

Implementation Guide: Standardizing Blog Templates

This guide provides step-by-step instructions for implementing the standardized blog template and CSS across all blog posts.

Step 1: File Structure Setup

Create a consistent file structure for your blog section:

```
/messeriadjusting/
├─ css/
│   └─ blog.css          # Centralized blog CSS
├─ img/                  # Image directory
├─ js/
│   └─ main.js           # Main JavaScript file
├─ index.html            # Homepage
├─ about.html
├─ services.html
├─ claims.html
├─ contact.html
├─ blog.html             # Blog index page
└─ blog/                 # Individual blog posts
    ├─ insurance-appraisal-blog.html
    ├─ 5-things-insurance-companies-dont-want-you-to-know.html
    ├─ 2025-hurricane-season-forecast.html
    └─ ...
```

Step 2: CSS Implementation

1. Create the standardized CSS file:

- Create `/css/blog.css` using the provided standardized CSS
- This centralizes all blog-specific styles in one file for easier maintenance

2. Update the main styles.css:

- Ensure your main CSS file includes appropriate variable definitions:

CSS

```
:root {  
  --primary-color: #0066cc; /* Adjust to match your branding */  
  --heading-color: #333333;  
  --text-color: #555555;  
  --text-light: #777777;  
  --border-color: #eeeeee;  
  --light-color: #f8f9fa;  
  --box-shadow: 0 5px 15px rgba(0, 0, 0, 0.05);  
  --border-radius: 8px;  
  --transition: all 0.3s ease;  
}
```

3. Remove inline styles:

- Transfer all inline styles from blog posts to the centralized blog.css file
- This improves maintainability and page load performance

Step 3: Updating Existing Blog Posts

For each blog post, follow these steps:

1. Link to CSS File:

- Add the link to the centralized blog.css file:

html

```
<link rel="stylesheet" href="../css/blog.css">
```

- Remove any page-specific inline CSS

2. Update HTML Structure:

- Replace the main content container with the standardized two-column layout:

html

```
<main class="blog-page">
  <div class="container">
    <div class="blog-container">
      <div class="blog-content">
        <!-- Content here -->
      </div>
      <aside class="blog-sidebar">
        <!-- Sidebar here -->
      </aside>
    </div>
  </div>
</main>
```

3. Content Structure:

- Ensure consistent heading structure (H1 for title, H2 for sections)
- Implement standard meta information (date, author, category)
- Add missing elements (testimonials, CTAs) using the template

4. Sidebar Implementation:

- Add the complete sidebar structure as shown in the template
- Update recent posts, categories, and other dynamic content

Step 4: Testing and Validation

For each updated page, perform the following checks:

1. Cross-Browser Testing:

- Test in Chrome, Firefox, Safari, and Edge
- Verify display on both Windows and Mac OS

2. Responsive Design:

- Test on various screen sizes (desktop, tablet, mobile)
- Confirm that breakpoints work correctly (991px and 768px)

3. Standardization Checks:

- Confirm all pages have:
 - Proper two-column layout
 - Consistently sized featured images (400px height)
 - Complete sidebar with all widgets

- Author box with photo
- Tags/related topics
- Testimonial
- CTA
- Related posts

4. **Validation:**

- Run HTML validation check at <https://validator.w3.org/>
- Run CSS validation check at <https://jigsaw.w3.org/css-validator/>

Step 5: Common Issues & Solutions

Problem: Images not displaying correctly

Solution: Check image paths. Use relative paths (../img/filename.jpg)

Problem: Layout breaking on mobile

Solution: Check for any fixed widths that should be responsive. Ensure media queries are being applied.

Problem: Featured post image too large/small

Solution: Check that the .featured-post-img class is applied and has the correct height (400px)

Problem: Sidebar displaying below content on desktop

Solution: Verify the grid-template-columns property is set correctly and not being overridden

Problem: Inconsistent spacing

Solution: Use the standardized margin values from the CSS file; avoid inline margins

Step 6: Rails Implementation (For Future Development)

When implementing in Rails:

1. Create a blog layout template in `app/views/layouts/blog.html.erb`
2. Create partials for reusable components:
 - `app/views/shared/_blog_sidebar.html.erb`
 - `app/views/shared/_blog_header.html.erb`
 - `app/views/shared/_blog_footer.html.erb`
3. Store blog CSS in the asset pipeline: `app/assets/stylesheets/blog.scss`

4. Set up routes to handle blog categories and posts
5. Implement controllers to handle blog functionality

Conclusion

Following this standardization guide will ensure all blog posts share a consistent structure, styling, and user experience. This standardization will improve maintainability, reader engagement, and SEO performance.