer) recommends you should only change it when facing problems with the cache where it's not updated on time (the cache will be overwritten, not purged).

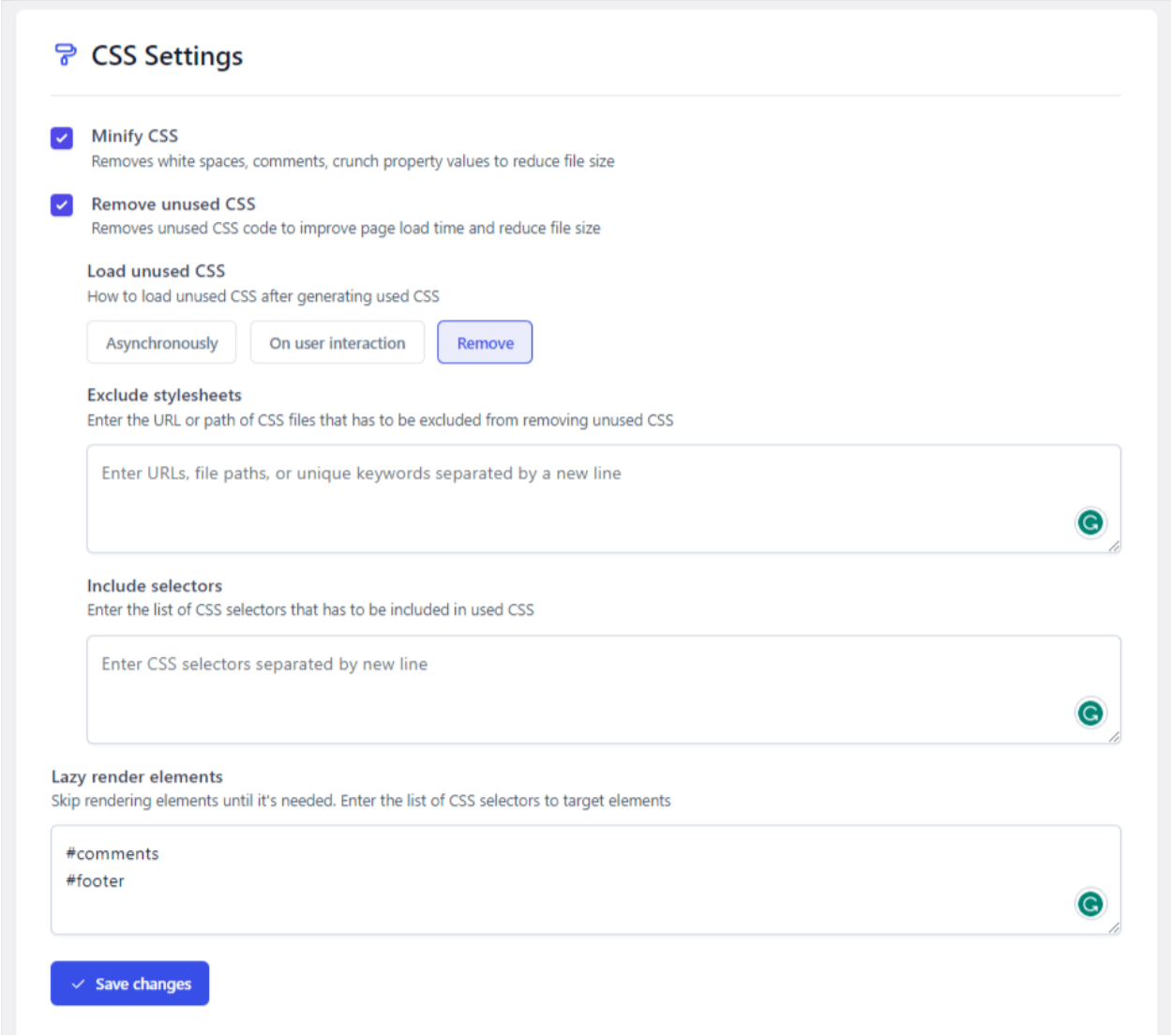
➔ **Exclude Pages from Caching:** Imp eCommerce pages, admin/login pages, and several others are automatically excluded from the cache (see complete list), but you can add more here.




➔ **Ignore Query Parameters:** If you have query parameters that should be ignored and not cached (eg. ad campaigns), add them here. FlyingPress has a list of query parameters that it ignores by default.

➔ **Bypass cookies:** add cookies that should bypass cache generation. This can decrease performance and is not recommended unless you need to.

## 3. CSS Settings




➔ **Minify CSS:** Enable – it usually gives you better results as it removes white spaces, comments, etc reducing the file size.

➔ **Remove unused CSS:** Enable – removes unused CSS code resulting in improved page load time.

FlyingPress loads used CSS in a separate file which is faster for users, but slightly worse for scores whereas WP Rocket loads

used CSS inline which is better for scores, but slower for actual users. Perfmatters has an option for both (I prefer using FlyingPress).

➔ **Load unused CSS:** Asynchronously – the rest of your CSS files will be loaded asynchronously with the rest of your files. This is the safest option. Setting it to remove is best for performance but it can break your site.

➔ **Exclude stylesheets:** if the setting above, breaks your site, simply exclude problematic CSS files.

➔ **Include selectors:** if FlyingPress isn't able to detect certain CSS selectors to include in used CSS, you can add them here.

➔ **Lazy render elements** – it's pretty much like lazy loading images only for anything on your site if it loads below the fold. Comments (#comments) and footer (#footer) are common as well as Elementor/Divi sections and WooCommerce-related products.

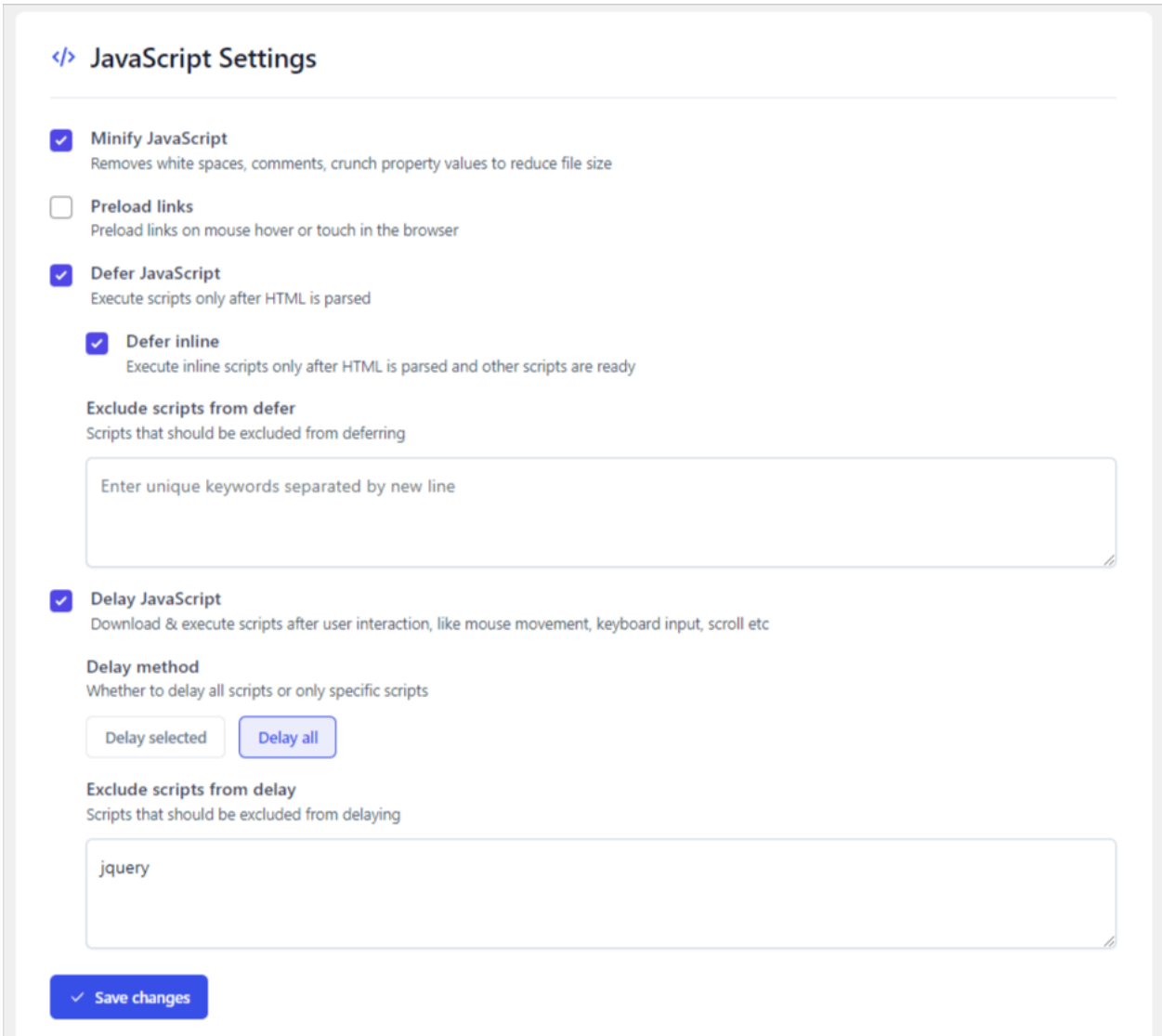
For this, you'll need to check your own site's CSS selectors by viewing your site → right-click any element you want to lazy render → inspect → right-click highlighted code → copy → copy selector → paste (for reference, check this short video)

## 4. JavaScript Settings

➔ **Minify JavaScript:** Enable – it's the same as minify CSS (removes white spaces, comments, etc) that reduces the file size.

➔ **Preload links:** Enable if you're on VPS hosting, Disable if you're on a shared server – when users hover over a link, that page is preloaded so by the time users actually click it, it appears to load instantly.

While this doesn't improve scores, it does improve perceived load time. Whether you should enable it depends on whether your hosting is powerful enough, and whether your users hover over lots of links. Ideally, you can enable it on a shared server as well.



➔ **Defer JavaScript: Enable** – it executes scripts only after HTML is parsed. This can fix render-blocking errors by loading JS asynchronously.

Speed Experts like WP Johnny suggest disabling it since it only improves page speed scores but actually slows down your site, so test it.

➔ **Defer inline: Enable** – if you enabled the previous setting, enable this too.

➔ **Exclude scripts from defer:** enter the scripts that should be excluded from deferring.

For this, you can check the Console tab in Chrome Dev Tools to see whether certain JS files are causing issues when being deferred, then exclude them here if needed. “hooks.min.js” and “i18n.min.js” are common exclusions in Perfmatters’ documentation.

➔ **Delay JavaScript: Delay all** – for best performance, set it to delay all but like removing unused CSS, you’ll need to exclude files from delaying. Turn it on & make sure to test it out.

If you’re having trouble, use “delay selected” and add JS files manually found in your “third-party code” + “remove unused JavaScript” report (test in page speed). FlyingPress already delays many of them like Google Analytics and reCAPTCHA, but there may be others that you can add here.

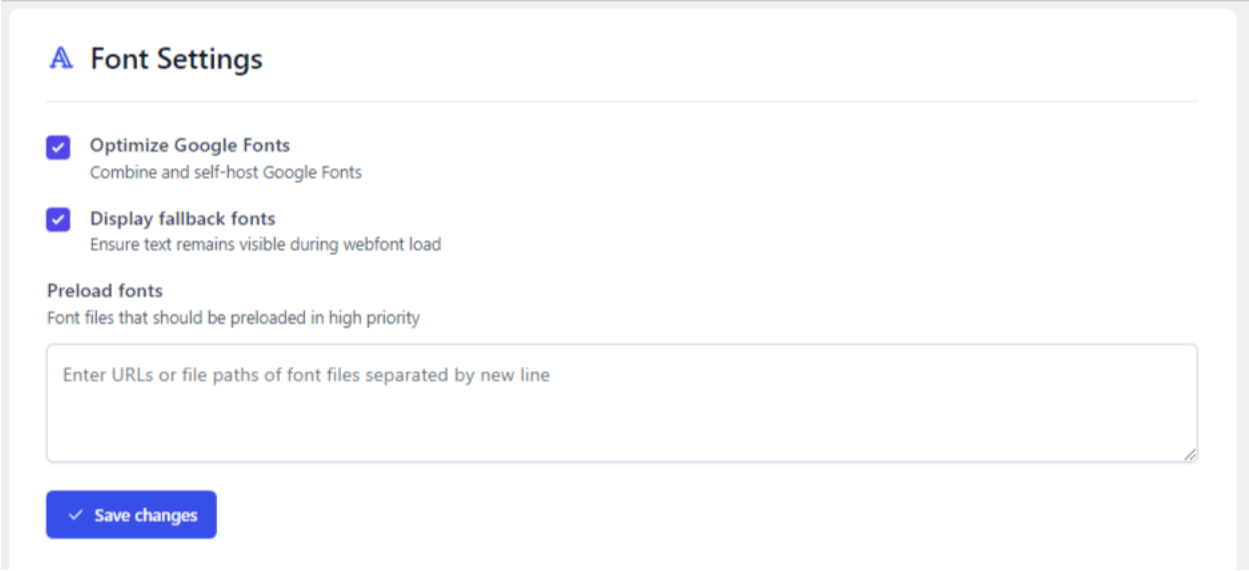
Here are some of the common third-party codes you can set to delay...

Google-analytics.com
Xfbml.customerchat.js
Fbevents.js
Widget.manychat.com
Cookie-law-info
Grecaptcha.execute
Static.hotjar.com
Hs-scripts.com
Embed.tawk.to
Disqus.com/embed.js
Client.crisp.chat
Matomo.js
Usefathom.com
Code.tidio.co
Metomic.io
Js.driftt.com
Cdn.onesignal.com

## 5. Font Settings

➔ **Optimize Google Fonts:** **Enable** – this will host fonts locally (instead of creating external requests to fonts.gstatic.com), combines, and inlines them.

➔ **Display fallback fonts:** **Enable** – this adds font-display: swap to CSS which fixes “ensure text remains visible during webfont load” in Page Speed Insights. Until your font is loaded, a fallback font is used to prevent FOIT (flash of invisible text), but it can cause FOUT (flash of unstyled text).




➔ **Preload fonts:** Add the font files that should be preloaded in high priority. To get data, view font files in a Waterfall chart (like GTmetrix) and test preloading them.

*Note that Gijo (FlyingPress developer), recommends only preloading fonts mentioned in the CSS file and fonts loading above the fold.*

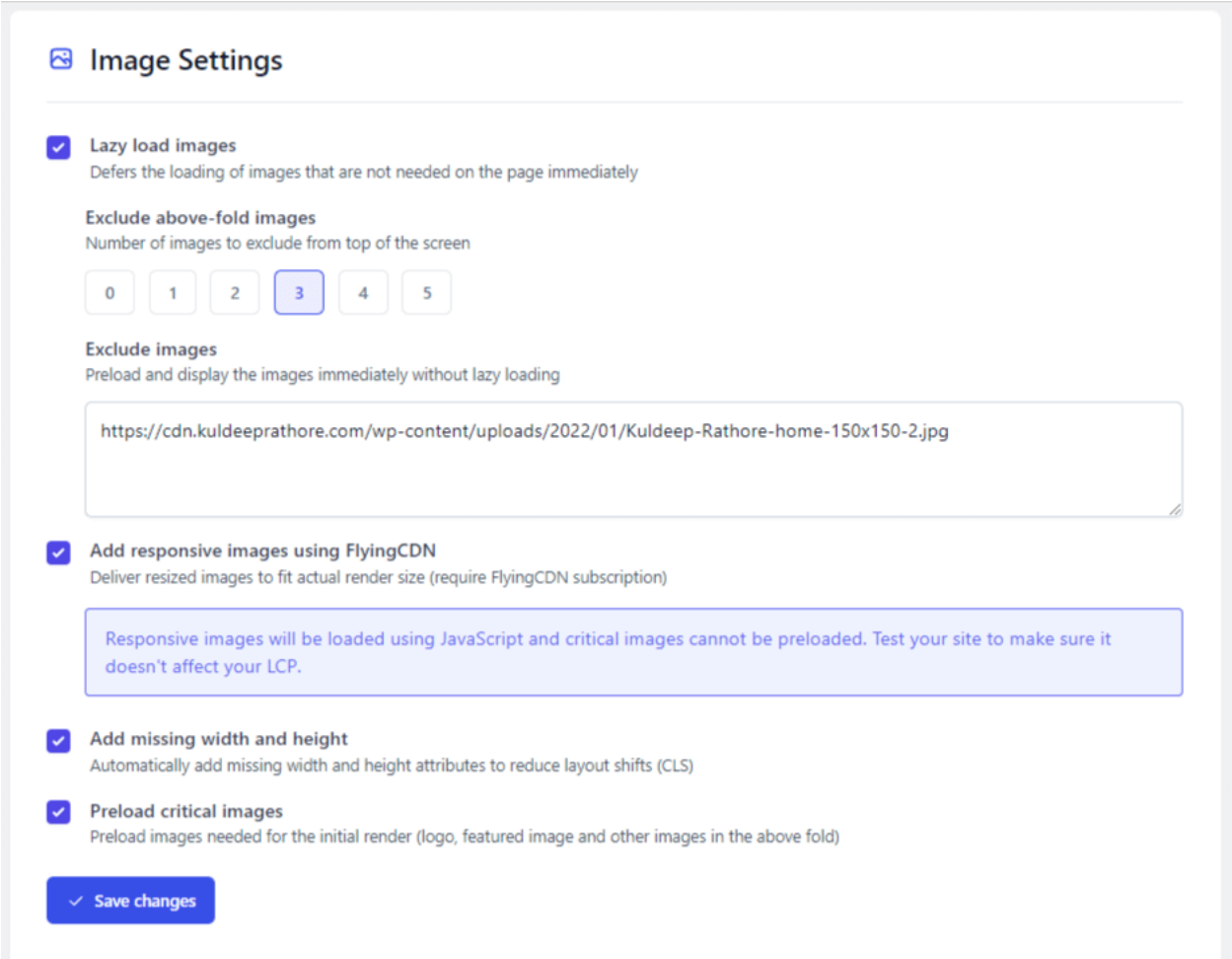
*When you're done, check for preloading errors in the Console tab of Chrome Dev Tools. Preloading fonts that aren't being loaded is just senseless, so make sure URLs are correct.*

## 6. Image Settings

➔ **Lazy load images:** Enable – It defers the images that are not needed to load on the page immediately. FlyingPress uses native lazy load & it works amazingly.

➔ **Exclude above-the-fold images:** Keep it between 2-3 – this should be the number of images that usually load above the fold.

This depends on site-to-site but for most, lazy loading the first 3 images should work great.




➔ I usually keep 1-2 images above the fold for faster page loading. Setting images from numbers (1-5) will automatically be excluded from lazy load and set as a high priority for better LCP.

If you want some of the images without lazy loading then you exclude them manually by adding their image URLs under Exclude Images.

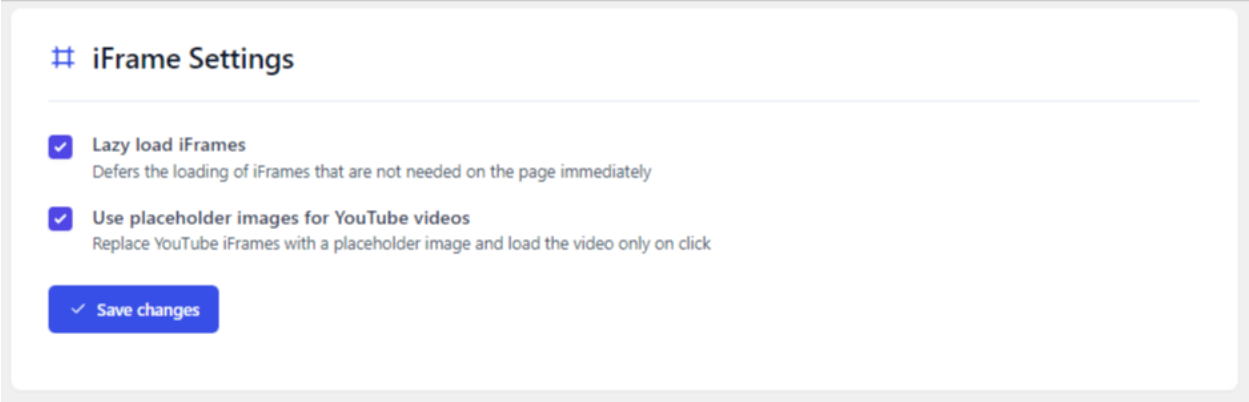
➔ **Add responsive images using FlyingCDN:** Enable if using FlyingCDN – serves smaller images to mobile for better mobile LCP when using FlyingCDN (which uses Bunny Optimizer from BunnyCDN). This can also be done with Cloudflare's image resizing, or ShortPixel Adaptive Images, and Optimole plugin.

➔ **Add missing width and height:** Enable – This adds width/height to images which helps reduce CLS & fixes "use explicit weight and height" in Page Speed Insights.

➔ **Preload critical images:** Enable – This will preload the images above the fold needed for initial rendering. FlyingPress automatically detects above-the-fold images and preloads them for better LCP.

➔ **Disable emoji:** Enable – This removes a small JavaScript code in WordPress that displays emojis & uses the default emojis of the visitor's browser.

## 7. iFrame Settings




➔ **Lazy load iFrames:** Enable – iFrames like embedded YouTube videos and Google Maps will only be loaded when they're near the viewport, then excluded when they're in the viewport.

➔ **Use placeholder image for YouTube iFrames:** Enable – replaces YouTube iFrames with a preview image and self hosts YouTube placeholder images to prevent external requests to i.ytimg.com.

This also means YouTube thumbnails can be cached and served from a CDN. It will load the video only when the user clicks on it.

## 8. CDN Settings

Although speed experts like Tom recommend using Cloudflare + APO, using FlyingCDN is also awesome as it uses BunnyCDN which comes with Bunny optimizer & geo-replication.

It costs $0.3/GB & it integrates perfectly with FlyingPress.

Gijo (FlyingPress developer) recommends using Cloudflare + BunnyCDN for some solid reasons like better cache hit ratio, traffic routing & geo-replication feature for better performance.

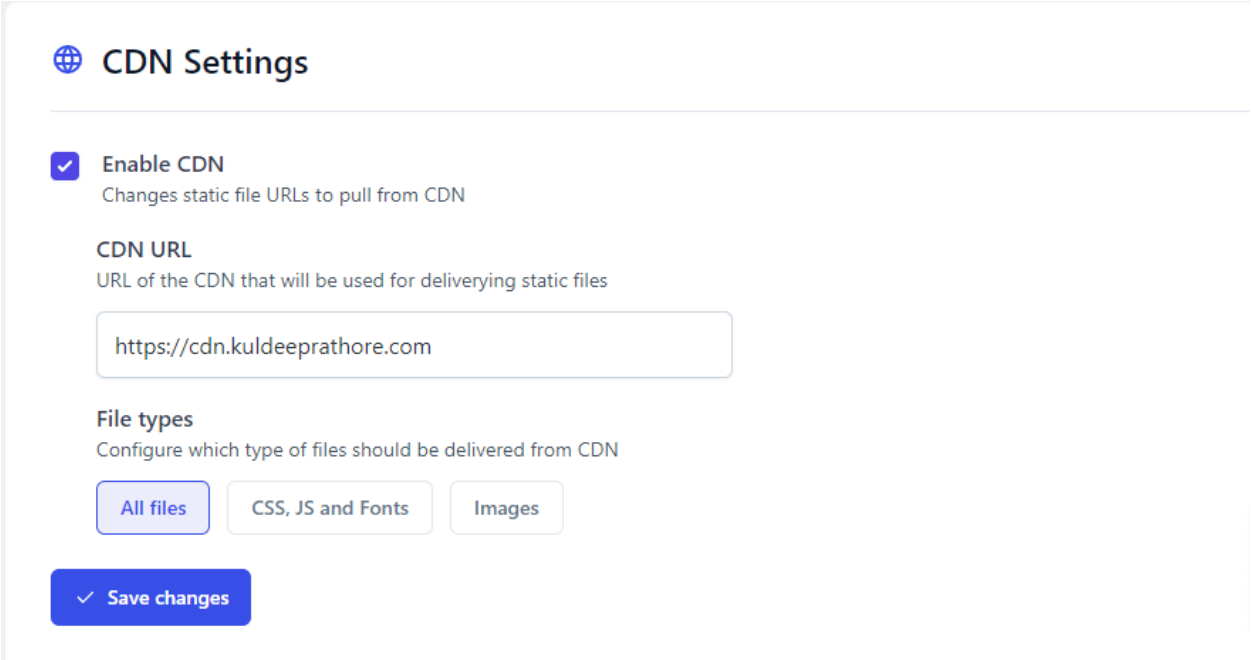
Using FlyingCDN, you can reap the benefits of BunnyCDN at lower costs.

Here's a quick guide on setting up FlyingCDN.
Once you're done with connecting a custom domain in FlyingCDN, simply come back to the CDN settings in FlyingPress & enable the following settings...

➔ **Enable CDN:** Enable

➔ **CDN URL:** Specify the URL of the CDN & then select the types of files to serve

➔ **File Types:** All Files




## 9. Bloat Settings

*Imp Note: Many of the bloat removal settings overlap with the Perfmatters plugin, so if you're using both, make sure to enable the duplicate settings in only one plugin.*

➔ **Remove Google Fonts:** Disabled – Although it improves performance, only use it if you're in the process of using system fonts or custom fonts and need them disabled. Otherwise, it removes Google Fonts from the site. (I use Google fonts & hence I've turned off this setting)

➔ **Disable XML-RPC:** Enable – This helps you with improving site performance + security (used to publish content from mobile).

➔ **Disable RSS feed: Disabled** – Only turn this on if your website doesn't have a blog

➔ **Disable Block editor CSS: Disabled** – If you're using Gutenberg, leave it off. If you're using a page builder instead, you can turn this on. This prevents a CSS file from loading across your site & improves performance.

➔ **Disable oEmbeds: Enable** – when you paste a URL into your WordPress editor, this loads a pretty preview (from YouTube, Facebook, Tweets, etc). If it's not needed, turn it on.

➔ **Disable Emojis: Enable** – removes a JavaScript file needed for emojis (it's better to use Unicode instead).

➔ **Disable WP Cron: Disabled** – this disables wp-cron which runs scheduled tasks. It's better to keep it off or you can use an external cron job instead.

➔ **Disable jQuery Migrate: Disabled** – if you're not using a page builder, turn it on. I use Elementor and hence, for compatibility, I've kept it off. Perfmatters has docs on dealing with jQuery in the script manager, defer, delay, and preload settings.

➔ **Disable Dashicons: Enable** – prevents admin icons from loading a CSS file on the front end.

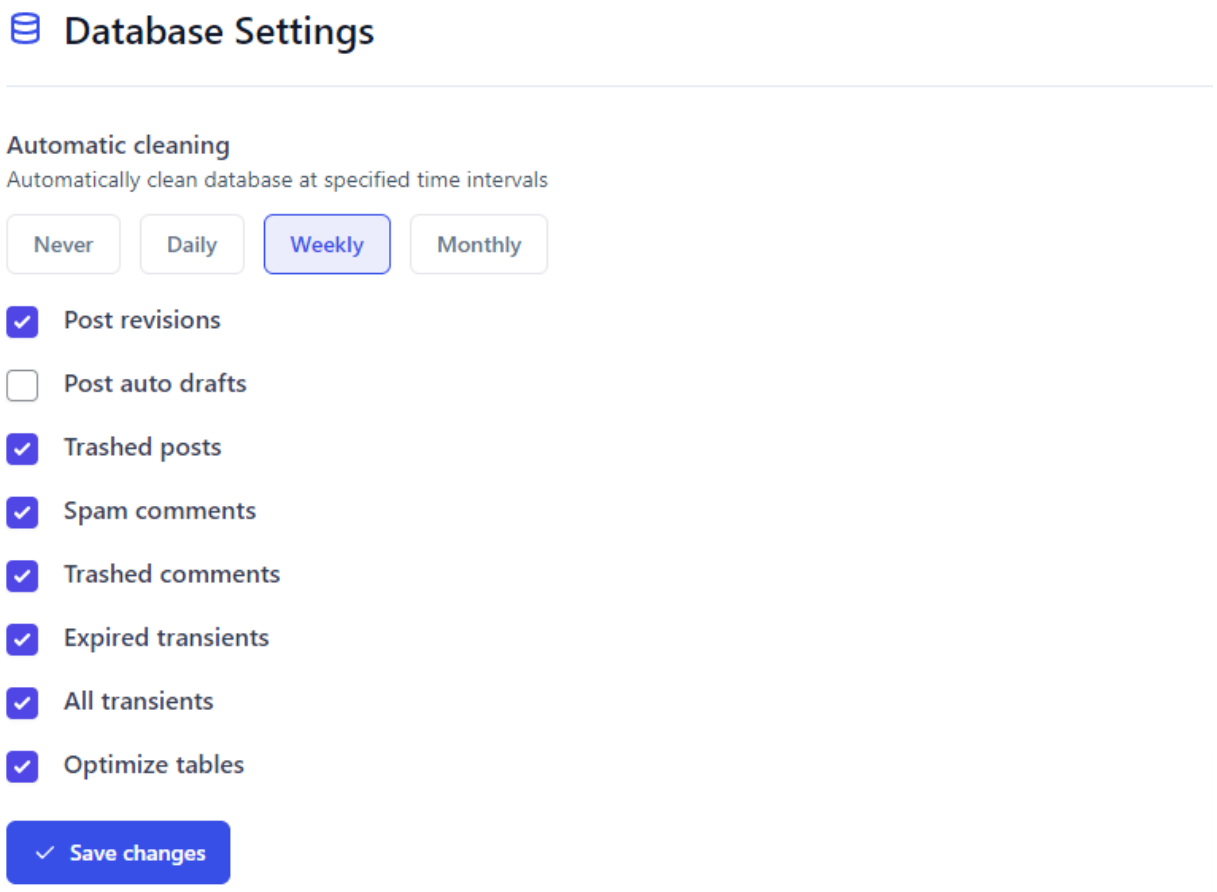
➔ **Control Post Revisions: Enable** & set Limit post revision to 5 or 10 – this gives you enough revisions where if you need a backup, you have a few to choose from, but still stops them from adding too much bloat. If you don't want post revisions, simply "disable revisions".

➔ **Control Heartbeat:** Enable only while editing posts for the 60s – this is also recommended by Perfmatters (especially if using page builders) to disable it unless you're editing content.

## 10. Database Settings

The FlyingPress Database section allows you to clean up your database and also gives you the option to schedule automatic cleaning which will optimize your site and reduce database bloat.

Here are the simple database settings...



➔ **Automatic Cleaning: Weekly** – You can choose how frequently you want your database cleaned up and optimized by FlyingPress. I recommend setting it to Weekly, so your database stays optimized every week.

➔ **Post Revisions: Disabled** – After officially publishing a post or page, revisions are the older versions of your post or page that are saved. I don't use post revisions, so I keep this setting enabled but for most, disabling will help you get your old versions easily.

➔ **Post Auto Drafts: Disabled** – When creating or editing a post or page, automatic drafts get saved of your content along the way so that you don't lose your work. Keep it disabled.

➔ **Trashed Posts: Enabled** – These are posts that you've deleted and put in the trash.

➔ **Spam Comments: Enabled**
➔ **Trashed Comments: Enabled**
➔ **Expired Transients: Enabled**
➔ **All Transients: Enabled**

➔ **Optimize Tables: Enabled** – This will optimize the tables in your database, so they are running efficiently without any bloat.

---

That's it!

You've successfully configured the best settings for FlyingPress & I hope it helps you boost your site performance & pass Google's Core Web Vitals.

Make sure to keep a constant check on your site (in incognito mode) while configuring in order to avoid breaking your site.

Lastly, if this guide helps you improve your Core Web Vital scores, then do share your scores with me.

Also, if you need any assistance, drop me a line & I'll be happy to assist you & help you as much as I can.

Until then, keep those scores high.

Cheers,
Kuldeep

Got an Elementor Website?
Check out the Best Elementor Resources

Want to reach out to me?
Email here: hello@kuldeeprathore.com