**CustomModalPopup Component – Detailed Requirements Specification**

**1. Advanced Features and Best Practices**

**1.1. Accessibility and Focus Management**

* **Focus Trap:**  
  Ensure keyboard navigation is trapped within the modal when open.
* **ARIA Attributes:**  
  Assign proper roles (role="dialog", aria-modal="true", accessible labels) for screen reader support.
* **Escape Key:**  
  Configurable option to close modal with the Escape key.

**1.2. Performance and Resource Management**

* **Lazy Load Content:**  
  Load modal content only when the modal opens, especially for heavy/complex content.
* **Destroy vs. Hide on Close:**  
  Configuration to either destroy modal content on close (default) or hide for quick re-opening.
* **Performance Optimization:**  
  Modal should only render active content and release all resources/events on close.

**1.3. Interaction and UX Enhancements**

* **Overlay Click/Close Options:**  
  Configurable: Close modal on backdrop click, ignore, or support double-click/long-press for mobile users.
* **Modal Queueing:**  
  Option to queue or prevent duplicate modals in notification-heavy use cases.
* **Confirmation/Pre-Close Guards:**  
  Support for async guards or confirmation callbacks before closing the modal (e.g., warn on unsaved changes).

**1.4. Layout and Background Handling**

* **Prevent Background Scroll:**  
  Prevent the background page from scrolling when the modal is open.
* **Scrollable Body with Sticky Header/Footer:**  
  Body content becomes scrollable if it exceeds viewport; header/footer remain fixed.
* **Customizable Backdrop:**  
  Configurable backdrop blur, color, opacity, or pattern.

**1.5. Integration and Placement**

* **Portal Support:**  
  Render modals at the top-level DOM (using Angular CDK Portal or equivalent) to avoid CSS stacking issues.
* **Dynamic Positioning:**  
  Modal can be centered or positioned near a trigger element (for context menus or popovers).
* **Z-Index/Stacking Order Management:**  
  Automatically manage stacking for multiple/nested modals so each appears above previous.
* **Global Modal Service:**  
  Expose a singleton service to open/close modals from anywhere in the app.
* **Accessibility Testing:**  
  Ensure the modal is fully accessible with keyboard, screen readers, and supports high-contrast themes.

**2. Core Component Structure and Configuration**

**2.1. Popup Structure**

* **Header**
  + Displays customizable title (left-aligned).
  + Minimize button (collapses modal content).
  + Close button (closes current modal instance).
  + Customizable header color.
* **Body**
  + Fully dynamic: Accepts any content (text, forms, Angular components, etc.).
* **Footer**
  + Optional, shown or hidden by configuration.
  + Accepts any user-defined controls (buttons, links, info, etc.).

**2.2. Responsiveness and Positioning**

* Fully responsive, looks/works well on all devices and screen sizes.
* By default, modal opens centered on the screen.
* **Custom Sizing:**
  + Width/height can be set via config.
  + If not provided, size should fit content.
  + If only one dimension given, the other should auto-fit to content.

**2.3. Multiple Instances and Nesting**

* Supports multiple modals open at once, including opening modals from inside modals (nested).
* Each modal instance is fully isolated and manages its own state/events.

**2.4. Draggable and Minimize/Maximize**

* Modal can be draggable via header (configurable).
* Minimize/maximize support to collapse/restore modal content.

**2.5. Close Behavior**

* Option to close modal via close button, overlay click, Escape key, or only by explicit action (fully configurable).

**2.6. Background Interaction**

* By default, shows semi-transparent backdrop and blocks interaction with page behind.
* Optionally allow background interaction (configurable per instance).

**3. Configuration Options & API**

* **Header:**
  + Title text
  + Custom color
  + Show/hide minimize/close buttons
* **Body:**
  + Content projection or service-driven injection
* **Footer:**
  + Show/hide
  + Custom content
* **Size:**
  + Width, height, min/max constraints
* **Positioning:**
  + Centered or near trigger element
* **Interactivity:**
  + Draggable (yes/no)
  + Close on overlay (yes/no)
  + Close on Escape (yes/no)
  + Allow background interaction (yes/no)
* **Accessibility:**
  + ARIA labels
  + Focus trap
* **Service API:**
  + Open, close, minimize, maximize, event callbacks
* **Other:**
  + Queueing, duplicate prevention, pre-close guards, etc.

**4. Additional Best Practices**

* **Mobile/Touch Support:**  
  Ensure draggable, scroll, and tap interactions work on mobile/touch devices.
* **Scroll/Resize Events:**  
  Modal should re-center or reposition itself if the window size changes.
* **Resource Cleanup:**  
  All event listeners and resources must be released when modal closes.
* **High Performance:**  
  Avoid unnecessary DOM operations and keep the modal lightweight.

**2. Core Features & Structure**

**2.1. Popup Structure**

* **Header**
  + Displays popup title (configurable per instance, left-aligned).
  + *Minimize button*: Collapses the modal content.
  + *Close button*: Closes the modal instance.
  + *Customizable color*: Header color can be set via configuration.
* **Body**
  + Fully dynamic: Accepts any content (simple text, forms, components, etc.).
* **Footer**
  + Optional: Shown or hidden based on configuration.
  + Accepts any user-defined controls (buttons, info, etc.).

**2.2. Responsiveness & Positioning**

* **Responsive Design:**  
  Must look and work well on all devices/screen sizes.
* **Initial Position:**  
  Opens centered on the viewport (unless otherwise specified).
* **Dynamic Sizing:**
  + Width and height can be set via config.
  + If not provided, adjust to fit content (“auto”).
  + If only width/height is provided, calculate the other dimension based on content.

**2.3. Interactivity**

* **Draggable:**
  + Optionally, popup can be moved by dragging the header.
  + Can be enabled/disabled per instance.
* **Minimize/Maximize:**
  + Can minimize popup to just the header; clicking maximize restores full view.
* **Multiple Instances:**
  + Can have multiple modals open simultaneously, including nested modals (open modal from within another).
* **Close Behavior:**
  + Option to close popup by clicking outside the modal (backdrop), controlled per instance.
  + Otherwise, background click is ignored.

**2.4. Background Handling**

* **Backdrop:**
  + By default, show a semi-transparent overlay (“gray out”) when modal is open, blocking interaction with background.
  + Optionally, allow user interaction with the background (if configured).

**3. Configurability & API**

* **Configurable via Input Properties/Service:**
  + Header text, color, buttons, icons.
  + Body content (projected or via service).
  + Footer presence and content.
  + Width, height, min/max constraints.
  + Draggability, minimize/maximize behavior.
  + Close-on-backdrop-click.
  + Allow background interaction.
* **Events & Outputs:**
  + onOpen, onClose, onMinimize, onMaximize, onDrag, etc.
  + Allow parent components to react to modal state changes.

**4. Accessibility & UX**

* **Keyboard Accessibility:**
  + Support keyboard navigation:
    - Focus trap within modal.
    - Escape key to close (if allowed).
  + ARIA roles for dialogs.
* **Focus Management:**
  + Focus moves to first focusable element in modal on open.
  + Focus returns to triggering element on close.
* **Screen Reader Support:**
  + ARIA labels, role="dialog", etc.
* **Tab Order:**
  + Controls inside modal are tab-navigable.

**5. Performance**

* **Efficient Rendering:**
  + Only render active modals.
  + Avoid global DOM pollution.
* **Easy Integration:**
  + Should require minimal code to use in new/existing components.
* **Service-Driven (Optional):**
  + Optionally expose a ModalService to open/close modals programmatically (for advanced use cases).

**6. Other Considerations**

* **Z-Index Management:**
  + Each modal instance appears above previous, handles stacking/nesting cleanly.
* **Animation (Optional):**
  + Add subtle open/close/minimize animations (CSS-based, lightweight).
* **Theming Support:**
  + Optionally support global or per-instance theming (light/dark, custom styles).

**7. Potential Missing Points or Enhancements**

* **Mobile Touch Support:**
  + Draggable modals must work on mobile/touch devices.
* **Prevent Background Scroll:**
  + Option to disable page scrolling when modal is open.
* **Maximum Modal Depth:**
  + Optionally set a limit on nesting to prevent infinite modal popups.
* **Loading/Busy State:**
  + Option to show spinner/loader for async content.
* **Destruction & Cleanup:**
  + Ensure resources are released on close/destroy.