art10m's BookStack Hacks

<head>

```
<!-- ============================================================================
     BOOKSTACK CUSTOM HEAD CONFIGURATION

     This file contains custom styles, scripts, and integrations for BookStack:
     - Typography and layout adjustments (zoom, colors, spacing)
     - CodeMirror 6 line wrapping for code blocks
     - Markdown-it configuration (soft line breaks)
     - MathJax integration for LaTeX math rendering
     - Mermaid diagram rendering with theme synchronization
     - Light/Dark mode toggle button (works for both guests and logged-in users)
     ============================================================================ -->

<!-- ============================================================================
     CONSOLIDATED STYLES
     ============================================================================ -->
<style>
  /* ----------------------------------------------------------------------------
     DETAILS ELEMENT SPACING
     Ensures proper bottom margin for the last child inside <details> elements
     ---------------------------------------------------------------------------- */
  .page-content details > *:last-child {
    margin-bottom: .2em;
  }

  /* ----------------------------------------------------------------------------
     BLOCKQUOTE ADJUSTMENTS
     - Adds consistent vertical spacing for first/last children
     - Disables overflow scrolling to prevent unwanted scrollbars
     ---------------------------------------------------------------------------- */
  .content-wrap blockquote > :last-child {
```

```
  margin-bottom: .3em;
}

.content-wrap blockquote > :first-child {
  margin-top: .3em;
}

.content-wrap blockquote {
  overflow: visible;
  overflow-x: visible;
  overflow-y: visible;
}

/* ------------------------------------------------------------------------
   PAGE CONTENT ZOOM AND TYPOGRAPHY
   - Applies 1.15x zoom to page content for better readability
   - Compensates heading size with 0.9x zoom to maintain visual hierarchy
   ------------------------------------------------------------------------ */
.page-content {
  zoom: 1.15;
}

.page-content h1,
.page-content h2,
.page-content h3,
.page-content h4,
.page-content h5,
.page-content h6 {
  zoom: .9;
}

/* ------------------------------------------------------------------------
   COLOR SCHEME: LIGHT MODE
   - Dark text on light background for optimal contrast
   - Slightly muted headings for visual hierarchy
   ------------------------------------------------------------------------ */
html .page-content {
  color: hsl(0 0% 10%);
}
```

```
html .page-content h1,
html .page-content h2,
html .page-content h3,
html .page-content h4,
html .page-content h5,
html .page-content h6 {
  color: hsl(0 0% 30%);
}

/* ------------------------------------------------------------------------
   COLOR SCHEME: DARK MODE
   - Light text on dark background
   - Adjusted heading colors for dark theme
   - Muted gutter colors for CodeMirror editor
   ------------------------------------------------------------------------ */
html.dark-mode .page-content {
  color: hsl(0 0% 90%);
}

html.dark-mode .page-content h1,
html.dark-mode .page-content h2,
html.dark-mode .page-content h3,
html.dark-mode .page-content h4,
html.dark-mode .page-content h5,
html.dark-mode .page-content h6 {
  color: hsl(0 0% 70%);
}

html.dark-mode .co .cm-gutters {
  color: hsl(0 0% 33%);
}

/* ------------------------------------------------------------------------
   CODEMIRROR EDITOR STYLING
   Removes border-radius for a cleaner, squared appearance
   ------------------------------------------------------------------------ */
.page-content .cm-editor {
  border-radius: 0;
```

```
}

/* ------------------------------------------------------------------------
   MERMAID DIAGRAM CONTAINER
   - Dashed border for visual identification during development/editing
   - Centered layout with horizontal scroll for large diagrams
   ------------------------------------------------------------------------ */
.mermaid-container {
  border: 1px dashed #4238ff !important;
}

.mermaid {
  margin: 1em auto;
  overflow-x: auto;
  text-align: center;
}

.mermaid svg {
  max-width: 100%;
  height: auto;
  display: inline-block;
}

/* ------------------------------------------------------------------------
   THEME TOGGLE BUTTON (FIXED POSITION)
   - Positioned at bottom-left corner for easy access
   - Circular button with hover/active states
   - Semi-transparent by default, full opacity on hover
   ------------------------------------------------------------------------ */
.theme-toggle-fixed {
  position: fixed;
  bottom: 20px;
  left: 20px;
  z-index: 9999;
  background: var(--color-primary, #206ea7);
  border: none;
  border-radius: 50%;
  width: 44px;
  height: 44px;
```

```
    cursor: pointer;
    box-shadow: 0 2px 8px rgba(0, 0, 0, 0.25);
    display: flex;
    align-items: center;
    justify-content: center;
    transition: transform 0.2s ease, box-shadow 0.2s ease, opacity 0.2s ease;
    opacity: 0.7;
  }

  .theme-toggle-fixed:hover {
    transform: scale(1.1);
    box-shadow: 0 4px 12px rgba(0, 0, 0, 0.35);
    opacity: 1;
  }

  .theme-toggle-fixed:active {
    transform: scale(0.95);
  }

  .theme-toggle-fixed svg {
    width: 22px;
    height: 22px;
    fill: #ffffff;
  }
</style>

<!-- ============================================================================
     MATHJAX CONFIGURATION
     Enables LaTeX math rendering with $ for inline and $$ for display math
     ============================================================================ -->
<script>
  window.MathJax = {
    tex: {
      inlineMath: [['$', '$']],
      displayMath: [['$$', '$$']]
    }
  };
</script>
<script id="MathJax-script" async src="https://cdn.jsdelivr.net/npm/mathjax@4/tex-mml-
```

```
chtml.js"></script>

<!-- ============================================================================
     BOOKSTACK EVENT LISTENERS AND THEME TOGGLE
     - CodeMirror 6 line wrapping configuration
     - Markdown-it soft line break configuration
     - Light/Dark mode toggle for guests and logged-in users
     ============================================================================ -->
<script>
  (function () {
    'use strict';

    /* ============================================================================
       SHARED THEME STORAGE KEY
       Used by both the toggle button and Mermaid for consistent theme state
       ============================================================================ */
    var THEME_STORAGE_KEY = 'bookstack-guest-dark-mode';

    // Expose storage key globally for the Mermaid module to access
    window.BOOKSTACK_THEME_STORAGE_KEY = THEME_STORAGE_KEY;

    /* ============================================================================
       CODEMIRROR 6: LINE WRAPPING FOR CODE BLOCKS
       Listens for the CM6 pre-init event and enables line wrapping
       for content code blocks to prevent horizontal scrolling
       ============================================================================ */
    window.addEventListener('library-cm6::pre-init', function (event) {
      var detail = event.detail;
      var config = detail.editorViewConfig;
      var EditorView = detail.libEditorView;

      // Only apply line wrapping to content code blocks (not the main editor)
      if (detail.usage === 'content-code-block') {
        config.extensions.push(EditorView.lineWrapping);
      }
    });

    /* ============================================================================
       MARKDOWN-IT: SOFT LINE BREAKS
```

```
   Configures the Markdown editor to convert single newlines to <br> tags
   (GFM-style line breaks)
   ============================================================================ */
window.addEventListener('editor-markdown::setup', function (event) {
  event.detail.markdownIt.set({breaks: true});
});

/* ============================================================================
   LIGHT/DARK MODE TOGGLE BUTTON

   Features:
   - Works for both guests (localStorage) and logged-in users (server-side)
   - Applies saved preference immediately to prevent flash of wrong theme
   - Detects system color scheme preference as fallback for guests
   ============================================================================ */

/**
 * Checks if the guest has dark mode enabled based on localStorage
 * Falls back to system preference if no stored value exists
 * @returns {boolean} True if dark mode should be enabled
 */
function isGuestDarkModeEnabled() {
  var stored = localStorage.getItem(THEME_STORAGE_KEY);
  if (stored !== null) {
    return stored === 'true';
  }
  // Fallback: Check system color scheme preference
  return window.matchMedia && window.matchMedia('(prefers-color-scheme: dark)').matches;
}

/**
 * Determines if the current user is logged in
 * Checks for CSRF token AND user menu elements (both required)
 * @returns {boolean} True if user is logged in
 */
function isUserLoggedIn() {
  var csrfMeta = document.querySelector('meta[name="csrf-token"]');
  var hasToken = csrfMeta && csrfMeta.getAttribute('content');
  var hasUserMenu = document.querySelector('.dropdown-container [data-
```

```
shortcut="favourites_view"]')
        || document.querySelector('[href*="/logout"]');
      return !!(hasToken && hasUserMenu);
    }

    /**
     * Applies the theme by toggling the 'dark-mode' class on <html>
     * @param {boolean} isDark - Whether to enable dark mode
     */
    function applyTheme(isDark) {
      if (isDark) {
        document.documentElement.classList.add('dark-mode');
      } else {
        document.documentElement.classList.remove('dark-mode');
      }
    }

    // ---------------------------------------------------------------------------
    // IMMEDIATE THEME APPLICATION (before DOMContentLoaded)
    // Prevents flash of wrong theme by applying saved preference early
    // ---------------------------------------------------------------------------
    var currentlyDark = document.documentElement.classList.contains('dark-mode');
    var guestPref = localStorage.getItem(THEME_STORAGE_KEY);

    if (guestPref !== null) {
      if (!currentlyDark && guestPref === 'true') {
        document.documentElement.classList.add('dark-mode');
      } else if (guestPref === 'false' && currentlyDark) {
        document.documentElement.classList.remove('dark-mode');
      }
    }

    // ---------------------------------------------------------------------------
    // CREATE TOGGLE BUTTON (after DOM is ready)
    // ---------------------------------------------------------------------------
    document.addEventListener('DOMContentLoaded', function () {
      var isDarkMode = document.documentElement.classList.contains('dark-mode');
      var loggedIn = isUserLoggedIn();
```

```
      // SVG icons for sun (light mode) and moon (dark mode)
      var sunIcon = '<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="M20 15.31 23.31 12 20 8.69V4h-4.69L12 .69 8.69 4H4v4.69L.69 12 4 15.31V20h4.69L12 23.31 15.31 20H20zM12 18c-3.31 0-6-2.69-6-6s2.69-6 6-6 6 2.69 6 6-2.69 6-6 6"/></svg>';
      var moonIcon = '<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="M12 3a9 9 0 1 0 9 9c0-.46-.04-.92-.1-1.36a5.389 5.389 0 0 1-4.4 2.26 5.403 5.403 0 0 1-3.14-9.8c-.44-.06-.9-.1-1.36-.1z"/></svg>';

      // Create the toggle button element
      var button = document.createElement('button');
      button.className = 'theme-toggle-fixed';
      button.type = 'button';
      button.title = isDarkMode ? 'Activate Light Mode' : 'Activate Dark Mode';
      button.innerHTML = isDarkMode ? sunIcon : moonIcon;

      // Handle click: server-side for logged-in users, client-side for guests
      button.addEventListener('click', function () {
        if (loggedIn) {
          // Logged-in users: Submit form to BookStack's preference endpoint
          var csrfToken = '';
          var csrfMeta = document.querySelector('meta[name="csrf-token"]');
          if (csrfMeta) {
            csrfToken = csrfMeta.getAttribute('content') || '';
          }

          var form = document.createElement('form');
          form.method = 'POST';
          form.action = '/preferences/toggle-dark-mode';
          form.innerHTML =
            '<input type="hidden" name="_token" value="' + csrfToken + '">' +
            '<input type="hidden" name="_method" value="PATCH">' +
            '<input type="hidden" name="_return" value="' + window.location.href + '">';
          document.body.appendChild(form);
          form.submit();
        } else {
          // Guests: Toggle theme client-side and save to localStorage
          var nowDark = document.documentElement.classList.contains('dark-mode');
          var newMode = !nowDark;
```

```
          applyTheme(newMode);
          localStorage.setItem(THEME_STORAGE_KEY, newMode.toString());

          // Update button appearance
          button.innerHTML = newMode ? sunIcon : moonIcon;
          button.title = newMode ? 'Activate Light Mode' : 'Activate Dark Mode';
        }
      });

      document.body.appendChild(button);
    });
  })();
</script>

<!-- ============================================================================
     MERMAID DIAGRAM RENDERING (ES MODULE)

     Features:
     - Automatic detection of mermaid code blocks (multiple selector formats)
     - Theme synchronization with BookStack's dark/light mode
     - Uses shared localStorage key with theme toggle button
     - Theme changes apply on next page load (no live re-rendering)
     ============================================================================ -->
<script type="module">
  import mermaid from "https://cdn.jsdelivr.net/npm/mermaid@11/dist/mermaid.esm.min.mjs";

  // Use the shared storage key from the main script
  const STORAGE_KEY = window.BOOKSTACK_THEME_STORAGE_KEY || "bookstack-guest-dark-mode";

  /* ============================================================================
     THEME DETECTION UTILITIES
     ============================================================================ */

  /**
   * Checks if BookStack is currently in dark mode
   * @returns {boolean} True if 'dark-mode' class is present on <html>
   */
  function isBookStackDarkMode() {
    return document.documentElement.classList.contains("dark-mode");
```

```
}

/**
 * Retrieves the stored theme preference from localStorage
 * @returns {boolean|null} true = dark, false = light, null = not set
 */
function getStoredThemeIsDark() {
  const stored = localStorage.getItem(STORAGE_KEY);
  if (stored === null) return null;
  return stored === "true";
}

/**
 * Saves the theme preference to localStorage
 * @param {boolean} isDark - Whether dark mode is enabled
 */
function storeThemeIsDark(isDark) {
  localStorage.setItem(STORAGE_KEY, isDark ? "true" : "false");
}

/**
 * Determines the Mermaid theme based on current BookStack mode
 * Uses actual DOM state (dark-mode class) for accurate theme selection
 * @returns {string} Mermaid theme name ("dark" or "default")
 */
function getMermaidTheme() {
  return isBookStackDarkMode() ? "dark" : "default";
}

/* ============================================================================
   MERMAID INITIALIZATION
   Disabled auto-start to allow manual control over rendering
   ============================================================================ */
mermaid.initialize({
  startOnLoad: false,
  securityLevel: "strict",
  theme: getMermaidTheme()
});
```

```
/* ================================================================================
   DOM UTILITIES FOR MERMAID CODE BLOCKS
   ================================================================================ */

/**
 * Finds all elements containing Mermaid diagram source code
 * Supports multiple code block formats used by different editors
 * @returns {HTMLElement[]} Array of source elements
 */
function findMermaidSources() {
  const selectors = [
    "pre.mermaid",
    "pre > code.language-mermaid",
    "pre > code.lang-mermaid",
    "code.language-mermaid"
  ];

  const nodes = Array.from(document.querySelectorAll(selectors.join(",")));

  // Normalize to parent <pre> element when applicable
  return nodes.map(n =>
    (n.tagName.toLowerCase() === "code" && n.parentElement?.tagName.toLowerCase() === "pre")
      ? n.parentElement
      : n
  );
}

/**
 * Extracts the text content (Mermaid code) from a source element
 * @param {HTMLElement} node - The source element
 * @returns {string} The trimmed Mermaid code
 */
function extractCodeFromNode(node) {
  const codeEl = node.tagName.toLowerCase() === "pre"
    ? (node.querySelector("code") || node)
    : node;

  return (codeEl.textContent || "").trim();
}
```

```
/**
 * Replaces the original code block with a Mermaid container div
 * Initially hidden to prevent flash of unstyled content
 * @param {HTMLElement} originalNode - The original code block element
 * @param {string} code - The Mermaid diagram code
 * @returns {HTMLElement} The new container element
 */
function replaceWithContainer(originalNode, code) {
  const container = document.createElement("div");
  container.className = "mermaid";
  container.textContent = code;
  container.style.visibility = "hidden";

  originalNode.replaceWith(container);
  return container;
}

/* ============================================================================
   PAGE READY DETECTION
   ============================================================================ */

/**
 * Waits for the page to be fully loaded and rendered
 * Uses multiple techniques to ensure DOM is stable:
 * 1. Wait for 'load' event (or skip if already complete)
 * 2. Wait for two animation frames (ensures layout is calculated)
 * 3. Additional timeout for any async CSS/font loading
 */
async function waitForPageReady() {
  // Wait for window load event
  await new Promise(resolve => {
    if (document.readyState === "complete") return resolve();
    window.addEventListener("load", () => resolve(), {once: true});
  });

  // Wait for layout stabilization (two animation frames)
  await new Promise(r => requestAnimationFrame(() => requestAnimationFrame(r)));
```

```
  // Additional buffer for async resources
  await new Promise(r => setTimeout(r, 150));
}

/* ========================================================================
   MAIN RENDERING FUNCTION
   ======================================================================== */

/**
 * Main entry point: finds, transforms, and renders all Mermaid diagrams
 * Theme is determined once at render time and applied consistently
 * Theme changes will take effect on next page load
 */
async function renderAllMermaid() {
  const sources = findMermaidSources();

  // Even without diagrams, ensure theme state is stored for consistency
  if (!sources.length) {
    await waitForPageReady();
    storeThemeIsDark(isBookStackDarkMode());
    return;
  }

  // Transform code blocks into Mermaid containers
  const containers = [];
  for (const src of sources) {
    const code = extractCodeFromNode(src);
    if (!code) continue;

    const container = replaceWithContainer(src, code);
    // Store original code for potential future use
    container.setAttribute("data-original-code", code);
    containers.push(container);
  }

  await waitForPageReady();

  // Render all diagrams with the current theme
  for (const el of containers) {
```

```
      try {
        await mermaid.run({nodes: [el]});
        el.style.visibility = "visible";
      } catch (e) {
        el.style.visibility = "visible";
        console.error("Mermaid render failed for element:", el, e);
      }
    }

    // Store current theme state for next page load
    storeThemeIsDark(isBookStackDarkMode());
  }

  // Start rendering process
  renderAllMermaid();
</script>
```

functions.php

`/opt/bookstack/bookstack/www/themes/custom/functions.php`

```
<?php

// Import the ThemeEvents class which contains event constants for the theming system
use BookStack\Theming\ThemeEvents;
// Import the Theme facade to register event listeners
use BookStack\Facades\Theme;

/**
 * Register a listener for the CommonMark environment configuration event.
 * This hook is triggered when BookStack sets up the Markdown parser.
 *
 * CommonMark is the Markdown parsing library used by BookStack.
 */
Theme::listen(ThemeEvents::COMMONMARK_ENVIRONMENT_CONFIGURE, function ($environment) {

  // Merge custom configuration into the CommonMark environment
  $environment->mergeConfig([
```

```
    'renderer' => [
      // Convert soft line breaks (single newlines) into <br> HTML tags
      // By default, CommonMark ignores single newlines in Markdown.
      // This setting makes single line breaks visible in the rendered output,
      // which is useful for preserving line formatting in user content.
      'soft_break' => "<br>",
    ]
  ]);

  // Return the modified environment so other listeners can further customize it
  return $environment;
});
```

oEmbeds in `<head>`

```
<style>
  .auto-embed {
    position: relative;
    width: 100%;
    max-width: 100%;
    margin: 1rem 0;
    overflow: hidden;
    background: #000;
  }

  .auto-embed.video {
    aspect-ratio: 16/9;
  }

  .auto-embed.audio {
    aspect-ratio: 16/5;
  }

  .auto-embed.code {
    aspect-ratio: 4/3;
    min-height: 400px;
  }
```

```
  .auto-embed.square {
    aspect-ratio: 1/1;
  }

  .auto-embed iframe {
    position: absolute;
    inset: 0;
    width: 100%;
    height: 100%;
    border: 0;
  }
</style>

<script>
  document.addEventListener('DOMContentLoaded', () => {
    const providers = [
      // □□Video
      {
        regex: /youtu(?:be\.com\/watch\?v=|\.be\/)([\w-]+)/i,
        embed: id => `https://www.youtube-nocookie.com/embed/${id}`, type: 'video'
      },
      {
        regex: /vimeo\.com\/(\d+)/i,
        embed: id => `https://player.vimeo.com/video/${id}`, type: 'video'
      },
      {
        regex: /dailymotion\.com\/video\/([\w-]+)/i,
        embed: id => `https://www.dailymotion.com/embed/video/${id}`, type: 'video'
      },
      {
        regex: /twitch\.tv\/videos\/(\d+)/i,
        embed: id => `https://player.twitch.tv/?video=${id}&parent=${location.hostname}`,
type: 'video'
      },
      {
        regex: /twitch\.tv\/([\w-]+)$/i,
        embed: id => `https://player.twitch.tv/?channel=${id}&parent=${location.hostname}`,
type: 'video'
```

```
      },
      {
        regex: /loom\.com\/share\/([\w-]+)/i,
        embed: id => `https://www.loom.com/embed/${id}`, type: 'video'
      },
      {
        regex: /streamable\.com\/([\w-]+)/i,
        embed: id => `https://streamable.com/e/${id}`, type: 'video'
      },

      // □□Audio
      {
        regex: /open\.spotify\.com\/(track|album|playlist|episode)\/([\w-]+)/i,
        embed: (t, id) => `https://open.spotify.com/embed/${t}/${id}`, type: 'audio'
      },
      {
        regex: /soundcloud\.com\/([\w-]+\/[\w-]+)/i,
        embed: path =>
`https://w.soundcloud.com/player/?url=https://soundcloud.com/${path}&color=%23ff5500&auto_play
=false&hide_related=true&show_comments=false&show_user=true&show_reposts=false&show_teaser=fal
se`,
        type: 'audio'
      },

      // □□Code - CodePen

      // □ CodePen v2.0 Editor-Format (DIREKTES IFRAME!)
      {
        regex: /codepen\.io\/editor\/([\w-]+)\/pen\/([\w-]+)/i,
        embed: (user, penId) => `https://codepen.io/editor/${user}/embed/${penId}?default-
tab=html%2Cresult&theme-id=dark&editable=true`,
        type: 'code'
      },

      // □ Klassisches CodePen-Format
      {
        regex: /codepen\.io\/([\w-]+)\/(?:pen|full|details)\/([\w-]+)/i,
        embed: (user, penId) => `https://codepen.io/${user}/embed/${penId}?default-
tab=html%2Cresult&theme-id=dark&editable=true`,
```

```
        type: 'code'
      },

      // Andere Code-Plattformen
      {
        regex: /codesandbox\.io\/s\/([\w-]+)/i,
        embed: id => `https://codesandbox.io/embed/${id}`, type: 'code'
      },
      {
        regex: /stackblitz\.com\/edit\/([\w-]+)/i,
        embed: id => `https://stackblitz.com/edit/${id}?embed=1`, type: 'code'
      },
      {
        regex: /jsfiddle\.net\/([\w-]+\/[\w-]+)/i,
        embed: path => `https://jsfiddle.net/${path}/embedded/result,js,html,css/`, type:
'code'
      },
      {
        regex: /gist\.github\.com\/([\w-]+)\/([\w-]+)/i,
        embed: (u, id) => `data:text/html,<script
src="https://gist.github.com/${u}/${id}.js"><\/script>`,
        type: 'code'
      },

      // □□□Maps & Design
      {
        regex: /figma\.com\/(file|design)\/([\w-]+)/i,
        embed: (t, id) =>
`https://www.figma.com/embed?embed_host=bookstack&url=https://www.figma.com/${t}/${id}`,
        type: 'code'
      },
      {
        regex: /google\.com\/maps\/embed\?pb=([^&\s]+)/i,
        embed: pb => `https://www.google.com/maps/embed?pb=${pb}`, type: 'video'
      },
    ];

    document.querySelectorAll('p').forEach(p => {
      const text = p.textContent.trim();
```

```
      if (!/^https?:\/\/\S+$/i.test(text)) return;

      for (const {regex, embed, type} of providers) {
        const match = text.match(regex);
        if (!match) continue;

        const src = embed(...match.slice(1));
        p.outerHTML = `<div class="auto-embed ${type}"><iframe src="${src}" allowfullscreen
loading="lazy" allowtransparency="true"></iframe></div>`;
        return;
      }
    });
  });
</script>
```

And this must be added to `docker-compose.yml`

Under `environment`:

```
- ALLOWED_IFRAME_SOURCES=https://*.codepen.io https://codepen.io https://*.jsfiddle.net
https://jsfiddle.net https://*.codesandbox.io https://codesandbox.io https://open.spotify.com
https://gist.github.com https://*.youtube.com https://youtube.com https://www.youtube-
nocookie.com https://*.vimeo.com https://player.vimeo.com https://*.dailymotion.com
https://*.twitch.tv https://player.twitch.tv https://clips.twitch.tv https://*.tiktok.com
https://*.loom.com https://*.wistia.com https://fast.wistia.net https://streamable.com
https://*.vidyard.com https://rumble.com https://*.soundcloud.com https://w.soundcloud.com
https://embed.music.apple.com https://*.deezer.com https://widget.deezer.com
https://*.bandcamp.com https://*.mixcloud.com https://*.audiomack.com https://anchor.fm
https://*.transistor.fm https://*.stackblitz.com https://*.replit.com https://*.glitch.com
https://jsbin.com https://*.observablehq.com https://carbon.now.sh https://asciinema.org
https://platform.twitter.com https://*.instagram.com https://*.reddit.com
https://*.linkedin.com https://assets.pinterest.com https://*.figma.com https://*.canva.com
https://docs.google.com https://*.pitch.com https://prezi.com https://*.slideshare.net
https://speakerdeck.com https://maps.google.com https://*.openstreetmap.org
https://*.notion.so https://*.notion.site https://*.airtable.com https://*.typeform.com
https://form.typeform.com https://calendly.com https://*.miro.com https://excalidraw.com
https://*.diagrams.net https://whimsical.com https://*.lottiefiles.com https://lottie.host
```

```
https://*.sketchfab.com
```

https://www.youtube.com/watch?v=UclrVWafRAI

Revision #8
Created 2026-04-19 06:57:12 UTC by art10m
Updated 2026-04-19 22:05:26 UTC by art10m