

# ArcGIS Instant Apps Portfolio – CSS Element Documentation

This document describes the structure and styling of the **Portfolio** template used in ArcGIS Instant Apps. It combines observations from the published app at <https://aallc.maps.arcgis.com/apps/instant/portfolio/index.html?appid=a54718cc241b4322a715fc9f4cec90dc>, the un-minified CSS shipped with that app and the custom CSS provided by the user. The goal is to help you understand which CSS selectors control each part of the interface, how the pieces fit together and which aspects can be safely overridden in a custom style sheet.

## 1 Overview of the Portfolio template

The Portfolio template is a web component-based application built with the ArcGIS **Instant Apps** framework. It allows you to present a collection of web maps, scenes, apps or external content in a single interface. The layout can switch between **Tabbed**, **Carousel** and **Accordion** modes and it includes optional components such as a cover page, location search, description panel, bookmarks and autoplay controls.

Internally, the template is composed of a number of top-level custom elements:

Element	Role
<code>instant-apps-landing-page</code>	Optional cover/landing page shown before the main application. Contains the title, subtitle, hero image and <b>Enter</b> button.
<code>instant-apps-header</code>	Fixed header shown on the main page; houses the app title, logos, search, bookmarks and other controls. It exposes <code>::part(container)</code> and <code>::part(wrapper)</code> for limited styling and defines several CSS variables for colours and spacing.
<code>esri-portfolio-...</code> classes	A collection of classes prefixed <code>esri-portfolio-</code> make up the main body, navigation and content stage. These are defined in the app's compiled CSS. Examples include <code>esri-portfolio-tabbed</code> , <code>esri-portfolio-carousel</code> , <code>esri-portfolio-stage</code> , <code>esri-portfolio-description</code> , etc.
<code>calcite-button</code> , <code>calcite-card</code> , <code>calcite-icon</code>	Components from the Calcite Design System used for buttons, cards and icons. They are encapsulated in Shadow DOM and expose stylable parts via <code>::part(button)</code> , <code>::part(content)</code> , etc.

## 2 Cover page (instant-apps-landing-page)

When enabled, the cover page is rendered by the `instant-apps-landing-page` component. It fills the viewport and overlays a background image or video with a centred title and subtitle. The **Enter** button (or “Explore”) lets the user proceed to the main app. The native CSS for this component isn’t included in the compiled portfolio stylesheet, but you can override many properties through CSS variables and part selectors:

### 2.1 Container and background

The cover page container inherits `height:100%` and `width:100%`. It uses two variables to control text colour (`--instant-apps-landing-page-text-color`) and the fallback button colour. In the custom stylesheet provided by the user, these variables are overridden so that the text on the cover page is white and the button uses the brand red colour:

- `--instant-apps-landing-page-text-color: #ffffff !important;` – sets the colour of the title/subtitle and general text on the landing page.
- `--instant-apps-landing-page-entry-button-color: var(--brand-red) !important;` – fallback colour for the button if the gradient fails.

To adjust the overall look of the cover, you can apply styles directly to `instant-apps-landing-page` or its `::part(container)`. For example, the user’s CSS uses a slight saturation and contrast adjustment on the background:

```
instant-apps-landing-page::part(container) {  
  backdrop-filter: saturate(1.05) contrast(1.05);  
}
```

### 2.2 Title and subtitle

The compiled CSS defines an `esri-page` class used for cover pages within some Instant Apps (this may differ slightly across releases). It positions the text container in the centre of the page and sets responsive font sizes. The class `.esri-page__title-text` controls the title’s font weight and margin, while `.esri-page__subtitle-text` handles the subtitle. Both receive a text shadow for legibility <sup>1</sup>. Font sizes adjust at breakpoints of 480 px and 768 px (`@media screen and (max-width: 480px)` etc.) <sup>2</sup>.

If you need to change the title font size or alignment you can override these classes in your custom CSS. For example:

```
.esri-page__title-text { font-size: 3rem; text-align: center; }  
.esri-page__subtitle-text { font-size: 1.5rem; }
```

## 2.3 Enter/Explore button

In modern builds, the **Enter** button on the cover page is a `calcite-button`. Because this component uses a Shadow DOM, its visual styling comes from CSS variables and the exposed `::part(button)` pseudo-element. The user's custom CSS applies a brand-coloured gradient, drop shadow and border radius to the host `calcite-button`, then makes the internal button transparent to let the gradient show through. The `:hover` and `:focus-visible` states are also handled. For older builds where the button is rendered as an anchor element with the class `.landing-page__entry-button`, equivalent styles are provided.

If you wish to change the button's colour, gradients or animation, target either `instant-apps-landing-page calcite-button` or `.landing-page__entry-button`. Note that you cannot style the internal Shadow DOM of `calcite-button` without using `::part(button)`.

## 3 Main page structure

After clicking **Enter**, the application renders the main Portfolio interface. The page is structured in layers:

1. **Header** (`instant-apps-header`) – A sticky bar at the top containing logos, title, search and optional controls.
2. **Navigation** – Depending on the chosen layout, this is either a *Tabbed* list (`esri-portfolio-tabbed`), a *Carousel* (`esri-portfolio-carousel`) or an *Accordion* (`esri-portfolio-accordion`).
3. **Stage** (`esri-portfolio-stage`) – The area where the current section's content (map, scene, PDF or embedded app) is displayed.
4. **Description panel** – An optional sliding panel with text and links.
5. **Control panel** – Buttons for play/pause, next/previous, showing description, bookmarks, etc.

### 3.1 Header (`instant-apps-header`)

The header is a custom element that exposes parts for limited styling. Its default CSS is not visible in the compiled stylesheet, but the user's custom CSS demonstrates how to override it effectively:

- **Gradient background** – A multi-stop linear gradient transitioning from brand red to dark gray is applied to the header background. A secondary linear gradient adds a dark overlay for contrast. The background animates slowly via the `appHeaderShift` keyframe.
- **Calcite colour tokens** – Within the header, several CSS variables control the colours of embedded Calcite components. For example, `--calcite-color-text-1` and `--calcite-color-text-2` define primary and secondary text colours, while `--calcite-color-brand` and `--calcite-color-brand-hover` affect button highlights.
- **Container and wrapper parts** – Using `::part(container)` and `::part(wrapper)`, padding and background can be customised. The user sets both parts to transparent and applies consistent side padding.
- **Overlay logo** – The custom CSS injects a logo into the header by using `::after` on `instant-apps-header`. It positions the logo centrally via absolute positioning and CSS variables controlling the logo's width and height.

Because `instant-apps-header` is a Web Component, you cannot directly style its internal markup. Instead, use CSS variables exposed by the component or `::part` selectors, as shown above. Avoid changing its layout (height, position) because other elements expect the header to occupy a fixed height defined by `--instant-apps-header-height`.

### 3.2 Tabbed navigation

In the **Tabbed** layout (`#tabbedLayout`), the navigation bar is built from an unordered list with the class `.esri-portfolio-tabbed__tab-list`. Each tab is a list item with the base class `.esri-portfolio-tabbed-item`, and the currently selected tab also carries `.esri-portfolio-tabbed-item--selected`<sup>3</sup>. Important selectors include:

- `.esri-portfolio-tabbed__tab-list` – Defines the horizontal bar that holds the tabs. In the default dark theme the bar has a dark gray background; in the user's CSS it is replaced with a navy-to-gray gradient, includes inner shadows, gap spacing and extra padding.
- `.esri-portfolio-tabbed-item` – Styles individual tabs. Default styles include a dark background, white text, border radius and a transition on background and colour<sup>3</sup>. The user's CSS overrides idle tabs to have semi-transparent white backgrounds and defines a subtle hover state.
- `.esri-portfolio-tabbed-item--selected` – Indicates the active section. In the default CSS it simply changes the background colour to blue<sup>4</sup>. The custom CSS replaces this with a red-to-gray animated gradient (`tabRedShift`) and adds an inset border and text shadow for legibility. The animation speed is controlled by the `--tab-gradient-speed` variable.
- `.esri-portfolio-tabbed-item__title-info-container` and `.esri-portfolio-tabbed-item__item-title` – Contain the tab title text and optional info button. The compiled CSS truncates long titles with ellipses using `-webkit-line-clamp:1` and `text-overflow: ellipsis`<sup>5</sup>. You can increase font size or allow wrapping by overriding these selectors.

If the tab bar is clipped by its parent (this occurred in earlier builds), the user's CSS ensures overflow is visible and removes an unwanted mask element by targeting `[class*="esri-portfolio-tabbed__tabs"]` and `.esri-portfolio-tabbed__tab-list::after`.

### 3.3 Carousel navigation

In **Carousel** mode the navigation is rendered as a horizontally scrolling list. Each section is represented by a `.esri-portfolio-carousel-item` containing a `calcite-card`. The compiled CSS defines fixed widths (220 px) and heights for thumbnails, positions icons, and truncates long titles<sup>6</sup>. The selected item uses blue foreground and white text via Calcite variables<sup>7</sup>. On smaller screens (max-width 813 px), images are hidden and only icons are shown<sup>8</sup>.

You can override card styling by targeting `.esri-portfolio-carousel-item calcite-card` and adjusting colours using Calcite variables (`--calcite-color-foreground-1`, etc.). Be cautious not to alter widths or overflow properties, as the carousel relies on fixed sizing for scroll behaviour.

### 3.4 Accordion navigation

The **Accordion** layout stacks sections vertically. Each entry is an `esri-portfolio-accordion-item` inside `.esri-portfolio-accordion`. The compiled CSS sets `word-break: break-word` for long

descriptions and caps image widths <sup>9</sup>. Hovering a selected accordion item updates the Calcite accordion header background colour through the `--calcite-accordion-item-header-background-color-hover` variable <sup>10</sup>. For large accordions, `.esri-portfolio-accordion--large` widens the panel, while `.esri-portfolio-accordion--light` switches to a white background <sup>11</sup>.

### 3.5 Content stage ( `esri-portfolio-stage` )

The **Stage** displays the content of the current section. It contains either an `iframe` (for embedded websites) or a map/scene container. Key classes include:

- `.esri-portfolio-stage` - Sets the stage to `position: relative; width:100%; height: 100%` and applies a dark background <sup>12</sup>. The `.esri-portfolio-stage--light` variant switches to a light background <sup>13</sup>.
- `.esri-portfolio-stage__view-container` - The container for maps/scenes; absolutely positioned at the top left and fills the available space <sup>14</sup>.
- `.esri-portfolio-stage iframe` - Ensures embedded content takes up the full stage area <sup>14</sup>.
- `.esri-portfolio-stage__open-link-in-app` - A full-screen overlay shown when an item cannot be embedded (for example, if a web map denies embedding). It contains a `calcite-button` that prompts the user to open the item in its native viewer <sup>15</sup>.

Avoid changing the absolute positioning and sizing of these elements, as they are critical for responsive behaviour. You can, however, modify background colours or overlay styles to better match your brand.

### 3.6 Description panel ( `esri-portfolio-description` )

The description panel is an off-canvas panel that slides into view when the user clicks the info button. It uses several classes:

- `.esri-portfolio-description` - Positions the panel absolutely, sets its width to roughly **30 %** of the stage and makes it semi-transparent dark by default <sup>16</sup>. In light mode ( `.esri-portfolio-description--light` ) the background becomes white <sup>17</sup>.
- `.esri-portfolio-description__toggle-btn` - A vertical toggle button anchored to the side; it has a border to separate it from the panel and shows a hover state <sup>18</sup>. A light variant changes its colour <sup>19</sup>.
- `.esri-portfolio-description--open` - Moves the panel completely off screen ( `top: calc(-100% - 2px)` ) until it is opened again <sup>20</sup>. Toggling this class is handled by app logic.
- `.esri-portfolio-description__close-button-container` and `__main-content-container` - Provide margin and justify text within the panel <sup>21</sup>.

You can adjust the panel width, colours and typography by overriding these selectors. However, altering the positioning rules (top/bottom) may interfere with slide animations.

### 3.7 Control panel ( `esri-control-panel` )

The control panel sits above the stage and contains play/pause buttons, next/previous section navigation and other controls. It is defined by `.esri-control-panel` and subdivided into nested classes such as `.esri-control-panel__controls`, `.esri-control-panel__control-button` and `.esri-`

`control-panel__play-pause-button`. In dark theme the controls use dark backgrounds and white icons <sup>22</sup>; a light theme variation updates background and icon colours <sup>23</sup>.

If you want larger buttons or different icon colours you can override `.esri-control-panel__control-button calcite-icon` and `.esri-control-panel__play-pause-button`. The user's custom CSS does not override these but they can be styled similarly to the tabs. Avoid changing the `position` property of `.esri-control-panel` as it is tied to the navigation height.

### 3.8 Location search (`esri-portfolio-location`)

When enabled, the **Location** panel lets users search for an address or coordinate and navigate directly to that location in the content. It is a flex column container with a dark or light background and contains search inputs and buttons. The compiled CSS sets widths, minimum widths (300 px), margins and colours <sup>24</sup>. At smaller breakpoints the width is set to 100 % and font sizes are increased <sup>25</sup>. To adjust its appearance you can override `.esri-portfolio-location`, `.esri-portfolio-location__header-text`, `.esri-portfolio-location__search-container` and `.esri-portfolio-location__enter-button`. Avoid altering `position` or `z-index`, which ensures the panel sits above the stage.

### 3.9 Mobile/responsive classes

The Portfolio template adapts to mobile viewports by switching the navigation into a column and adjusting heights. Responsive behaviour is controlled by classes prefixed `.esri-portfolio-mobile`. For example, `.esri-portfolio-mobile__header-selector-container` changes the header height on mobile (`--instant-apps-header-height: 55px`) <sup>26</sup>, while `.esri-portfolio-mobile__section-selector-container` provides a dropdown for section selection <sup>27</sup>. Panels such as `.esri-portfolio-mobile__description` slide up from the bottom and occupy 40 % of the viewport <sup>28</sup>.

When overriding styles, always test on multiple screen sizes. Changing heights or widths in desktop mode may cause mobile breakpoints to behave unexpectedly if the CSS relies on percentage-based calculations (for example `height: calc(100% - 110px)` <sup>29</sup>).

## 4 The user's custom CSS (`test_tellepsen.css`)

The provided stylesheet establishes a brand identity across the application and overrides several Instant Apps defaults. Key features include:

- **Brand tokens** – Custom CSS variables (`--brand-red`, `--brand-gray-1`, `--brand-navy-1`, etc.) define a palette of reds, grays and navy blues for reuse throughout the file. Animation speeds for the header and tabs are defined by variables (`--app-header-gradient-speed`, `--tab-gradient-speed`). Overlay logo URL and dimensions are also set via variables.
- **Header override** – Applies a red-to-gray animated gradient to `instant-apps-header`, sets foreground colours via Calcite variables and inserts a centred PNG logo with a drop shadow. Motion is disabled for users who prefer reduced animation.

- **Tab styling** – Replaces default tab backgrounds with semi-transparent white, defines a hover effect and creates an animated gradient for the selected tab via the `tabRedShift` keyframe. Idle and hover states for the tab bar are customised later in the file.
- **Cover page button** – Provides a gradient background, rounded corners and drop shadows for the `calcite-button` used on the landing page. The fallback anchor button receives matching styles. Focus and hover states are clearly defined to improve accessibility.
- **Tab strip background** – The bar behind the tabs uses a vertical gradient from navy to dark gray, includes a subtle inner shadow and ensures that overflow is visible and not clipped.

## What can and cannot be overridden

- **Can override:** Colours, gradients, borders and shadows of most top-level elements, including header backgrounds (`instant-apps-header`), navigation bars (`esri-portfolio-tabbed__tab-list`, `.esri-portfolio-carousel`, `.esri-portfolio-accordion`), tab items (`.esri-portfolio-tabbed-item`), description panel backgrounds, location panel colours, and the landing page button. Font sizes and spacing can be adjusted by targeting the appropriate classes (`.esri-page__title-text`, `.esri-portfolio-tabbed-item__item-title`, etc.).
- **Can override via CSS variables:** Many Calcite components expose variables prefixed `--calcite-color-*`, `--calcite-font-size-*` and `--calcite-radius-*`. Changing these variables inside the appropriate scope (e.g., `instant-apps-header`) updates colours and typography for nested controls.
- **Requires `::part` selectors:** Components like `instant-apps-header` and `calcite-button` encapsulate their internals in Shadow DOM. Use `::part(container)` and `::part(wrapper)` on `instant-apps-header` and `::part(button)` on `calcite-button` to style internal wrappers. The sample CSS demonstrates this approach.
- **Cannot override:** Structural layout properties (e.g., absolute positioning, flex directions, viewport heights) should generally not be changed, as they are tightly coupled to responsive behaviour. Avoid overriding classes that control `position`, `top`, `left`, or `transform` unless you understand the impact. For example, `.esri-portfolio-stage` and `.esri-portfolio-description` rely on absolute positioning <sup>12</sup> <sup>16</sup>. Similarly, you cannot modify the internals of third-party iframes embedded in the stage.

## 5 Practical tips for developing a custom CSS theme

1. **Use variables consistently** – Define brand colours and animation durations at the top of your stylesheet as variables. This makes it easy to adjust the palette across the entire app.
2. **Respect themes** – Test your overrides in both dark and light mode. The compiled CSS uses `.calcite-mode-light` and `.calcite-mode-dark` selectors to switch colours. Provide complementary colours in your overrides to ensure adequate contrast.
3. **Leverage part selectors** – Whenever you are styling a Web Component (`instant-apps-header`, `calcite-button`), use the `::part()` pseudo-element to target internal wrappers. Without it your rules will not penetrate the Shadow DOM.
4. **Avoid layout breaks** – Keep the existing layout intact. Only change visual aspects (colours, shadows, border radii, typography) unless you are building a custom layout from scratch. Unexpected changes to sizing or positioning can lead to clipped content or broken responsiveness.

5. **Test with long titles and many sections** – Portfolio apps can house many sections. Make sure your tab/accordion/carousel styles cope with long names, high item counts and mobile screens. Use `overflow: visible` to prevent clipping and avoid absolute widths.

## 6 Conclusion

The ArcGIS Instant Apps **Portfolio** template exposes a rich set of CSS selectors that allow you to customise its appearance without altering the underlying application logic. By understanding how the cover page, header, navigation bars, stage and panels are structured—and by using CSS variables and part selectors for Web Components—you can craft a polished, branded user experience. The accompanying `test_tellepsen.css` file serves as a concrete example of how to apply gradients, animations and custom colours to match corporate branding while preserving accessibility and responsiveness.

---

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29

[aallc.maps.arcgis.com](https://aallc.maps.arcgis.com)

<https://aallc.maps.arcgis.com/apps/instant/portfolio/assets/index.064c98d1.css>