

Headless CMS Architecture Overview This system is a headless CMS for creating memorial/missing person websites. Here's the architecture: Core Concept Users fill out a form about a missing person or memorial case, and the system generates a customizable website using that data. The sites are deployed to subdomains (e.g., johndoe.caseclosure.org). Key Components 1. Data Flow User Form → Case Data → Template System → Customizable Website → Deployed Site 2. Case Creator Modal (/components/CaseCreator/) - Multi-step form collecting case information - Steps: Basic Info → Details → Photos → Contact → Template Selection - Saves to backend API as a "case" object 3. Template System (/templates/beacon/) Pre-built website templates (currently "Beacon" template) Components: - Layout.jsx - Wrapper with navigation/footer - Home.jsx, About.jsx, etc. - Individual pages - Pages receive caseData and customizations as props 4. Editor/Customization System (/components/CaseCreator/views/CustomizationView/) - CustomizationView.jsx - Main editor interface - EditableSection.jsx - Wrapper making content editable (shows + buttons) - InlineEditorModal.jsx - TipTap rich text editor popup - TemplatePreview.jsx - Live preview of the template 5. Data Structure

```
caseData = {
  // Form data
  first_name, last_name, case_type,
  description, photos, detective_info, etc.

  // Template data
  template_data: {
    customizations: {
      hero_title: "Custom HTML content",
      quick_facts: "Edited content",
      // ... other section overrides
    }
  }
}
```

How Editing Works - Template pages use EditableSection components: {caseData.first_name} - In edit mode, blue + buttons appear on editable sections - Clicking + opens InlineEditorModal with TipTap editor - Saved changes go to customizations object under that section's ID - Template displays custom content if available, otherwise shows default from caseData Current State - Form data properly saves to backend - Template renders with case data - Editor route (/editor/:caseId) loads the CustomizationView - Plus buttons visible in edit mode - Working on: Making sections editable with TipTap when + buttons clicked File Structure /components/CaseCreator/ ■■■■ CaseCreator.jsx (main modal) ■■■■ EditorWrapper.jsx (loads case for editor) ■■■■ /views/CustomizationView/ ■■■■ CustomizationView.jsx ■■■■ /components/ ■■■■ EditableSection.jsx ■■■■ InlineEditorModal.jsx ■■■■ TemplatePreview.jsx ■■■■ /hooks/ ■■■■ useTemplateRegistry.js ■■■■ /services/ ■■■■ caseAPI.js /templates/beacon/ ■■■■ /src/pages/ ■■■■ Layout.jsx ■■■■ Home.jsx ■■■■ About.jsx The system allows non-technical users to create professional memorial websites by filling out a form and customizing pre-built templates through a visual editor.