Website Development Checklist

(WordPress)

This document serves as a general checklist for the development process of WordPress based web applications, the main goal being to develop websites in an efficient manner and most importantly to develop websites that deal with common problems related to web development proactively and not reactively. This document is meant for the Primary Developer as well as any individuals interacting with the website on a development or post-live basis, for example: Designers, Copywriters, Marketing/Sales personnel, and Administrators.

# Pre-Development

This section outlines steps, configurations, plugin installs, etc, that should be done before the development of the website commences.

To reduce the complexity of the WordPress setup and minimize bloat on the website, where adding a server configuration, PHP function or WordPress hook can suffice, these methods will be preferred over installing an additional WordPress plugin.

## Establish account access

Declare which users have access to specific parts of the web server (cPanel, WP Admin panel, SSH access, FTP access, etc) for accountability.

## cPanel Account setup

Create the cPanel account for the domain on which the new website will be installed.

### Username conventions

The username will usually be the part of domain immediately preceding the TLD. For example, ***mywebsite.com*** will have the username ***mywebsite***.

If there are multiple domains which will cause a username conflict, example, ***mywebsite.com*** and ***mywebsite.info*** then omit the dot and use the full domain name. For example, ***mywebsite.info*** will have the username ***mywebsiteinfo***. If possible, add the alternate domain as an addon domain to the other cPanel account.

**Separate users with single domains**

|  |  |
| --- | --- |
| **Username** | **Primary Domain** |
| mywebsite | mywebsite.com |
| mywebsiteinfo | mywebsite.info |

**One user with multiple domains** (Ideal)

|  |  |  |
| --- | --- | --- |
| **Username** | **Primary Domain** | **Addon Domain** |
| mywebsite | mywebsite.com | mywebsite.info |

### Account Package

The relevant package should be chosen based on the support level of the client and expected usage on the server.

|  |  |
| --- | --- |
| **Package** | **Expected Server Use** |
| Thin | Less than 1 GB disk usage  Low bandwidth requirements  *Example: Simple 1 to 5-page websites* |
| Light | More than 1 GB but less than 5 GB disk usage  Low to moderate bandwidth requirements  *Example: WishList eCommerce websites or sites where clients upload lots of photos* |
| Standard | More than 1 GB but less than 5 GB disk usage  High bandwidth requirements  *Example: Heavy traffic business sites* |
| Heavy | More than 5 GB but less than 10 GB disk usage  High bandwidth requirements  *Example: Very special purpose websites* |

**(Package names not yet finalised)**

## DNS Setup for Website and Email Availability

Ideally, the nameservers of the domains should be changed to the nameservers of the cPanel server. This way, the server manages the DNS zone files which will typically have the required settings to ***(1)*** *make the domain available on the web*, and ***(2)*** *prevent emails sent by the server from ending up in SPAM/Junk mail by enabling both SPF and DKIM records*.

This will also make creating subdomains and adding additional DNS records easier from the developer's perspective.

|  |  |
| --- | --- |
| **Nameserver 1** | ns7.dns.com or ns7.<CompanyInitials>.com |
| **Nameserver 2** | ns8.dns.com or ns8.<CompanyInitials>.com |

In the case where the client is unable or unwilling to change their nameservers, then at the very least, the client must setup the following DNS records on their domain.

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Name** | **Value** | **Reason** |
| A | @ or <domain.name> | 52.191.167.168 | So that the website is accessible |
| CNAME | www | @ or <domain.name> | DNS simplification: If the IP has to be changed in the future only one A record needs changing to set the IP for both www and non-www |
| TXT | @ or <domain.name> | v=spf1 a mx ip4:52.191.167.168 ~all | SPF record reduces likelihood of email from the server ending up in SPAM/Junk |

## Custom Server Configurations

For consistency among developers and ease of identifying problems related to PHP, all configurations should be done in the .user.ini file located in the website’s root folder, while all rewrite rules are to be done in a .htaccess file located in the website’s root folder.

### PHP Version Settings

Use the cPanel MultiPHP manager to select the PHP version for the website

The default PHP version is **7.2**

### PHP INI Settings

Using the cPanel MultiPHP INI Editor will ensure that PHP settings are stored in the correct places.

These are the recommended settings:

|  |  |  |
| --- | --- | --- |
| **PHP Directive** | **Information** | **Setting** |
| display\_errors | This determines whether errors should be printed to the screen as part of the output or if they should be hidden from the user. | Disabled |
| max\_execution\_time | This sets the maximum time in seconds a script is allowed to run before it is terminated by the parser. This helps prevent poorly written scripts from tying up the server. The default setting is 90. | 180 |
| max\_input\_time | This sets the maximum time in seconds a script is allowed to parse input data, like POST, GET and file uploads. | 120 |
| max\_input\_vars | This sets the maximum number of input variables allowed per request and can be used to deter denial of service attacks involving hash collisions on the input variable names. | 5000 |
| memory\_limit | This sets the maximum amount of memory in bytes that a script is allowed to allocate. This helps prevent poorly written scripts for eating up all available memory on a server. Note that to have no memory limit, set this directive to -1. | 256M |
| post\_max\_size | Sets max size of post data allowed. This setting also affects file upload. To upload large files, this value must be larger than upload\_max\_filesize. Generally speaking, memory\_limit should be larger than post\_max\_size. | 10M |
| session.gc\_maxlifetime | This specifies the number of seconds after which data will be seen as "garbage" and potentially cleaned up. | 1440 |
| session.save\_path | session.save\_path defines the argument which is passed to the save handler. If you choose the default files handler, this is the path where the files are created.  **Make sure this matches the MultiPHP configured version** | /var/cpanel/php/sessions/ea-php72 |
| upload\_max\_filesize | The maximum size of an uploaded file. | 8M |
| zlib.output\_compression | Whether to transparently compress pages. If this option is set to "On" in php.ini or the Apache configuration, pages are compressed if the browser sends an "Accept-Encoding: gzip" or "deflate" header. | Disabled |

Settings in BLUE are the values recommended for BeTheme

Settings in RED are those for which optimum values are not yet determined

## Optimization – Management Dashboard

Install the WPMU Dev Dashboard plugin then use it to install the following plugins:

1. Smush Pro
2. Hummingbird Pro
3. Snapshot Pro

## Optimization – Page Caching

Caching dynamic PHP content and serving it as static html files can reduce CPU usage and bandwidth usage of the server.

These optimizations can be done automatically with one of the following setups.

|  |  |  |
| --- | --- | --- |
| **Setup** | **Using** | **Settings** |
| WPMU Dev (preferred) | Hummingbird Pro plugin | Page Caching activated  RSS Caching activated  (Hummingbird Pro page caching not yet tested with geo-targeting) |
| Free | WP Super Cache plugin | ***Under “Advanced”***  **Caching:** Enable  **Delivery Method:** Simple  **Miscellaneous:**   * Disable caching for logged in visitors * Cache rebuild * 304 Browser caching * Extra homepage checks * Cache HTTP headers with page content (only in the case of dynamic content such as geo-targeted content *– also requires Country Caching For WP Super Cache plugin*) * Late init (only in the case of dynamic content such as geo-targeted content *– also requires Country Caching For WP Super Cache plugin*) |

## Optimization – Browser Caching

Capitalising on the cache in visitors' browsers reduces both bandwidth use on the server and second-time-round load time of static website assets.

These optimizations can be done with one of the following setups.

|  |  |  |
| --- | --- | --- |
| **Setup** | **Using** | **Settings** |
| WPMU Dev (preferred) | Hummingbird Pro plugin | Browser Caching activated |
| Free | .htaccess | # BEGIN WP-HUMMINGBIRD-CACHING  # The directives (lines) between `BEGIN WP-HUMMINGBIRD-CACHING` and `END WP-HUMMINGBIRD-CACHING` are  # dynamically generated, and should only be modified via WordPress filters.  # Any changes to the directives between these markers will be overwritten.  <IfModule mod\_expires.c>  ExpiresActive On  ExpiresDefault A0  <FilesMatch "\.(txt|xml|js)$">  ExpiresDefault A31536000  </FilesMatch>  <FilesMatch "\.(css)$">  ExpiresDefault A31536000  </FilesMatch>  <FilesMatch "\.(flv|ico|pdf|avi|mov|ppt|doc|mp3|wmv|wav|mp4|m4v|ogg|webm|aac|eot|ttf|otf|woff|svg)$">  ExpiresDefault A31536000  </FilesMatch>  <FilesMatch "\.(jpg|jpeg|png|gif|swf|webp)$">  ExpiresDefault A31536000  </FilesMatch>  </IfModule>  <IfModule mod\_headers.c>  <FilesMatch "\.(txt|xml|js)$">  Header set Cache-Control "max-age=31536000"  </FilesMatch>  <FilesMatch "\.(css)$">  Header set Cache-Control "max-age=31536000"  </FilesMatch>  <FilesMatch "\.(flv|ico|pdf|avi|mov|ppt|doc|mp3|wmv|wav|mp4|m4v|ogg|webm|aac|eot|ttf|otf|woff|svg)$">  Header set Cache-Control "max-age=31536000"  </FilesMatch>  <FilesMatch "\.(jpg|jpeg|png|gif|swf|webp)$">  Header set Cache-Control "max-age=31536000"  </FilesMatch>  </IfModule>  # END WP-HUMMINGBIRD-CACHING |

## Optimization – CSS/JavaScript

Optimizing CSS/JavaScript to prevent render blocking will help web pages to start loading sooner.

Minifying these files will allow them to load faster and improve page load times.

These optimizations can be done with one of the following setups.

|  |  |  |
| --- | --- | --- |
| **Setup** | **Using** | **Settings** |
| WPMU Dev (preferred) | Hummingbird Pro plugin | Asset Optimization **activated**  **But disable WPMU DEV CDN setting** |
| Free | Autoptimize plugin | ***Under “Extra”***   * Combine and preload in head * Remove WordPress core emojis * Remove query strings   ***Under “JS, CSS & HTML”***   * Optimize JavaScript * Aggregate JS-files * Optimize CSS * Aggregate CSS-files * Also aggregate inline CSS * Exclude CSS from Autoptimize: wp-content/cache/, wp-content/uploads/, admin-bar.min.css, dashicons.min.css * Optimize HTML code * Save aggregated script/css as static files * Minify excluded CSS and JS |

## Optimization – GZIP Compression

Compressing files being served will reduce bandwidth consumption.

These optimizations can be done with one of the following setups.

|  |  |  |
| --- | --- | --- |
| **Setup** | **Using** | **Settings** |
| WPMU Dev (preferred) | Hummingbird Pro plugin | **Gzip compression activated**  **Choose Server type as Apache** (not NGINX) **before clicking activate** |
| Free | .htaccess | # BEGIN WP-HUMMINGBIRD-GZIP  # The directives (lines) between `BEGIN WP-HUMMINGBIRD-GZIP` and `END WP-HUMMINGBIRD-GZIP` are  # dynamically generated, and should only be modified via WordPress filters.  # Any changes to the directives between these markers will be overwritten.  <IfModule mod\_deflate.c>  SetOutputFilter DEFLATE  <IfModule mod\_setenvif.c>  <IfModule mod\_headers.c>  SetEnvIfNoCase ^(Accept-EncodXng|X-cept-Encoding|X{15}|~{15}|-{15})$ ^((gzip|deflate)\s\*,?\s\*)+|[X~-]{4,13}$ HAVE\_Accept-Encoding  RequestHeader append Accept-Encoding "gzip,deflate" env=HAVE\_Accept-Encoding  </IfModule>  </IfModule>  <IfModule mod\_filter.c>  AddOutputFilterByType DEFLATE "application/atom+xml" \  "application/javascript" \  "application/json" \  "application/ld+json" \  "application/manifest+json" \  "application/rdf+xml" \  "application/rss+xml" \  "application/schema+json" \  "application/vnd.geo+json" \  "application/vnd.ms-fontobject" \  "application/x-font-ttf" \  "application/x-font-opentype" \  "application/x-font-truetype" \  "application/x-javascript" \  "application/x-web-app-manifest+json" \  "application/xhtml+xml" \  "application/xml" \  "font/eot" \  "font/opentype" \  "font/otf" \  "image/bmp" \  "image/svg+xml" \  "image/vnd.microsoft.icon" \  "image/x-icon" \  "text/cache-manifest" \  "text/css" \  "text/html" \  "text/javascript" \  "text/plain" \  "text/vcard" \  "text/vnd.rim.location.xloc" \  "text/vtt" \  "text/x-component" \  "text/x-cross-domain-policy" \  "text/xml"  </IfModule>  <IfModule mod\_mime.c>  AddEncoding gzip svgz  </IfModule>  </IfModule>  # END WP-HUMMINGBIRD-GZIP |

## Optimization – Images

Higher resolution images look better but negatively affect SEO via slow load times, while lower resolution images load faster but negatively affect user experience.

Enabling lazy loading allows the rest of the page to load while images are still being downloaded for display, making the page appear sooner to the visitor.

Image resolution, file size, and lazy loading optimizations can be done automatically using one of the following setups:

|  |  |  |
| --- | --- | --- |
| **Setup** | **Using** | **Settings** |
| WPMU Dev (preferred) | Smush Pro | ***Under “Bulk Smush”***   * Automatically compress images on upload * Super-Smush my images * Strip my image metadata * Resize full size images 1920 × 1920 * Smush my original full size images * **Disable** store a copy of my small originals * **Disable** Auto-convert PNGs to JPEGS   ***Under “Lazy Load”***   * **Activated** * All settings can be left as default   ***Under “CDN”***   * **Activated** * Serve background images from the CDN * Enable automatic resizing of my images * **Do NOT enable** WebP compression |
| Free | Resize Image After Upload (ShortPixel) | * Enable re-sizing: **yes** * Max image dimensions: **1920 × 1920** * JPEG Compression: **100** * Force JPEG re-compression: **no** * Convert PNG to JPEG: **NO** |
| reSmush.it Image Optimizer plugin | * Image quality: **88** * Optimize on upload: ✔️ * Enable statistics: ✔️ * Do not preserve backups: ✔️ |
| Autoptimize plugin | ***Under “Images”***   * Lazy-load images?: ✔️ |

## Set up Remote Backups

Adding remote backups before development starts helps the developer recover from critical plugin bugs (common issues with WordPress) that critically impact the website as well as user mistakes.

|  |  |  |
| --- | --- | --- |
| **Setup** | **Using** | **Settings** |
| WPMU Dev (preferred) | Snapshot plugin | ***Destinations***   * Dropbox configured using the Company’s account credentials   ***Snapshots***   * Destination: **Dropbox** * **Include common files** * **Archive** (not Mirror/Sync) * **Include all database tables** * **Run daily, weekly or monthly**   + **Weekly** on **Sunday** at **Midnight**   + Keep a certain number of latest snapshots and remove the oldest: **48**   + Keep a local copy of a certain number of latest snapshots only: **1** |
| Free | Not applicable |  |

## Security - Hardening

### .htaccess Settings

Basic server hardening can be achieved by adding the following lines to the .htaccess file in the root directory

|  |  |
| --- | --- |
| **Lines** | **Purpose** |
| Options -Indexes | Prevents users from viewing the “index” or list of files present in a directory that does not have an index.html or index.php file |

## Security - SSL Certificate Installation

Set up SSL and enforce HTTPS protocol security to eliminate Mixed Protocol errors at any point throughout development.

AutoSSL on cPanel can be configured to use either Sectigo (cPanel/WHM default) or Let's Encrypt SSL certificates.

**The domain must first be pointed to the server by either changing the nameservers or editing the DNS records.** This is a prerequisite for AutoSSL to be able to automatically install SSL certificates for the domain.

### .htaccess Settings

HTTPS enforcing can be achieved without the need for an SSL WordPress plugin. **A valid SSL certificate must first be installed for these settings to work.**

|  |  |
| --- | --- |
| **Lines** | **Purpose** |
| Header always set Content-Security-Policy "upgrade-insecure-requests;" | Eliminates Mixed Content issues in the browser  *Ref:* [*https://www.tbs-certificates.co.uk/FAQ/en/upgrade-insecure-requests.html*](https://www.tbs-certificates.co.uk/FAQ/en/upgrade-insecure-requests.html) |
| # BEGIN rlrssslReallySimpleSSL rsssl\_version[3.3.2]  <IfModule mod\_rewrite.c>  RewriteEngine on  RewriteCond %{HTTPS} !=on [NC]  RewriteCond %{REQUEST\_URI} !^/\.well-known/acme-challenge/  RewriteRule ^(.\*)$ https://%{HTTP\_HOST}/$1 [R=301,L]  </IfModule>  # END rlrssslReallySimpleSSL | Redirects all requests to the server to use HTTPS.  (this is a copy of lines added by the Really Simple SSL plugin but will work independently of the plugin) |

***Note:*** *If the site for any reason isn't being developed on its primary domain, but on a temporary domain with the intention to migrate it after development, then* ***SSL setup*** *needs to be repeated* ***after*** *either the website is transferred, the nameservers are changed, and/or DNS records are changed.*

## Email Account Setup

### SMTP Address

Best practice is for WordPress to use a valid email address via SMTP to send mail rather than its default email function.

**An email account can be configured in the cPanel account for this purpose.** To make proper use of this email address, ensure that either the domain's nameservers match the ones on the server, or the SPF record is added as a DNS TXT record. Otherwise, these emails will most likely end up in SPAM/Junk.

Recommended values are as follows:

|  |  |  |
| --- | --- | --- |
| **Address** | **Password** | **Notes** |
| noreply@<domain.name> | *Auto-generate this in cPanel* | Save the password in a secure location. |

WordPress must then be configured to use this address instead of its default mail function.

**A valid SSL certificate must first be installed for these settings to work.**

|  |  |  |
| --- | --- | --- |
| **Setup** | **Using** | **Settings** |
| Free | Post SMTP plugin | ***Under “Account”***   * **Type:** SMTP * **Mailer Type:** PostSMTP * **Outgoing Mail Server Hostname:** <domain.name> * **Outgoing Mail Server Port:** 465 * **Envelope-From Email Address:** noreply@<domain.name> * **Security:** SMTPS * **Authentication:** Plain / Login * **Username:** noreply@<domain.name> * **Password:** <the one autogenerated in cPanel>   ***Under “Message”***   * **Email Address:** noreply@<domain.name> * ✔️ Prevent plugins and themes from changing this * **Reply-To:** <client's email address>   ***Under “Logging”***   * **Enable Logging:** Yes * **Maximum Log Entries:** 50 * **Maximum Transcript Size:** 128 |

## Security - Change the Default Login URL

As an added layer of security, avoid the use of the standard “/wp-admin/” URL to login to the WP Admin panel, since this address is easy to attack for hackers/bots who simply would need to type “<yourdomain>.com/**wp-admin**/” to begin brute forcing your account.

|  |  |  |
| --- | --- | --- |
| **Setup** | **Using** | **Settings** |
| WPMU Dev | Defender Pro plugin | ***Login screen masking***   * <OptionalText> example: c7-hello or aegean-prime |
| Free (preferred) | WPS Hide Login plugin | ***Settings - Login url***   * <OptionalText> example: c7-hello or aegean-prime |

***Note*** *from Marlon****:*** *“I usually do the login URL step last.”*

# Development

This section does NOT outline how a website should be designed, instead it describes what should be taken into account when making decisions about the design and functionality of the website.

## Follow SSL Protocol

While adding links, files, and accepting input on a website, take into account that linking insecure content to the website compromises the SSL Certificate. Thus, only content secured over HTTPS should be added or accessed, and linking files (even locally) over HTTP should be redirected to HTTPS or prohibited.

## Google reCaptcha

Spammers and bots are a common occurrence on the internet but even more so on WordPress. Use Invisible reCaptcha v3 on Contact forms, Email subscription forms, and Comments as they are created.

***Note:*** *If the site for any reason isn't being developed on its primary domain, but on a temporary domain with the intention to migrate it after development, then* ***Google reCaptcha setup*** *needs to be done* ***after*** *either the website is transferred, the nameservers are changed, and/or DNS records are changed.*

## Follow SEO Principles

Pages, Posts, and Content added to the site should be added with one of the secondary goals being to increase SEO, this includes:

1. Using standard naming schemes for pages and posts
2. Practicing proper outlining structure using page headers

Header 1

Header 2

Header 3

Header 2

Header 3

* 1. Only one <h1> tag should be present on a page as the page title.
  2. Headers tags (<h1>, <h2>, etc) should **not** be used for text formatting (getting the desired size or weight) if it means breaking the outline structure. In this case, use CSS styles instead.

Instead of

<h5>My Special Text</h5>

Try

<p style="font-size:16px;font-weight:700;">My Special Text</p>

1. Using Breadcrumbs
2. Backlinking pages and content
3. Avoiding duplicate content
4. Avoiding query variables in URLs where possible
5. Forcing HTTPS redirects

## Plugin Optimization

WordPress websites are known for slower loading times than custom built websites, however this problem has a lot to do with the inefficient use of plugins. Thus, having multiple plugins with similar functionality active should be avoided.

For example, if one plugin is installed to assist in SEO, then no other plugins should be installed with similar features.

If multiple plugins are installed to serve different purposes but have overlapping features, carefully configure each plugin, disabling features which are provided by another plugin. Where possible, use a single plugin which consolidates all of these features.

## Image Optimization

Higher resolution images look better but negatively affect SEO via slow load times, while lower resolution images load faster but negatively affect user experience. Standard pixel sizes for images and acceptable Reduced sizes, in cases where optimization is needed, are outlined here. These also help to create a Content Management and Content Quality standard across development.

Sizes are outlined below:

***Note:*** *Images that do not fit the 16:9 ratio (wide screen aspect ratio), for example, images that are 1:1 (square), should use the higher pixel value (1920) for Exports, and the lower value (720) for Reduction.*

1. Designer Exports (Standard)
   1. Posters/Intro Sliders/Wallpapers etc. should be 1920 × 1080
   2. Posts/Profile/Product etc should be 1000px
   3. Logos/Emblems etc should be 500px
2. Developer Upload (Reduced)
   1. Posters/Intro Sliders/Wallpapers etc. should be 1280 × 720
   2. Posts/Profile/Product etc should be 500px
   3. Logos/Emblems etc should be 250px
   4. Generated Thumbnail Sizes
      1. Large: 500px
      2. Medium: 250px
      3. Small: 100px

***(this Sub-section pending removal as it is covered under the “Pre-Development” section)***

# Post-Live Checklist

This section serves as a final checklist that needs to be completed after a website is live on its domain. Some features and configurations by nature require a website to be completed in development as well as operating on its primary domain before they are implemented.

## Google Analytics

This should be implemented on all websites, aside from being used for marketing, the data gathered here is very useful for doing server wide maintenance, such as deciding how much resources to allocate to a specific website based on its user traffic.

## Dead Links Check

Performing a dead link check helps ensure that assets which were added to the website early in development have not become corrupt or missing, and won't throw 404 errors. In addition, check links to external websites, social media pages, etc., to make sure that they are also not broken and are still served over a secure protocol.

## Responsive UI Test

Websites are built responsive by default however there might be some small undesirable UI element that was missed during development, for example, an icon on one page being misaligned on mobile devices, while working perfectly on PCs and tablets.

## Mail Server Test

Checking the client’s/website’s email address for valid delivery and forwarding of incoming email. Use the Client’s email to perform validation tests on the Contact Form, Subscriber List, SignUp form, etc

## Web Crawler and Open Graph Share Test

Perform search engine crawl tests using Google Search Console to verify that web crawlers can access all parts of the website that should be public. Also check Open Graph tags to make sure that social media sites can access needed metadata when links to the site are shared.