**Comprehensive Vue 3 Component Creation Prompt for Existing Library**

I need you to create a new component for my Vue 3 component library that follows the same architecture, patterns, and code style already established. First, analyze my request to determine which type of component is needed:

**Component Type Decision Logic:**

* **Core Component**: If my request describes a simple UI element with no business logic, data fetching, or composition of other components (e.g., button, input, icon, badge).
* **Composite Component**: If my request describes a component that combines multiple core components with local state management but doesn't connect to external data sources (e.g., form, tabs, accordion, button group).
* **Smart Component**: If my request describes a component that requires data fetching, uses adapters, or integrates with APIs/services (e.g., data table, search, auto-complete, file uploader).

**Library Structure Overview:**

My component library follows a three-tier architecture with this structure:

vue-component-library/

├── src/

│ ├── adapters/ # API abstraction layer with adapter pattern

│ │ ├── httpAdapter.ts

│ │ ├── graphQLAdapter.ts

│ │ ├── mockAdapter.ts

│ │ └── dataTableAdapter.ts

│ ├── components/

│ │ ├── core/ # Simple UI components with no business logic

│ │ ├── composite/ # Combinations of core components

│ │ └── smart/ # Complex components using adapters

│ ├── composables/ # Reusable composition functions

│ │ ├── useTheme.ts

│ │ └── useI18n.ts

│ ├── styles/

│ │ ├── themes/ # Theme configuration

│ │ ├── tokens/ # Design variables (colors, spacing, typography)

│ │ └── utils/ # Mixins and utilities

│ ├── types/ # TypeScript definitions

│ └── index.ts # Main library entry point

├── tests/

│ └── unit/ # Unit tests for components

└── examples/ # Example implementations

**Implementation Requirements:**

1. **TypeScript Types**: Create in src/types/[componentName].ts
   * Define detailed interfaces for props, events, models
   * Follow existing patterns for type organization
   * Export types for library users
2. **Component Implementation**:
   * Use the appropriate folder based on component type
   * Follow Vue 3 Composition API with <script setup lang="ts"> syntax
   * Use existing composables like useTheme and useI18n where appropriate
   * Ensure components are accessible with proper ARIA attributes
   * Follow the established design system with CSS variables
3. **Adapter Pattern** (for smart components):
   * Create/extend adapters in the appropriate adapter file
   * Implement factory functions for creating adapters
   * Make components data-source agnostic
4. **Testing**:
   * Create tests/unit/[ComponentName].test.ts
   * Use Vitest and Vue Test Utils as in existing tests
   * Test all props, events, and user interactions
   * Test various states (loading, error, success, etc.)
5. **Documentation**:
   * Add detailed JSDoc comments
   * Document props, events, slots, and methods
   * Include usage examples and edge cases
6. **Library Integration**:
   * Update src/index.ts to export the new component
   * Add component to the Vue plugin installation function

**Coding Style Guidelines:**

* Use TypeScript throughout with proper typing
* Follow composition API best practices
* Prefer ref and computed over reactive
* Use destructuring and modern JS features
* Organize imports consistently
* Follow the existing naming conventions

**Example Component Creation Process:**

Please follow the example of existing components like VButton, VButtonGroup, and VDataTable, which showcase the implementation pattern for each tier. The smart components use the adapter pattern to abstract data fetching, while composite components combine and manage multiple core components.

**My Component Request:**

[I will describe the component I need here]

Based on my description, determine the appropriate tier, implement the complete component with all necessary files, and provide example usage showing how it integrates with the existing library.

**Universal Debugging Prompt for Vue Component Library**

I'm developing a Vue 3 component library with a three-tier architecture (core, composite, and smart components). Here's the context needed to help me solve the issue I'm facing:

**Project Structure**

My library is organized as follows:

vue-component-library/

├── src/

│ ├── adapters/ # API abstraction with adapter pattern

│ ├── components/

│ │ ├── core/ # Simple UI components

│ │ ├── composite/ # Combined components with local state

│ │ └── smart/ # Complex components with data fetching

│ ├── composables/ # Shared functionality (theme, i18n)

│ ├── styles/ # Design system and theme

│ ├── types/ # TypeScript definitions

│ └── index.ts # Main entry point

├── tests/ # Component tests

└── examples/ # Usage examples

**Technology Stack**

* Vue 3 with Composition API and <script setup> syntax
* TypeScript for type safety
* Vite for building
* Vitest for testing
* SCSS for styling with CSS variables for theming
* Adapter pattern for API abstraction

**Common Issues I Might Face**

* TypeScript type errors or missing type definitions
* Component communication problems (props/events)
* CSS scoping or styling inconsistencies
* Adapter implementation issues
* Testing component interactions
* Build configuration problems
* Dependency issues

**Current Problem**

[I will describe my specific issue here]

Please provide:

1. A diagnosis of what might be causing the problem
2. A detailed solution with code examples based on my project structure
3. Any relevant explanations about Vue 3, TypeScript, or the architecture pattern
4. Preventative measures to avoid similar issues in the future

If more information is needed, please let me know what would help diagnose the issue more effectively.