Technical Specifications:

1. General Information

Project: Optimizing the performance of the WordPress website

Objectives:

- Reducing website response time
- Speeding up image loading
- Improving performance and user experience

2. Task 1: Configuring WordPress caching

2.1. Goal

Implement multi-level caching to reduce server response time and speed up page loading.

2.2. Caching requirements

2.2.1. Page caching

- Install and configure a caching plugin (WP Rocket, W3 Total Cache, or similar)
- Configure caching for static HTML pages
- Set cache lifetime: 4 hours for the home page, 8 hours for internal pages
- Implement automatic cache clearing when content is updated

2.2.2. Object Caching

- Configure object caching via Redis or Memcached
- Install and configure the appropriate plugin (Redis Object Cache, etc.)
- Configure database query caching
- Set TTL for objects: 2 hours

2.2.3. Browser caching

- Configure Cache-Control headers for static resources:
  - CSS/JS files: 1 year
  - Images: 6 months
  - Fonts: 1 year
- Implement static file versioning for cache invalidation

2.2.4. Minification and concatenation

Enable minification of HTML, CSS, and JavaScript

Configure concatenation of CSS and JS files

Implement deferred loading of JavaScript (defer/async)

2.3. Server integration

  Configure Nginx/Apache to support caching

  Implement gzip compression for text resources

  Configure ETag headers

2.4. Acceptance criteria

  Server response time (TTFB) less than 200 ms

  Google PageSpeed Insights performance score of at least 90/100

  Cache is cleared correctly when content is updated

  Caching works for authorized users (with exceptions)

3. Task 2: Implement image loading in CDN

3.1. Goal

Integrate CDN for hosting and delivering images, reducing the load on the main server and speeding up the loading of media files.

3.2. Requirements for CDN integration

3.2.1. Selecting and configuring a CDN provider

Select a CDN provider (Cloudflare, BunnyCDN, CloudFront, or similar)

  Configure the CDN zone for the website domain

  Configure an SSL certificate for the CDN

3.2.2. Integration with WordPress

  Install and configure a plugin for working with CDN (WP Offload Media, CDN Enabler, or similar)

  Configure automatic uploading of new images to CDN

  Implement synchronization of existing media files

  Ensure support for the WebP format (with fallback)

3.2.3. Image optimization

  Configure automatic image optimization upon upload:

    Lossless compression

    Resize to a maximum width of 1920px

    Automatic preview generation

  Implement lazy loading for images

  Configure adaptive images for different devices

3.2.4. Content delivery configuration

  Configure image caching policy in CDN:

    TTL for images: 1 month

    Cache-Control headers

HTTP/2 support

Implement hotlinking protection

Configure watermarking if necessary

3.3. Migration of existing images

Create a script to migrate existing media files to CDN

Ensure redirection of old URLs to CDN addresses

Preserve the structure of the WordPress media library

3.4. Acceptance criteria

All new images are automatically uploaded to CDN

Existing images are transferred to CDN

Image URLs in content are replaced with CDN addresses

Image loading speed is improved by 50%

HTTPS is supported for all resources

4. Technical requirements

4.1. Compatibility

WordPress version 5.8+

PHP 7.4+

Multisite support

Compatibility with existing plugins and themes

4.2. Security

Image privacy preservation

Protection against direct access to originals

Secure data transfer to CDN

4.3. Monitoring and analytics

Integration with Google Analytics

Performance monitoring configuration

Caching and CDN error logging

5. Implementation stages

Stage 1: Preparation

Analysis of the current configuration

Website backup

Selection and configuration of tools

Stage 2: Caching configuration

Installation and configuration of caching plugins

Configuring server caching

Testing cache performance

Stage 3: CDN integration

Configuring a CDN account

Integration with WordPress

Migrating existing images

Stage 4: Testing and optimization

Performance testing

Configuring additional optimizations

Documenting changes

6. Documentation

Upon completion of the work, the following must be provided:

Technical documentation on settings

Instructions for the site administrator

Recommendations for further support

7. Budget and deadlines

Completion time: 30 business days