



AnyforSoft  
Amplify Digital

# Practical Assignment: Drupal 11 Component Development

This task evaluates how candidates actually build and reason within Drupal 11's modern architecture—not just what they know, but how they apply it.

## Assignment Overview

- Task:** Create a standalone "Team Member Card" component using Drupal 11's Single Directory Components (SDC) architecture.
- Goal:** Assess hands-on understanding of component structure, configuration management, accessibility, and documentation clarity.

## Requirements

### 1. Component Structure

Create a proper SDC directory (*team-member-card/*) with:

- *team-member-card.component.yml*
- *team-member-card.twig*
- *team-member-card.css*
- *team-member-card.js* (optional)

The component must be fully self-contained (no external dependencies) and follow Drupal 11 SDC naming conventions and directory structure.

### 2. YAML Schema

Define a *.component.yml* schema with these props:

- *name* (string, required)
- *title* (string, required)
- *bio* (text, optional)
- *photo* (image, optional)
- *email* (string, optional)
- *social\_links* (array, optional)

Clearly specify each property's type and requirement status (required: true/false). Ensure the schema file passes validation using `drupal sdc:validate`.

### 3. Twig Template

Render props with semantic HTML5 markup: `<article>`, `<header>`, `<section>`. Apply Twig 3 best practices—escaping, filters, minimal logic. Handle edge cases: missing photo? Empty bio? Malformed social link? Show us you've thought about real-world content messiness.

### 4. CSS

Mobile-first responsive layout with consistent spacing and typography. Scope selectors to avoid leaking global styles; use BEM or component scoping. You may include CSS variables for colors or spacing for consistency. Ensure keyboard accessibility and visible focus styles.

### 5. Optional JavaScript

Add lightweight logic for email obfuscation or social link tracking. Keep it modular, unobtrusive. No global pollution; graceful degradation if JS fails.

### 6. Documentation

Provide a concise *README.md* describing:

- How to integrate this component into a content type display mode
- How to override or extend the component in a custom theme
- Steps to import configuration and register the component globally
- Include one screenshot or GIF of the rendered card

## Deliverables

Submit your work as a repository link containing the complete component directory (*/team-member-card/*), a *README.md* with setup instructions, and one visual proof (screenshot or GIF) of the rendered component.

## Evaluation Criteria (Scoring Matrix)

Area	Criteria	Score Range	What to Look For
<b>Structure &amp; Conventions</b>	Directory layout follows SDC pattern, clear naming	1–5	Files logically organized; no redundant assets
<b>YAML Schema</b>	Correct prop types, validation, required/optional logic	1–5	YAML passes <code>sdc:validate</code> ; uses consistent typing
<b>Twig Template</b>	Escaping, semantic HTML, accessibility	1–5	Proper alt text, readable hierarchy, clean syntax
<b>CSS</b>	Responsive, scoped, maintainable	1–5	Uses BEM or component scoping; accessible contrast
<b>Optional JS</b>	Lightweight, non-intrusive, purposeful	1–5	No global pollution; graceful degradation
<b>Configuration Export</b>	Complete and functional YAML	1–5	All config exports reproducible; imports cleanly
<b>Documentation</b>	Clarity, completeness, reusability	1–5	Explains setup and overrides clearly
<b>Edge-Case Handling</b>	Missing data, malformed URLs, long text	1–5	Shows resilience to real-world content
<b>Overall Presentation</b>	Professionalism and attention to detail	1–5	Follows best practices; no hardcoded paths

### Total Score:

40–43 points

- **35+** → Production-ready and well-documented
- **25–34** → Technically sound; needs minor improvement
- **15–24** → Basic understanding; limited polish
- **<15** → Weak grasp of SDC and modern Drupal workflows

## Time Allocation

**Experienced developer:** 2–3 hours

**Junior or learning candidate:** Up to 4 hours

If someone takes significantly longer, that's a signal too.

## Follow-Up Questions for Interview Discussion

Use these to extend the assignment into deeper conversation:

1. What's your reasoning behind the directory and schema structure you chose?
2. How would you make this component reusable across multiple Drupal sites?
3. Which Twig 3 features did you apply to improve maintainability or readability?
4. How did you ensure CSS styles remain isolated?
5. If this component needed multilingual support, what changes would you make?
6. What edge cases did you anticipate, and how did you handle them?
7. If given more time, what would you refactor or improve?

## What This Assignment Reveals

Mastery of Drupal 11 component architecture (SDC). Solid understanding of configuration management and Composer hygiene. Front-end responsibility: accessibility, performance, scoping. Awareness of real-world maintainability and documentation standards. Whether they build for the happy path only or handle actual production messiness.