# CC Spec: Phase J.0.5 - Extract Inline CSS/JS from Prompt Detail Page

Date: December 18, 2025

Phase: J.0.5 (Prompt Detail Page Redesign - Code Extraction)

Priority: HIGH

Estimated Time: 20-30 minutes

## CRITICAL: READ FIRST

BEFORE starting this task, Claude Code MUST:

- Read docs/CC_COMMUNICATION_PROTOCOL.md - Contains mandatory agent usage requirements
- Read this entire specification - Don't skip sections
- Use required agents - Minimum 3 agents appropriate for the task
- Report agent usage - Include ratings and findings in completion summary

This is non-negotiable per the project's CC Communication Protocol.

## OVERVIEW

### *Task Name*

Extract Inline CSS and JavaScript from Prompt Detail Page

### *Context*

Phase J of the PromptFinder project focuses on redesigning the prompt detail page. The baseline analysis (Phase 0) identified that the current template has:

- 63 lines of inline CSS in {% block extra_head %}
- 325 lines of inline JavaScript in {% block extras %}

This results in a 6/10 maintainability score. Extracting these to external files will:

- Enable browser caching (performance improvement)
- Improve code maintainability (8+/10 target)
- Make future Phase J work easier (layout changes)
- Follow project's established patterns (external CSS/JS)

### *What Came Before*

- Phase 0: Baseline analysis complete (documented in docs/PROMPT_DETAIL_ANALYSIS.md)
- Navigation fixes committed (search filtering, upload URL)
- All 59 tests passing, CI green

## OBJECTIVES

### *Primary Goal*

Move all inline CSS and JavaScript from prompt_detail.html to external files without breaking any functionality.

### *Success Criteria*

- All inline CSS moved to static/css/pages/prompt-detail.css
- All inline JS moved to static/js/prompt-detail.js
- Template updated with proper {% static %} includes
- Like button AJAX still works
- Report modal AJAX still works
- Copy to clipboard works (with fallback)
- Comment expand/collapse works (if present)
- Delete confirmation modal works
- All 8+ JavaScript functions operational
- No console errors in browser
- All 59 tests still pass
- Page renders correctly on desktop and mobile

## PROBLEM ANALYSIS

### *Current State*

The file prompts/templates/prompts/prompt_detail.html (818 lines) contains:

Inline CSS (lines 28-91, approximately 63 lines):

- Page-specific styles in {% block extra_head %}
- Styles for media container, comments, actions, tags
- Responsive media queries for mobile

Inline JS (lines 491-818, approximately 325 lines):

- toggleLike() - AJAX like/unlike functionality
- copyPromptText() - Clipboard API with fallback
- Report modal submission handler

- Delete confirmation modal
- Comment toggle functionality
- DOMContentLoaded event handlers
- CSRF token handling
- Bootstrap modal interactions

### *Issues/Challenges*

- Inline code cannot be browser-cached separately
- Large file (818 lines) hard to navigate
- JS functions tightly coupled to template
- Difficult to minify for production
- Code duplication risk across templates

### *Root Cause*

Original development prioritized speed over maintainability. Now that the page is stable, we can properly organize the code.

## SOLUTION

### *Approach*

- Create external CSS file with all inline styles
- Create external JS file with all inline scripts
- Ensure JS properly handles CSRF tokens
- Update template to reference external files
- Test all interactive features

### *Implementation Details*

#### Step 1: Create CSS File

File: static/css/pages/prompt-detail.css

Extract all content from {% block extra_head %} that is within <style> tags.

Structure the CSS file with clear sections:

```
/* ==========================================================================
   PROMPT DETAIL PAGE STYLES
   File: static/css/pages/prompt-detail.css
   Description: Styles specific to the prompt detail page
   ========================================================================== */

/* --------------------------------------------------------------------------
   1. Media Container
   -------------------------------------------------------------------------- */
```

```
/* ---------------------------------------------------------------------------
   2. Action Buttons
   --------------------------------------------------------------------------- */

/* ---------------------------------------------------------------------------
   3. Comments Section
   --------------------------------------------------------------------------- */

/* ---------------------------------------------------------------------------
   4. Tags
   --------------------------------------------------------------------------- */

/* ---------------------------------------------------------------------------
   5. Responsive / Mobile
   --------------------------------------------------------------------------- */
```

#### Step 2: Create JS File

File: static/js/prompt-detail.js

Extract all content from {% block extras %} that is within <script> tags.

Important JS Patterns to Preserve:

- CSRF Token Handling:

```
// Get CSRF token from cookie or hidden input
function getCsrfToken() {
    const cookieValue = document.cookie
        .split('; ')
        .find(row => row.startsWith('csrftoken='))
        ?.split('=')[1];
    return cookieValue || document.querySelector('[name=csrfmiddlewaretoken]')?.value;
}
```

- Like Toggle Function:

Must make proper AJAX POST request with CSRF token.

- Copy to Clipboard:

Must have fallback for older browsers.

- Report Modal:

Must submit via AJAX and show success/error feedback.

- Bootstrap Modal Integration:

Preserve all Bootstrap modal event handlers.

Structure the JS file:

```
/* ===========================================================================
   PROMPT DETAIL PAGE JAVASCRIPT
   File: static/js/prompt-detail.js
   Description: Interactive functionality for prompt detail page
   =========================================================================== */

/* ---------------------------------------------------------------------------
   1. CSRF Token Handling
   --------------------------------------------------------------------------- */
```

```
/* ------------------------------------------------------------------------
   2. Like Toggle
   ------------------------------------------------------------------------ */

/* ------------------------------------------------------------------------
   3. Copy to Clipboard
   ------------------------------------------------------------------------ */

/* ------------------------------------------------------------------------
   4. Report Modal
   ------------------------------------------------------------------------ */

/* ------------------------------------------------------------------------
   5. Delete Confirmation
   ------------------------------------------------------------------------ */

/* ------------------------------------------------------------------------
   6. Comment Functions (if any)
   ------------------------------------------------------------------------ */

/* ------------------------------------------------------------------------
   7. DOMContentLoaded Initialization
   ------------------------------------------------------------------------ */
```

#### Step 3: Update Template

Remove inline CSS: Delete the <style>...</style> block from {% block extra_head %}

Add CSS include: In {% block extra_head %}:

```
{% block extra_head %}
<link rel="stylesheet" href="{% static 'css/pages/prompt-detail.css' %}">
{% endblock %}
```

Remove inline JS: Delete the <script>...</script> block from {% block extras %}

Add JS include: In {% block extras %}:

```
{% block extras %}
<script src="{% static 'js/prompt-detail.js' %}"></script>
{% endblock %}
```

Ensure {% load static %} is at the top of the template (should already be present).

## FILES TO CREATE

### *File 1: static/css/pages/prompt-detail.css*

Purpose: External stylesheet for prompt detail page

Directory: Create /static/css/pages/ directory if it doesn't exist

Content: Extract all CSS from lines 28-91 of prompt_detail.html

### *File 2: static/js/prompt-detail.js*

Purpose: External JavaScript for prompt detail page interactivity

Directory: Confirm /static/js/ directory exists

Content: Extract all JavaScript from lines 491-818 of prompt_detail.html

## FILES TO MODIFY

### *File 1: prompts/templates/prompts/prompt_detail.html*

Current State: 818 lines with inline CSS/JS

Changes Needed:

- Replace inline <style> in {% block extra_head %} with <link> to external CSS
- Replace inline <script> in {% block extras %} with <script src> to external JS
- Keep all template logic, HTML structure, and Django template tags intact

After Changes: Template should be ~430 lines (818 - 63 - 325)

## AGENT REQUIREMENTS

MANDATORY: Use wshobson/agents during implementation

Per docs/CC_COMMUNICATION_PROTOCOL.md, CC must use appropriate agents for this task.

### *Required Agents (Minimum 3)*

1. @frontend-developer
    - Task: CSS and JavaScript extraction and organization
    - Focus: Proper file structure, browser compatibility, caching considerations
    - Rating requirement: 8+/10
2. @code-reviewer
    - Task: Verify no functionality is broken during extraction
    - Focus: All JS functions work, no missing code, clean extraction
    - Rating requirement: 8+/10
3. @django-expert
    - Task: Verify Django template integration

- Focus: {% static %} tags, template block structure, CSRF handling
- Rating requirement: 8+/10

### *Agent Reporting Format*

At the end of implementation, CC must report:

```
■ AGENT USAGE REPORT:

Agents Consulted:
1. @frontend-developer - [Rating/10] - [Brief findings]
2. @code-reviewer - [Rating/10] - [Brief findings]
3. @django-expert - [Rating/10] - [Brief findings]

Why These Agents:
[Brief explanation]

Agent Feedback Summary:
[Key findings and any concerns]
```

## TESTING CHECKLIST

After implementation, verify ALL of the following:

### *Functionality Tests (Manual Browser Testing Required)*

- [ ] Page loads without errors
- [ ] Featured image displays correctly
- [ ] Video displays with autoplay/loop (if video prompt)
- [ ] Like button toggles (AJAX) - check Network tab
- [ ] Like count updates without page reload
- [ ] Copy prompt button works - text copies to clipboard
- [ ] Copy button shows feedback (tooltip or animation)
- [ ] Report modal opens
- [ ] Report modal submits via AJAX
- [ ] Report modal shows success message
- [ ] Delete modal opens (owner only)
- [ ] Delete confirmation works
- [ ] Tags are clickable
- [ ] Author link works
- [ ] Comments display correctly
- [ ] Comment form works (if authenticated)
- [ ] Mobile layout works (resize browser or use DevTools)

### *Console/Network Checks*

- [ ] No JavaScript errors in browser console

[ ] No 404 errors for CSS/JS files

[ ] External CSS file loads (check Network tab)

[ ] External JS file loads (check Network tab)

[ ] AJAX requests have proper CSRF tokens

### *Django Tests*

```
python manage.py test
```

[ ] All 59 tests pass

### *Visual Checks*

[ ] Page looks identical to before extraction

[ ] No layout shifts or broken styles

[ ] Responsive design still works

## COMPLETION REPORT FORMAT

Upon completion, provide a report in this format:

```
■ TASK COMPLETE: Phase J.0.5 - Extract Inline CSS/JS

Summary:
■■■■■■■■■■
[Brief description of what was done]

Files Created:
■■■■■■■■■■■■■■■■■■
1. static/css/pages/prompt-detail.css (XX lines)
   - [List main sections]

2. static/js/prompt-detail.js (XX lines)
   - [List main functions]

Files Modified:
■■■■■■■■■■■■■■■■■■■
1. prompts/templates/prompts/prompt_detail.html
   - Before: 818 lines
   - After: XXX lines
   - Changes: [Summary of changes]

Testing Performed:
■■■■■■■■■■■■■■■■■■■■■■■
■ [Test 1]
■ [Test 2]
...

■ AGENT USAGE REPORT:
[Full agent report as specified above]

Status: ■ READY FOR TESTING BY USER
Next: User should verify in browser, then proceed to Phase 1 (Layout Restructure)
```

## IMPORTANT NOTES

- DO NOT modify any HTML structure - Only extract CSS/JS
- DO NOT change any functionality - This is a pure extraction task
- Preserve all Django template variables - Any {{ variable }} in JS must be handled
- If JS uses Django template variables, you may need to pass them as data attributes or keep a small inline script to initialize variables

### *Handling Django Template Variables in JS*

If the inline JS contains Django template variables like:

```
const promptSlug = '{{ prompt.slug }}';
const csrfToken = '{{ csrf_token }}';
```

Solution: Add data attributes to a container element:

```
<div id="prompt-detail-container"
     data-slug="{{ prompt.slug }}"
     data-csrf="{{ csrf_token }}">
```

Then in external JS:

```
const container = document.getElementById('prompt-detail-container');
const promptSlug = container.dataset.slug;
const csrfToken = container.dataset.csrf;
```

## GIT COMMIT MESSAGE FORMAT

After successful completion, suggest commit message:

```
refactor(prompt-detail): Extract inline CSS/JS to external files

Phase J.0.5 - Preparation for prompt detail page redesign

Files created:
- static/css/pages/prompt-detail.css (63 lines)
- static/js/prompt-detail.js (325 lines)

Changes:
- Extracted inline CSS from {% block extra_head %}
- Extracted inline JS from {% block extras %}
- Updated template to use {% static %} includes
- Preserved all functionality (like, copy, report, delete)

Maintainability improvement: 6/10 → 7/10
Target: 8+/10 after Phase 1 (layout restructure)

All 59 tests passing.
```

END OF SPECIFICATION