



INITIALYZE

WKND MUZIK

Implementation design and reference

Document Control

Document Title	WKND Muzik project Content Architecture Runbook
Authors	Varshish Bhanushali
Document Purpose	Establishes the framework that governs the content lifecycle for the WKND Muzik project.

Version Control

Version Number	Date	Author	Comments
1	10-13-2020	Varshish Bhanushali	Initial version
2.	10-15-2020	Tushar Mane	Design System version
3.	10-15-2020	Deepak Bhanushali	Design System version

Executive Summary

The purpose of this document is to guide the reader with the necessary information required to populate the WKND MUZIK site with content in a manner that aligns with AEM best practices and standards.

The document outlines and identifies the required steps to establish a content architecture that populates the site with content that fully leverages core functionality included in the WKND Muzik project. The content architecture section identifies the WKND Muzik project content types, components and methods to establish the site taxonomy. The Component & Design Reference section identifies components and style system classes available to establish the site look and feel.

Table of contents

Executive Summary	3
Table of contents	4
Introduction	6
Document Purpose	6
Document Audience	6
Content Architecture	7
Content Types	7
Content Page	7
Home Page	8
Form page	8
Experience Fragments	9
Components	9
Template Components	10
Page Authoring Components	10
Container Components	10
Form Components	10
Taxonomy	10
Menu Structure	11
Header	12
Footer	12
Asset Management	12
Folder Structure	13
Component & Design Reference	14
Support for dark and light mode	14
Teaser	15
List component design	19
Carousel	19
Title	20
Button and its variations:	21
Navigation and its variations:	22
Image	23
Form Validations	24

Embed (Video From DAM Assets)
Spacing Classes

25
27

1. Introduction

1.1. Document Purpose

The purpose of this document is to help a site implementer establish a site content and design architecture of the WKND Muzik project. This document should be used as a guide to establish the organizational structure of the site that will determine how the content will be organized throughout the site. It also provides details on what features and styles are supported by various components.

1.2. Document Audience

This document is intended to be used by the Lean Code Contest Judges as well as the community to understand the content and component design architecture of WKND Muzik project.

2. Content Architecture

The content architecture establishes how the content is governed throughout its lifecycle, from concept, to publish and archive. It is important to establish standards and protocols that align with best practices in order to take full advantage of the full power and capabilities of the platform.

2.1. Content Types

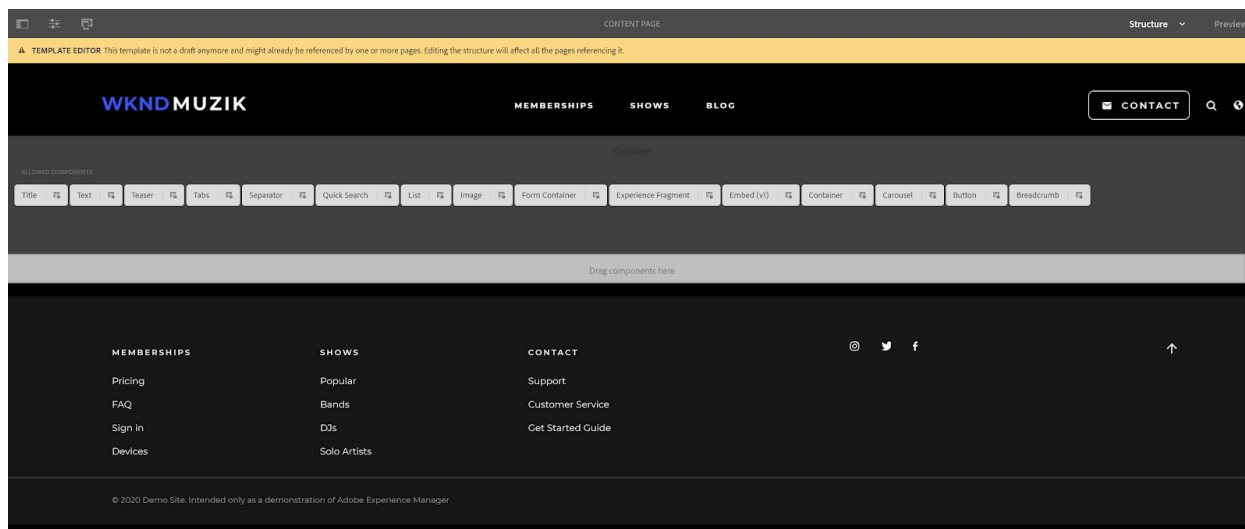
The WKND Muzik project contains configurable templates that are ready for use after the initial configuration and provisioning. These templates are designed to provide authors maximum flexibility for content creation, while at the same time, providing administrative governance and control ensuring brand and style standards are followed. The balance of flexibility and governance is achieved by determining what features and components are available for each template.

The following subsections describe the various content types that are available in the WKND Muzik project and their appropriate use cases.

2.1.1. Content Page

The content pages make up a majority of the site content. It is important to establish the elements that will be allowed on each content page template type, and minimize the element to only contain the necessary components required to convey the specific information associated with the page subject.

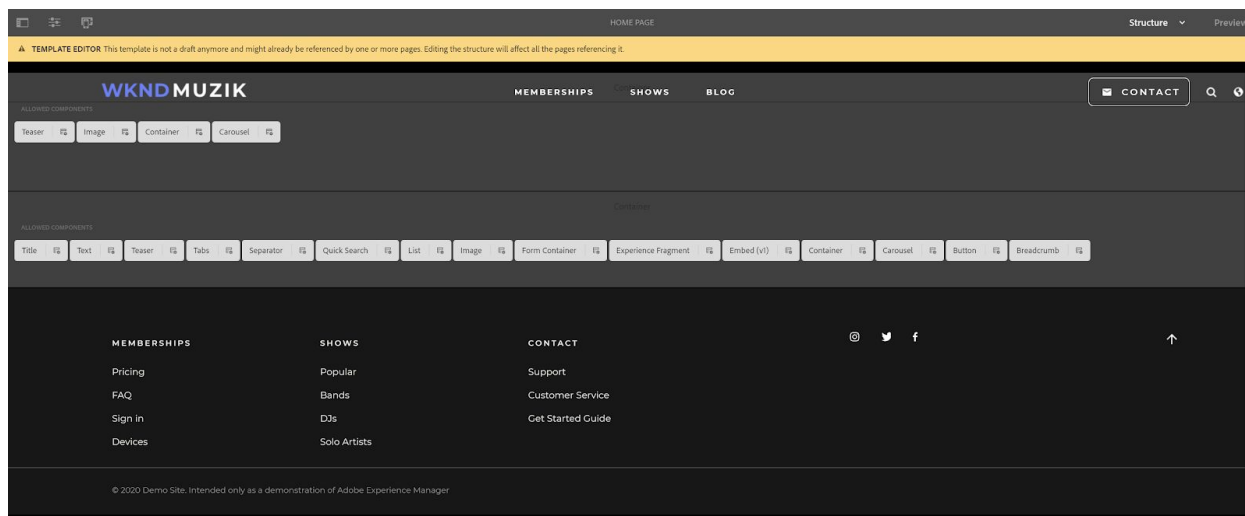
Content pages provide an opportunity to display content grouped in ways that promote information deemed important or relevant. Content pages will typically contain teaser and list components. These components typically display content from other pages and take many forms depending on the styles associated with the components.



2.1.2. Home Page

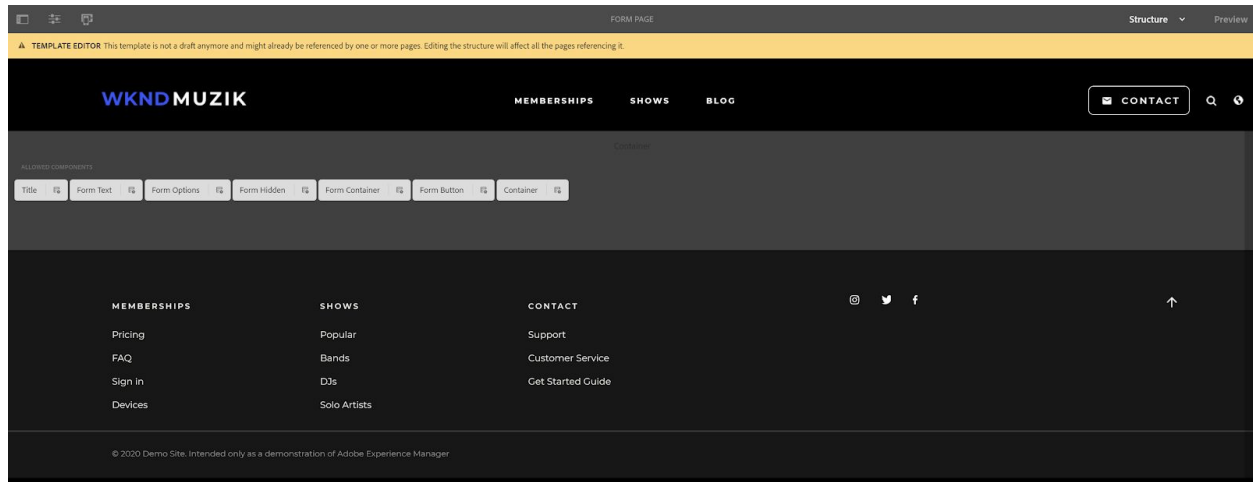
The home page typically receives the most attention, and yet for most digital experiences, contains less value than content pages. This dichotomy exists because the home page often represents the first impression of the business, and marketing initiatives often drive traffic directly to the home page of a site.

The WKND Muzik project's Home page template is divided into two groups. The 1st group contains the Hero components like carousel, teaser and Image. The 2nd group contains the components to build the body of the home page.



2.1.3. Form page

The WKND Muzik form page typically contains the form components to build signup or branding forms.



2.1.4. Experience Fragments

Experience fragments are available OOTB in Adobe Experience Manager. In WKND muzik project **the header and footer of the site are built using experience fragments.**

2.2. Components

The following table may be used as a guide to define the components that will be made available for any given page type:

Page Type	Required Components	Optional Components
Home Page	Carousel, Container, Teaser, List, Image, Title, Text, Button, Tabs,	Search, Separator, Experience fragment, Embed, From container, Breadcrumb
Content Page	Teaser, List, Container, Carousel, Tabs, Embed, Separator, Title, Breadcrumb, Text, Image	Search, Experience fragment, From container
Form Page	Title, Text, Form Components	

The following templates and components listed are available in the Digital Foundation Blueprint to be associated with the templates.

2.2.1. Template Components

1. Page
2. Navigation
3. Language Navigation
4. Breadcrumb
5. Quick Search
6. Image
7. Social Sharing

2.2.2. Page Authoring Components

8. Text
9. Title
10. Image
11. Teaser
12. Button
13. List
14. Separator
15. Experience Fragment
16. Embed

2.2.3. Container Components

17. Container
18. Carousel
19. Accordion
20. Tabs

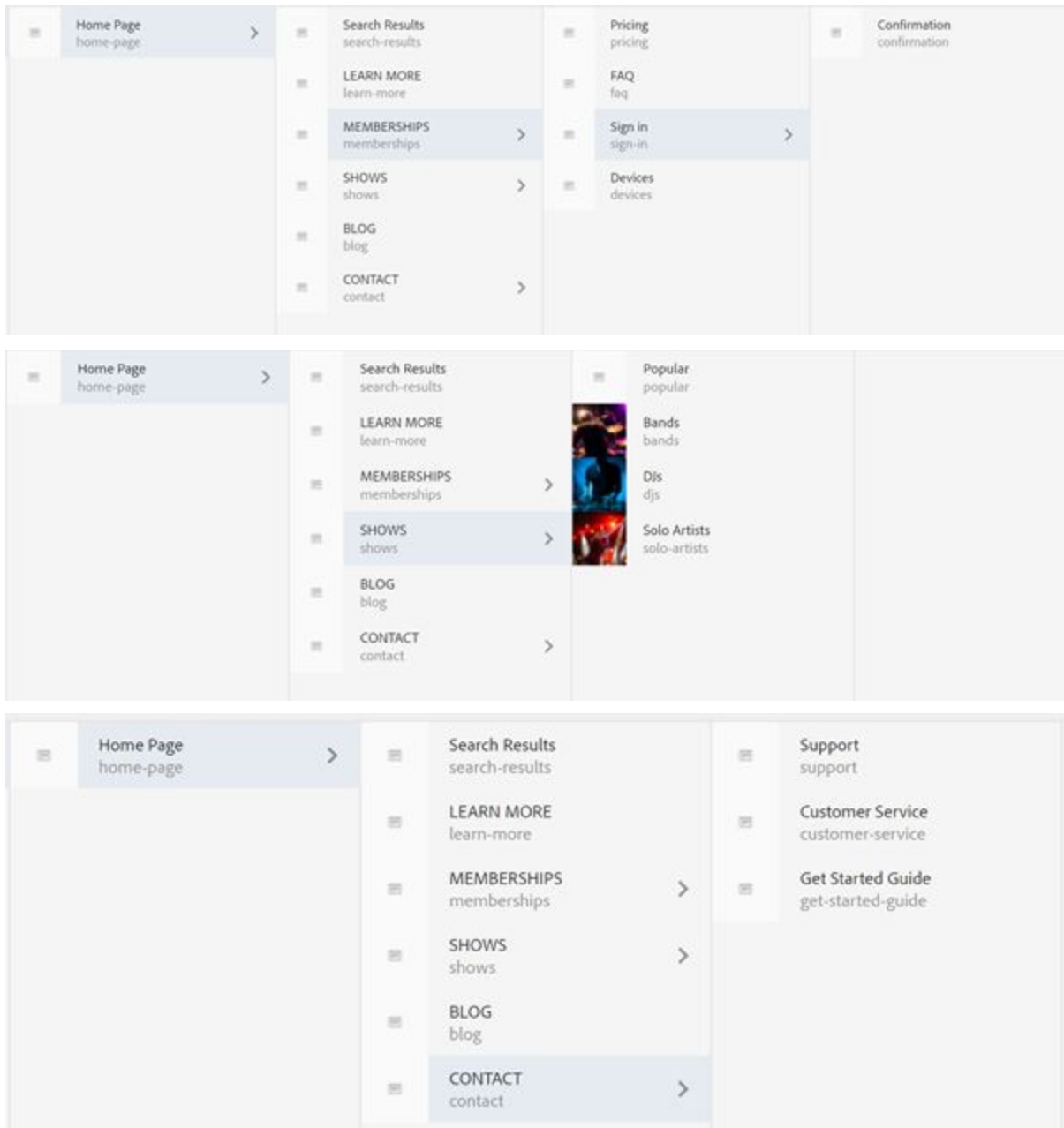
2.2.4. Form Components

21. Form Container
22. Form text field
23. Form options field
24. Form hidden field
25. Form button

2.3. Taxonomy

The taxonomy determines the hierarchical structure of the content which directly correlates to the URL structure of the site. The WKND Muzik site start with home page and its child pages used in header and footer navigation

2.3.1. Menu Structure



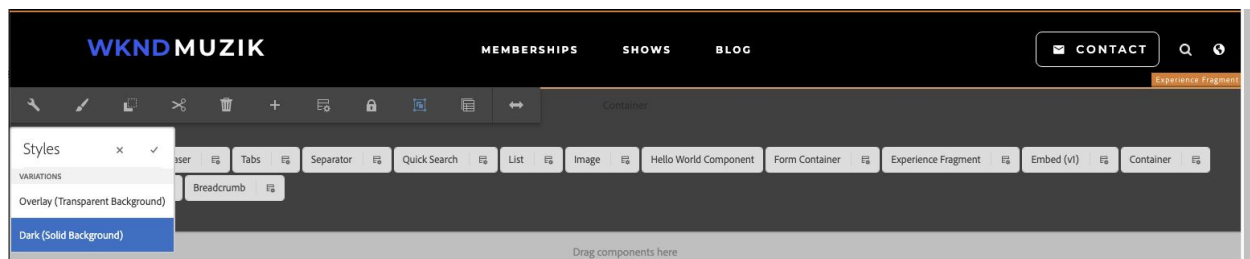
2.3.2. Header

The WKND Muzik header is built using an experience fragment. The header contains navigation, logo search, button and language navigation component



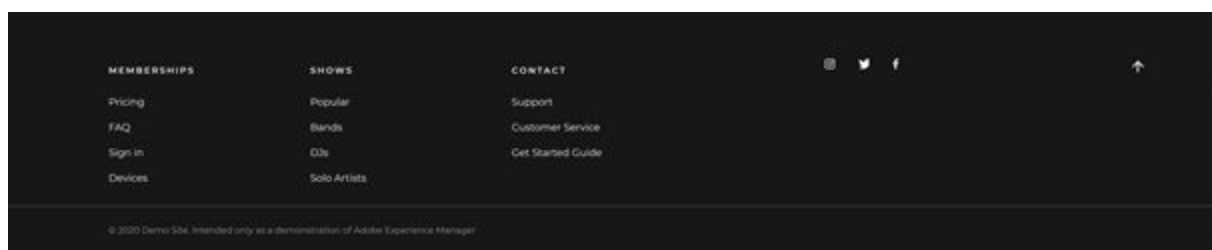
The Experience Fragment component is added to the template and supports two variations via style system

1. Overlay (transparent background)
This variation is used on the homepage template. It absolutely positions the header on top and overlays it, on top of the carousel.
2. Dark (solid background)
This variations ensures the navigation has a consistent dark background regardless of light or dark mode preference of the user



2.3.3. Footer

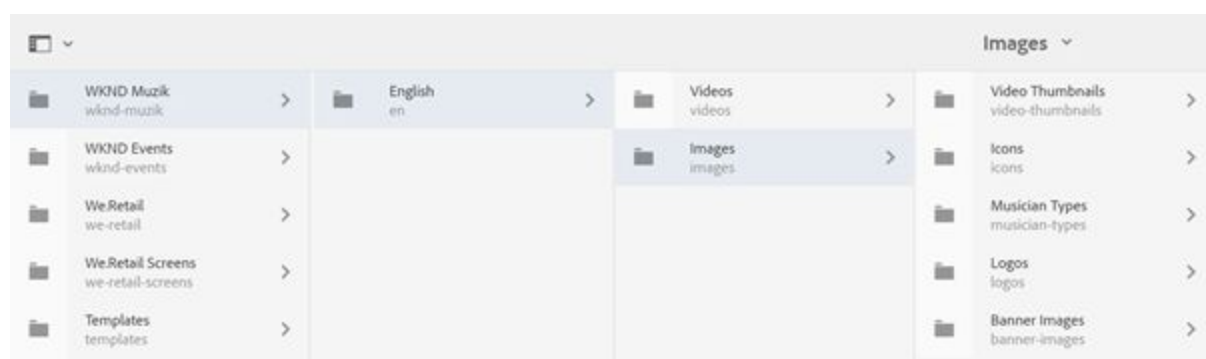
The footer allows an additional navigation area to include links to other important areas of the site's content that is less relevant than the primary content, but important to certain audiences. The WKND Muzik footer is built using experience fragment. The footer contains navigation, sharing, button, text, title and separator component



3. Asset Management

AEM Assets offers robust DAM (Digital Asset Management) functionality to ensure the assets are managed properly throughout their lifecycle. This includes ensuring they are organized in a folder structure that is compatible with the types of assets, and well as the asset function and purpose.

3.1. Folder Structure

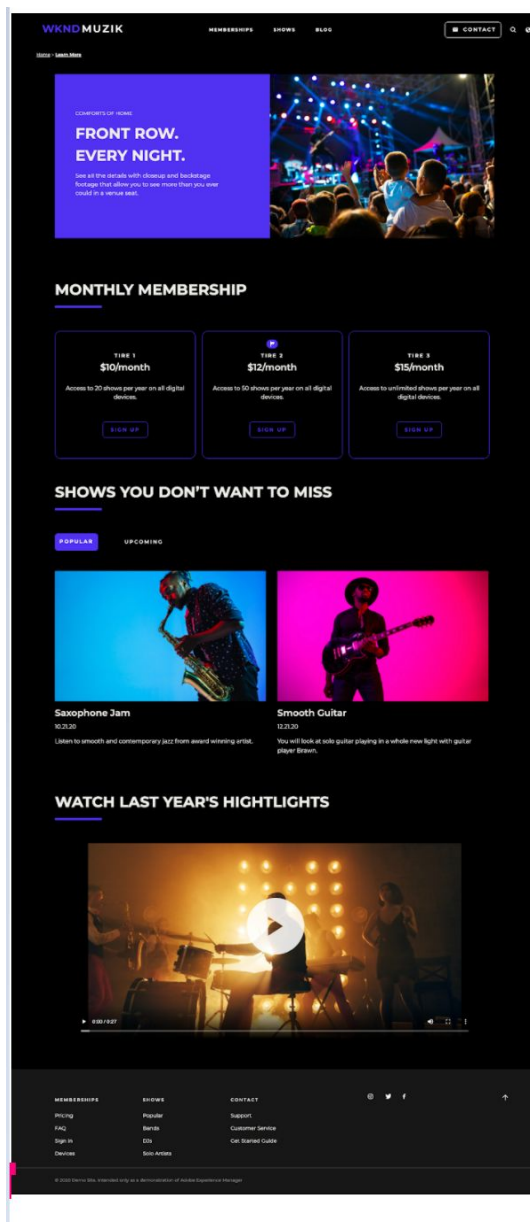


4. Component & Design Reference

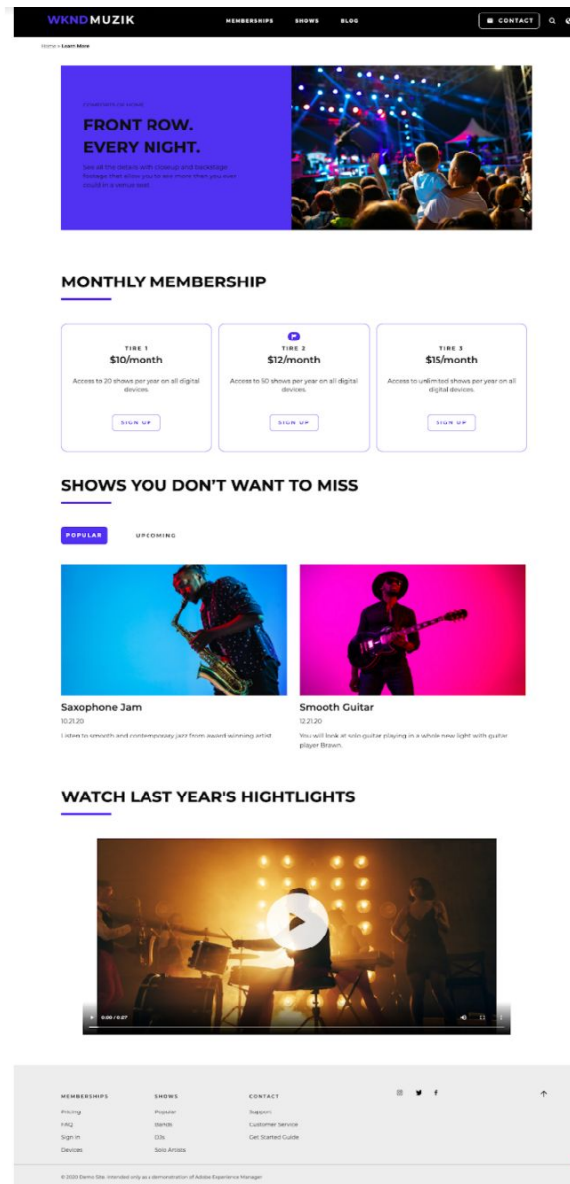
4.1. Support for dark and light mode

We have implemented the style sheet to support system dark and light modes. If the client is using the light mode the body color will use the lighter theme and vice versa.

When system display setting is in Dark mode



When system display setting is in light mode



4.2. Teaser

The teaser component is built by extending the WCM Core Teaser component and no customizations are implemented as all the display variations have been achieved using the Style system.

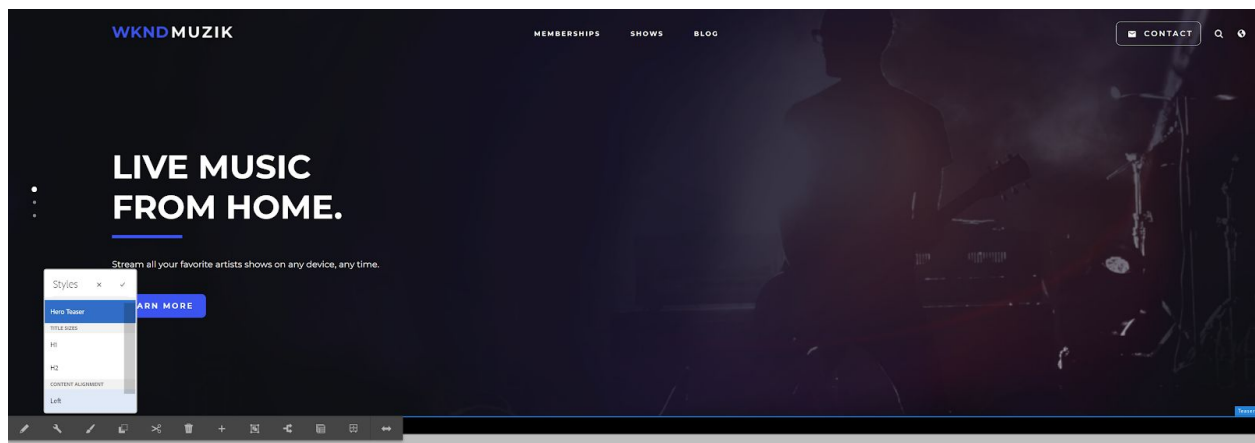
The teaser component supports various display attributes to be tweaked via the style system

1. Variation
2. Content Alignment
3. Title Sizes
4. Spacing (below?)

Hero Teaser 1:

In this variation the content is by default aligned to left. To achieve this look, user can select from the Style system dropdown:

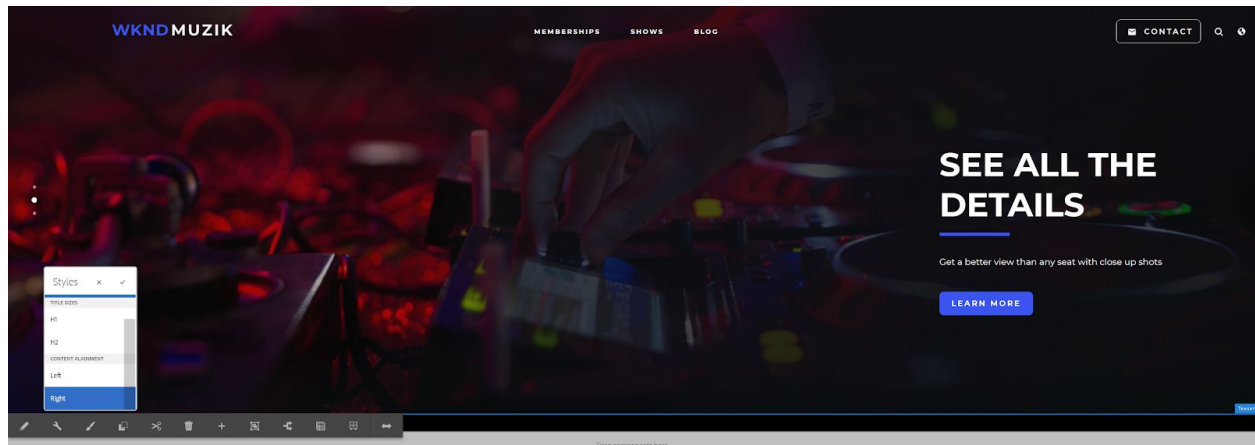
- Display Variations > “Hero Teaser”



Hero Teaser 2:

In this variation the content is aligned to the right. To achieve this look, the user can select following from the Style system dropdown:

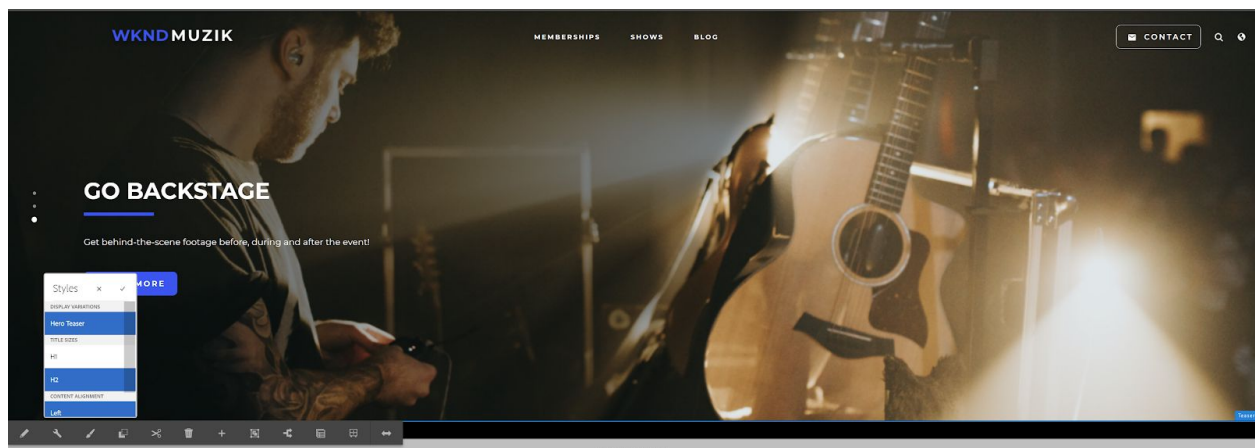
- Display Variations > “Hero Teaser”
- Content Alignment > “Right”



Hero Teaser 3 :

To achieve this look, the user can select following from the Style system dropdown:

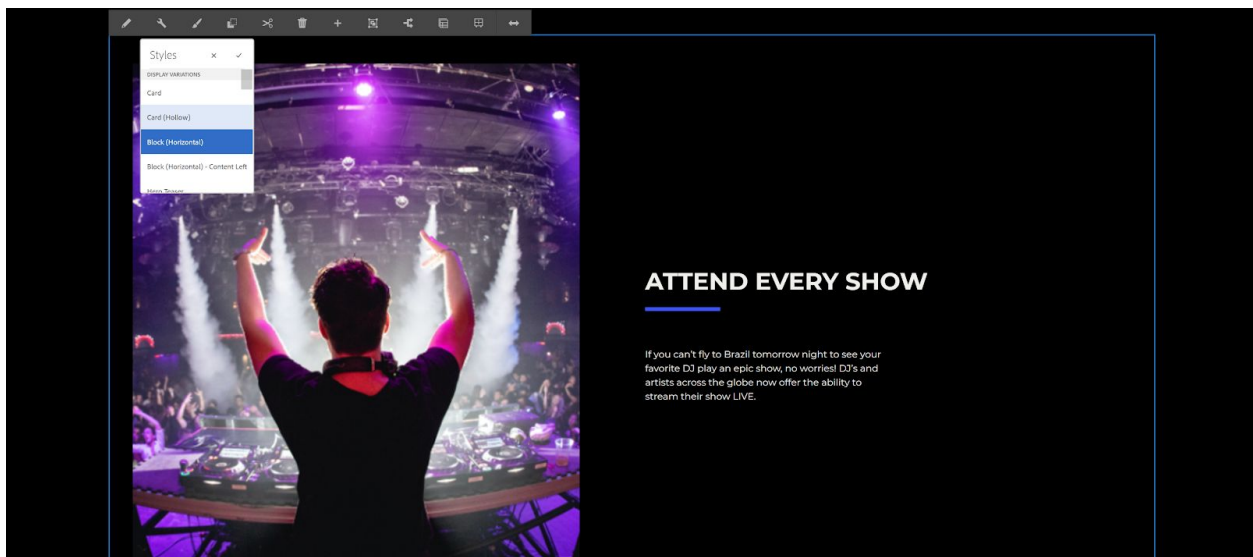
- Display Variations > “Hero Teaser”
- Text size > “H2”



Block Horizontal Teaser :

In this variation the teaser will look like a block . To achieve this look, the user can select following from the Style system dropdown:

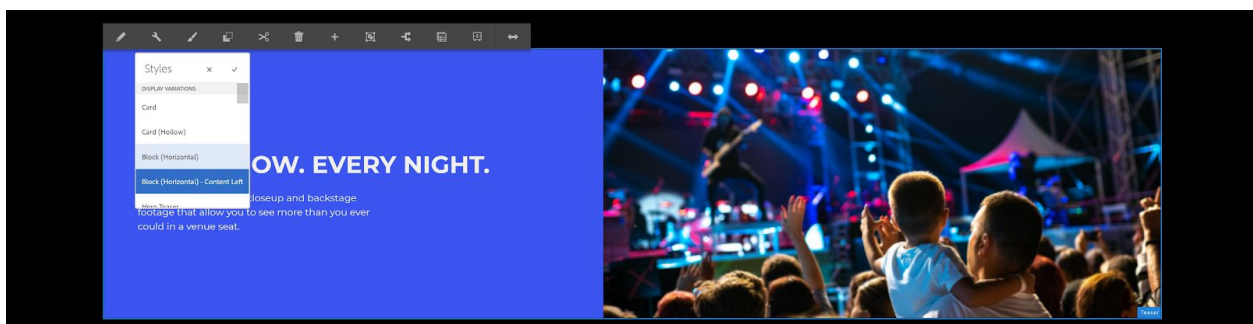
- Display Variations > “Block Horizontal”



Block Horizontal - Content left

In this variation the teaser will look like a block . To achieve this look, the user can select following from the Style system dropdown:

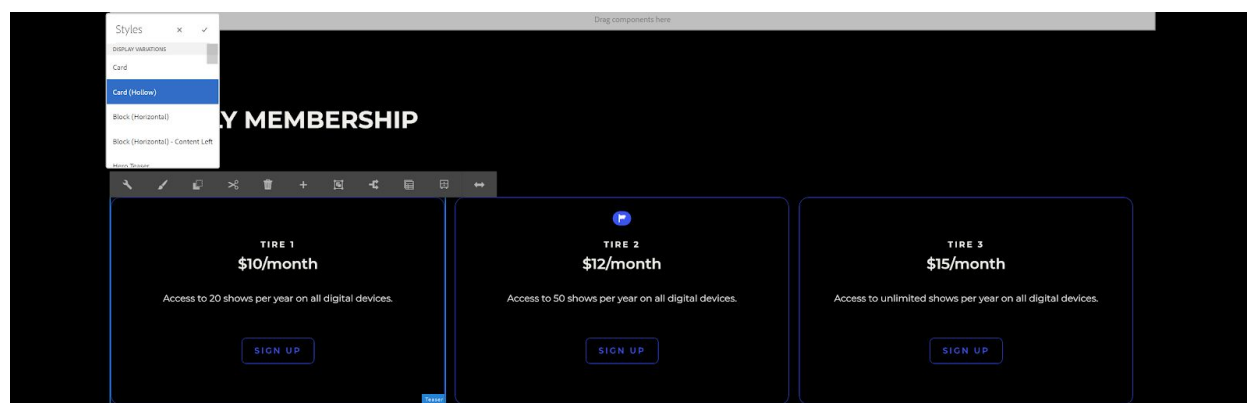
- Display Variations > “Block(Horizontal) - Content left”
- Background > “Blue”



Card Hollow Teaser:

In this variation the teaser will look like a hollow card. To achieve this look, the user can select following from the Style system dropdown:

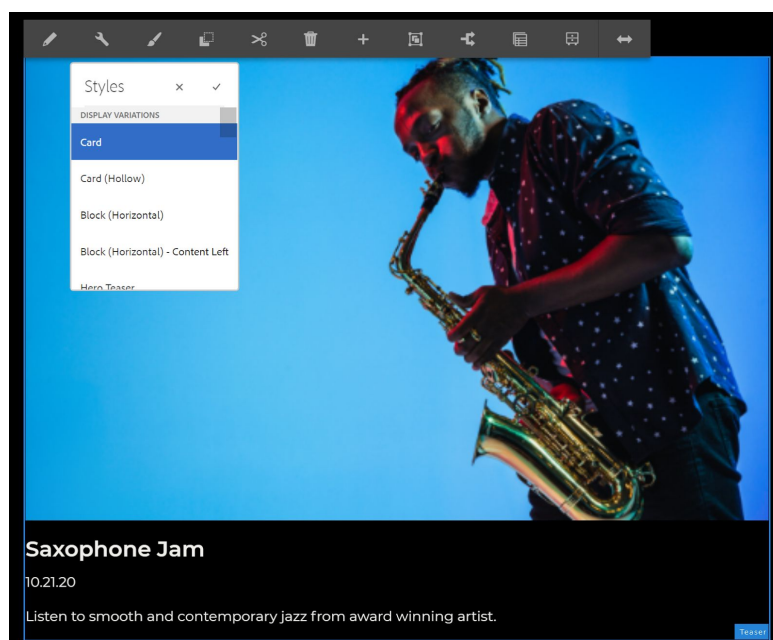
- Display Variations > “Card (Hollow)”
- Icon > “Flag”



Card Teaser:

In this variation the teaser will look like a card. To achieve this look, the user can select following from the Style system dropdown:

- Display Variations > “Card”

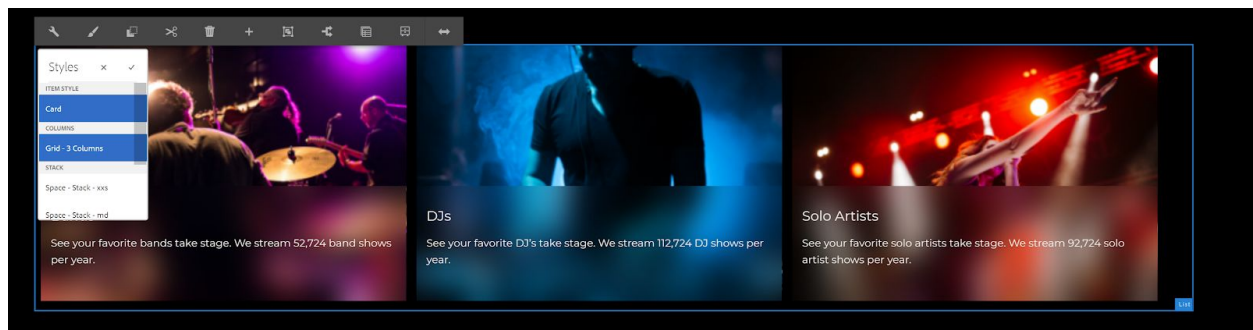


4.3. List component design

The title, image and description in the list component are fetched from page property. It supports card style with grid layout.

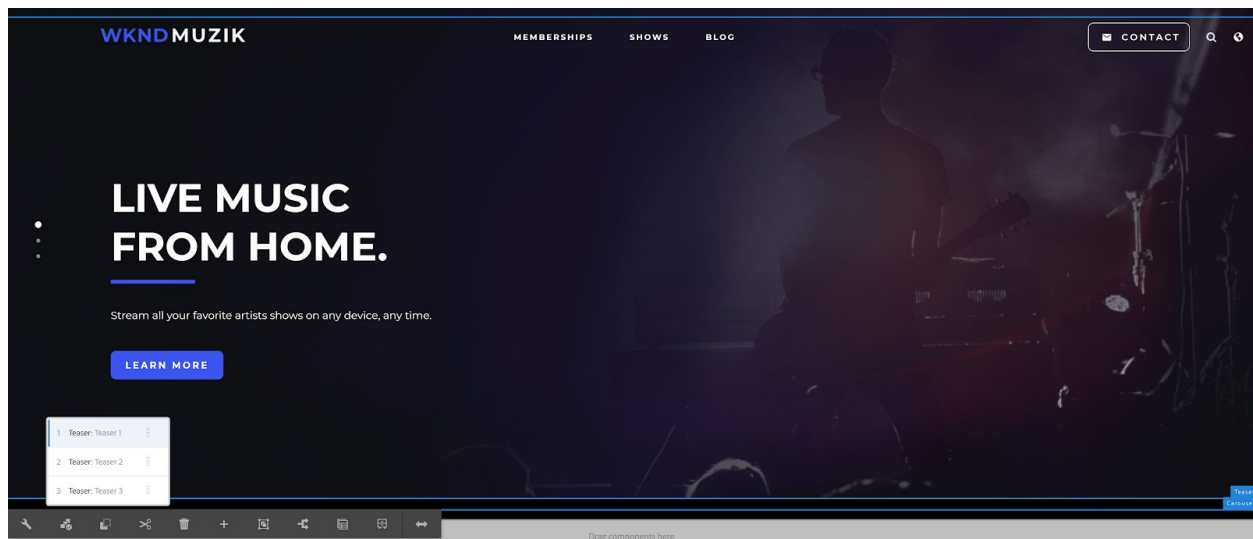
To achieve this look, the user can select following from the Style system dropdown:

- Item Style > “Card”
- Column > “Grid - 3 columns”



4.4. Carousel

The carousel component on the home page is authored using 3 hero teasers each with its own styles.



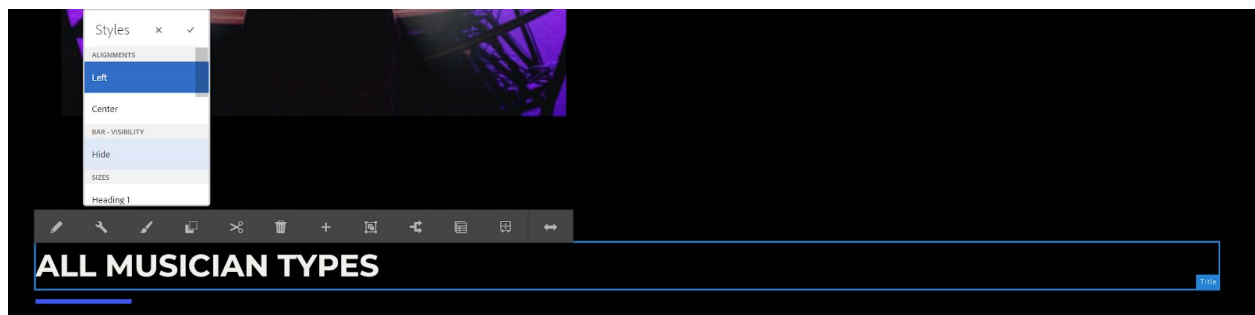
4.5. Title

Title component supports text alignment left and center with text sizes H1(60px) and H2(41px) and additional functionality of hide and show bar in the bottom of the title. Below are the examples with different variations.

Title with text and bar as left aligned

To achieve this look, the user can select following from the Style system dropdown:

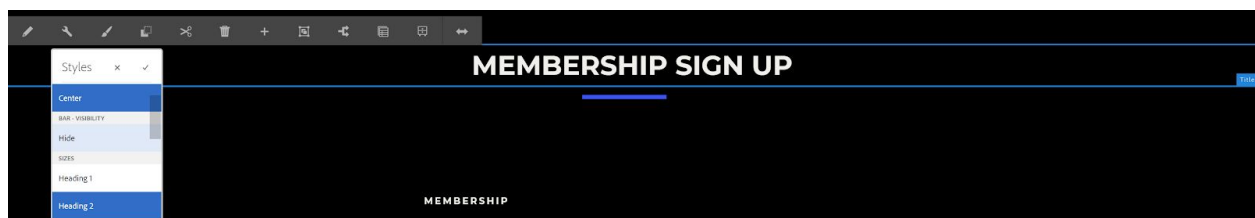
- Alignment > “Left”
- Size > “Heading 2”



Title with text and bar as center aligned

To achieve this look, the user can select following from the Style system dropdown:

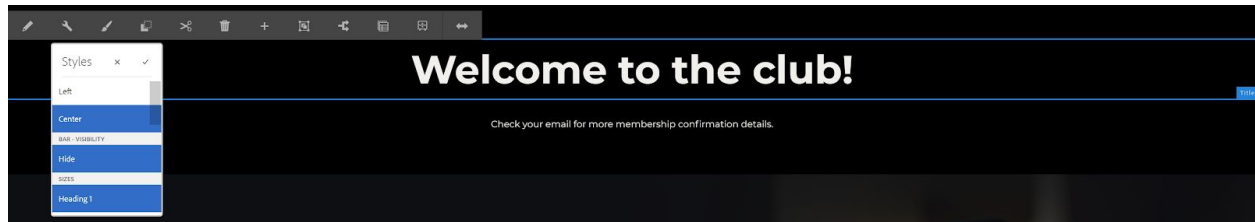
- Alignment > “Center”
- Size > “Heading 2”



Title with text aligned center and bar hidden

To achieve this look, the user can select following from the Style system dropdown:

- Alignment > “Center”
- Size > “Heading 1”
- Bar Visibility > “Hide”



4.6. Button and its variations:

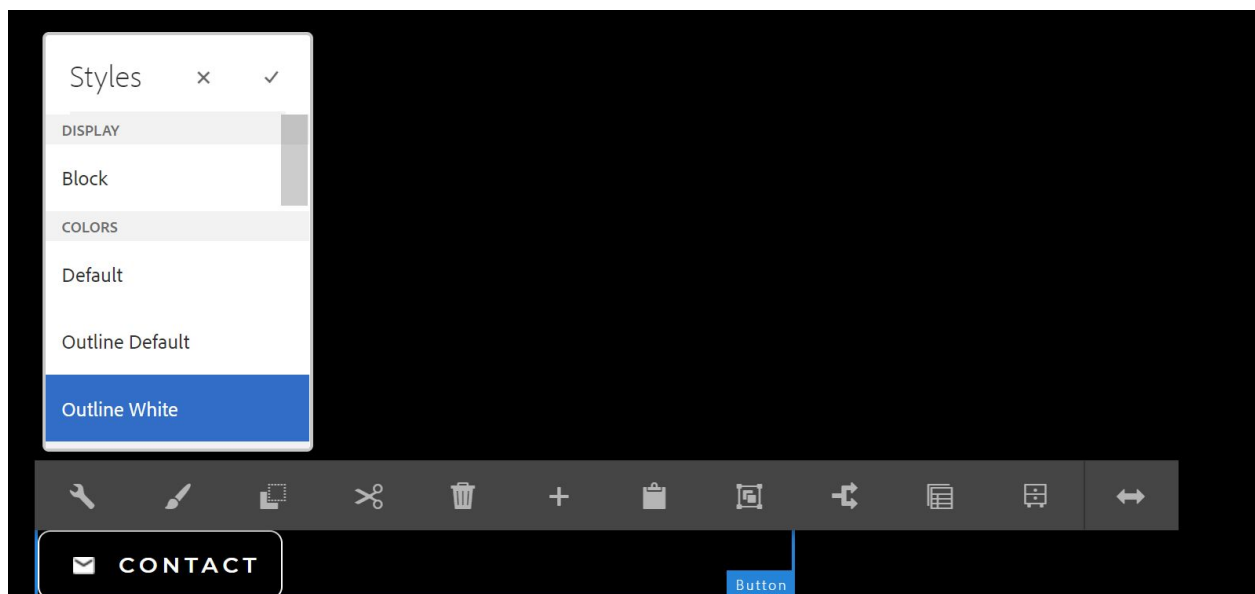
Button components supporting multiple designs below are the list of design variations with screenshot. It also has the functionality of adding an icon in the button and also supports the alignments.

Hollow button(white outline) with icon

To achieve this look, the user can select following from the Style system dropdown:

- Colors > “Outline White”

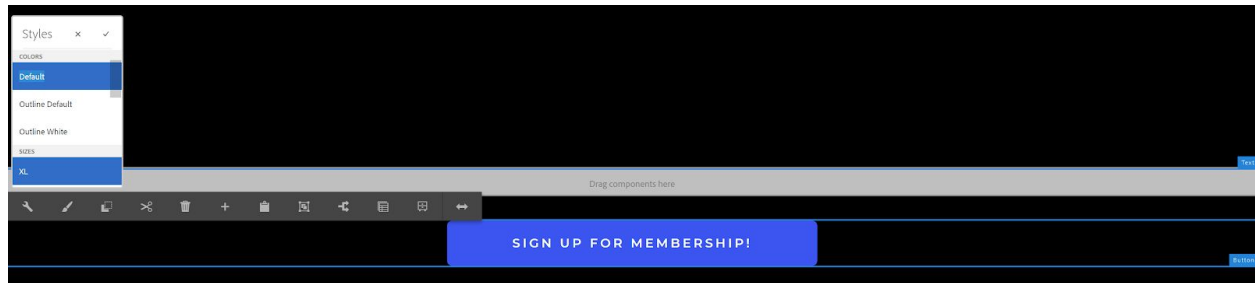
And need to add icon = email in property tab of cq:dialog



Default button

To achieve this look, the user can select following from the Style system dropdown:

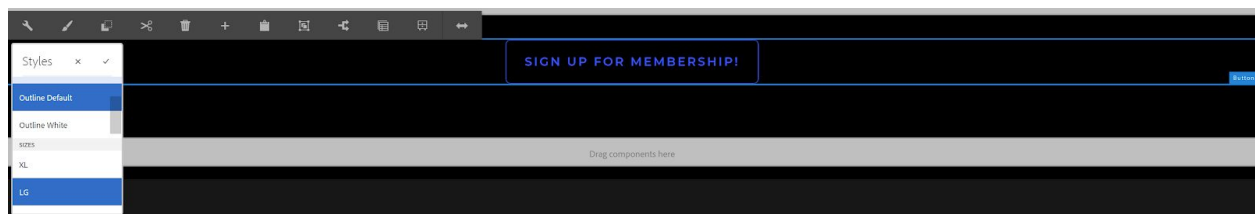
- Colors > “Default”
- Size > “XL”



Hollow button(default outline)

To achieve this look, the user can select following from the Style system dropdown:

- Colors > “Outline default”
- Size > “LG”



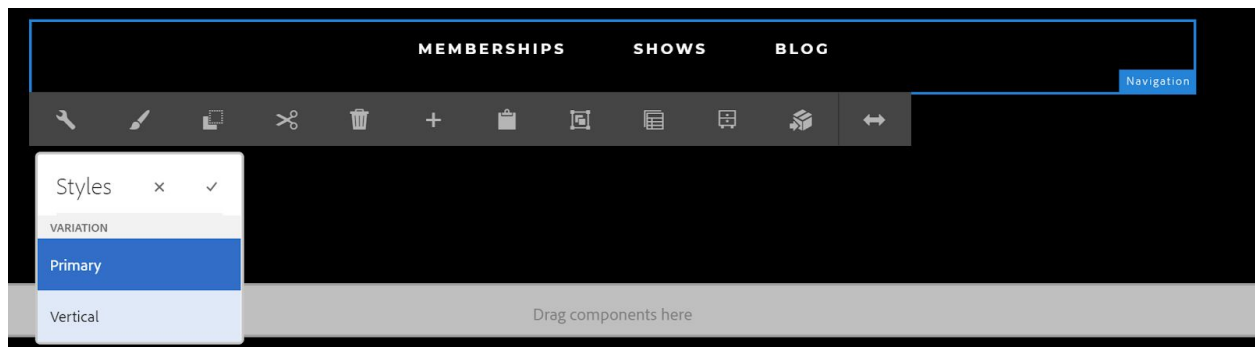
4.7. Navigation and its variations:

Navigation component supports the two variations one is primary which is used in the header of the page and second is vertical which is used in the footer of the page.

Navigation primary

To achieve this look, the user can select following from the Style system dropdown:

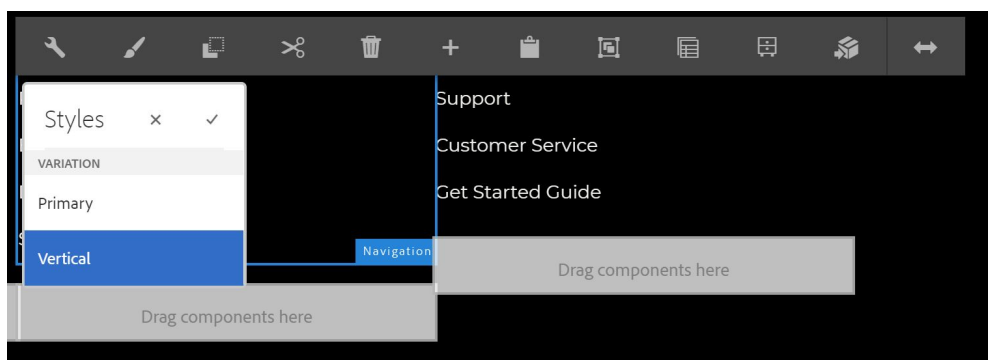
- Variation > “primary”



Navigation vertical

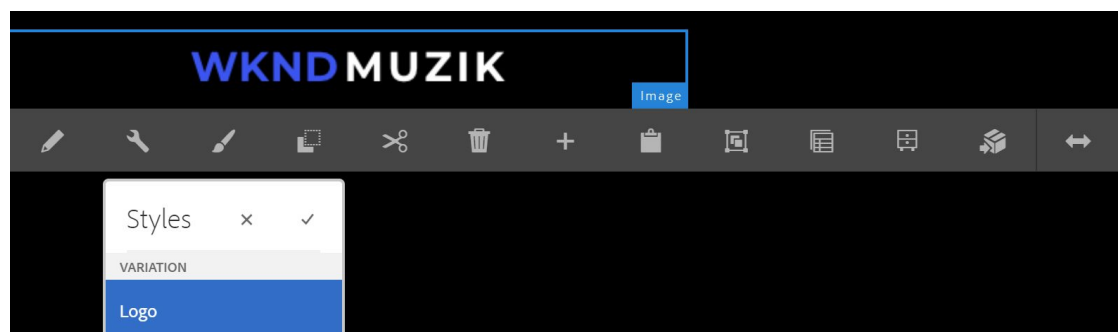
To achieve this look, the user can select following from the Style system dropdown:

- Variation > “vertical”



4.8. Image

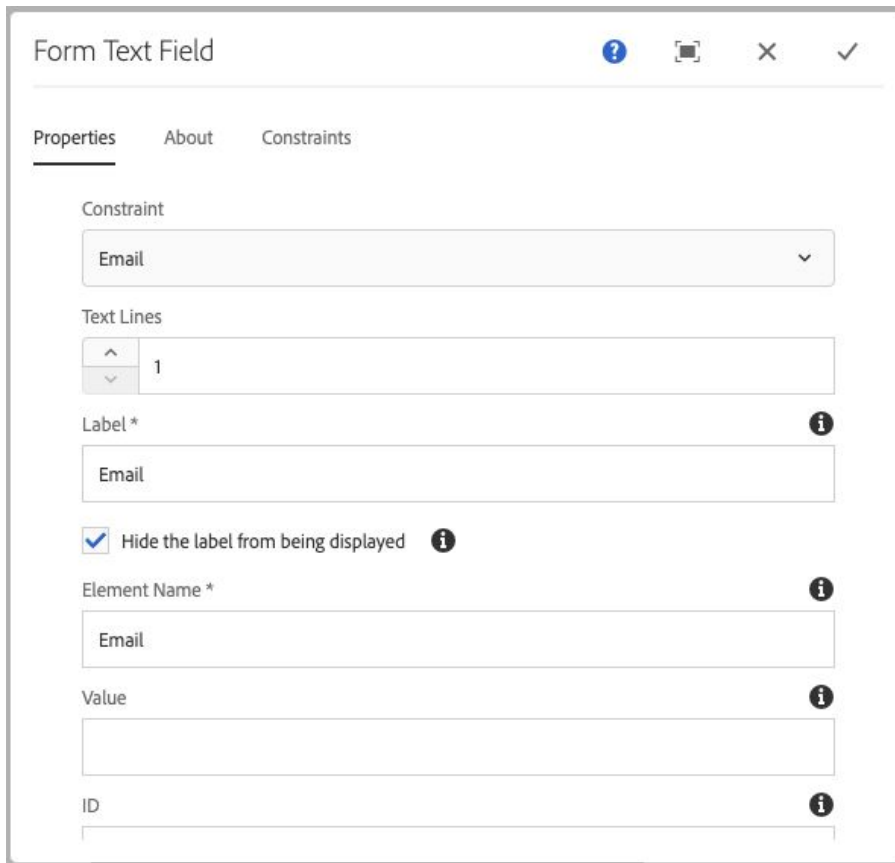
Logo style: Image component supports logo styling applied via style dropdown.



4.9. Form Validations

Form validation has wcm core standard form text-field validation and we have enhanced one HTML5 pattern validation, so the author can add regex based on the constraint authored.

Fields to author in dialog:



The screenshot shows a dialog box titled "Form Text Field" with three tabs: "Properties", "About", and "Constraints". The "Properties" tab is active. It contains the following fields:

- Constraint:** A dropdown menu with "Email" selected.
- Text Lines:** A numeric input field with "1" entered.
- Label *:** A text input field with "Email" entered. An information icon (i) is to the right.
- Hide the label from being displayed:** A checked checkbox with an information icon (i) to the right.
- Element Name *:** A text input field with "Email" entered. An information icon (i) is to the right.
- Value:** An empty text input field. An information icon (i) is to the right.
- ID:** An empty text input field. An information icon (i) is to the right.

Output:



The screenshot shows a text field containing the email address "tushar@initialyze.". To the right of the text is a red circle with a white exclamation mark, indicating an error. Below the text field, the message "Incorrect email formatting" is displayed in red text.

4.10. Embed (Video From DAM Assets)

The Embed component was implemented by extending WCM Core's Embed Component. It now supports a new embeddable which enables the author to select and display videos from AEM DAM.

The screenshot shows the 'Embed' dialog box with the following configuration:

- Embeddable:** Video From DAM Assets
- Video Asset Path *:** /content/dam/wknd-muzik/en/videos/www_wp-content_uploads_2020_05_P (checked)
- Poster Image Path:** /content/dam/wknd-muzik/en/images/video-thumbnails/adobe-stock-thumbn (checked)
- Width:** 1091
- Height:** 541
- Autoplay:** ☐ (unchecked)
- Loop:** ☐ (unchecked)
- Hide Controls:** ☐ (unchecked)

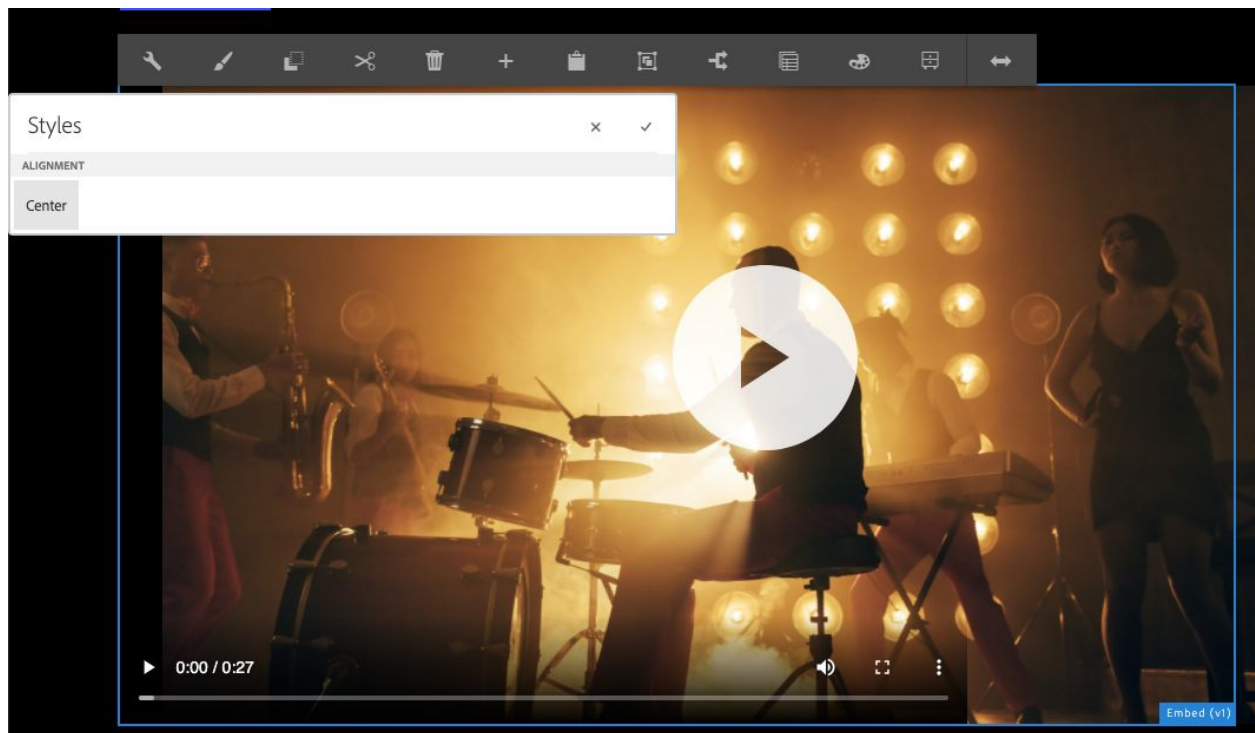
The embeddable has the following attributes.

1. Video Path
2. Poster Image
3. Height & Width
4. Autoplay ON/OFF
5. Loop On/OFF
6. Hide Controls ON/OFF

The Poster Image enables the author to select a specific image from DAM for the video as poster image for the video.

The play button is automatically displayed if the user selects video from DAM. To align the video in center author can select following from the style system drop down:

- Alignment -> Center



4.12. Spacing Classes

Every component in the WKND MUZIK project supports a global spacing concept. The authors or template editors can enable any of the following space classes via the style system to allow control over space.

The Spacer classes provide easy-to-understand concepts and code variables to apply and maintain space within and between components.

Tokens: (sm,md,lg,xl,xxl)

Inset (even padding all around)

An inset offers indents content on all four sides like the matte of the framed photo on a wall.

- space-inset--sm
- space-inset--md
- space-inset--lg
- space-inset--xl
- space-inset--xxl

Squish (padding 1:2)

A squished inset reduces space top and bottom, in our case by 50%. This is useful for buttons and pills where we need to increase space horizontally

- space-inset-squish--sm
- space-inset-squish--md
- space-inset-squish--lg
- space-inset-squish--xl
- Space-inset-squish--xxl

Stretch (padding 2:1)

Contrasted with a button or pill's squish helps with vertically stretching the insets of textboxes, textareas, and other form elements.

- space-inset-stretch--sm
- space-inset-stretch--md
- space-inset-stretch--lg
- space-inset-stretch--xl
- space-inset-stretch--xxl

Stack (margin bottom)

The stack spacing works by applying margin to the bottom.

- space-stack--sm
- space-stack--md

- space-stack--lg
- space-stack--xl
- space-stack--xxl

Inline (margin right)

The Inline spacing works by applying margin to the right side of the components, which is useful when we are trying to align things like buttons in the same line for example.

- space-inline--sm
- space-inline--md
- space-inline--lg
- space-inline--xl
- Space-inline--xxl