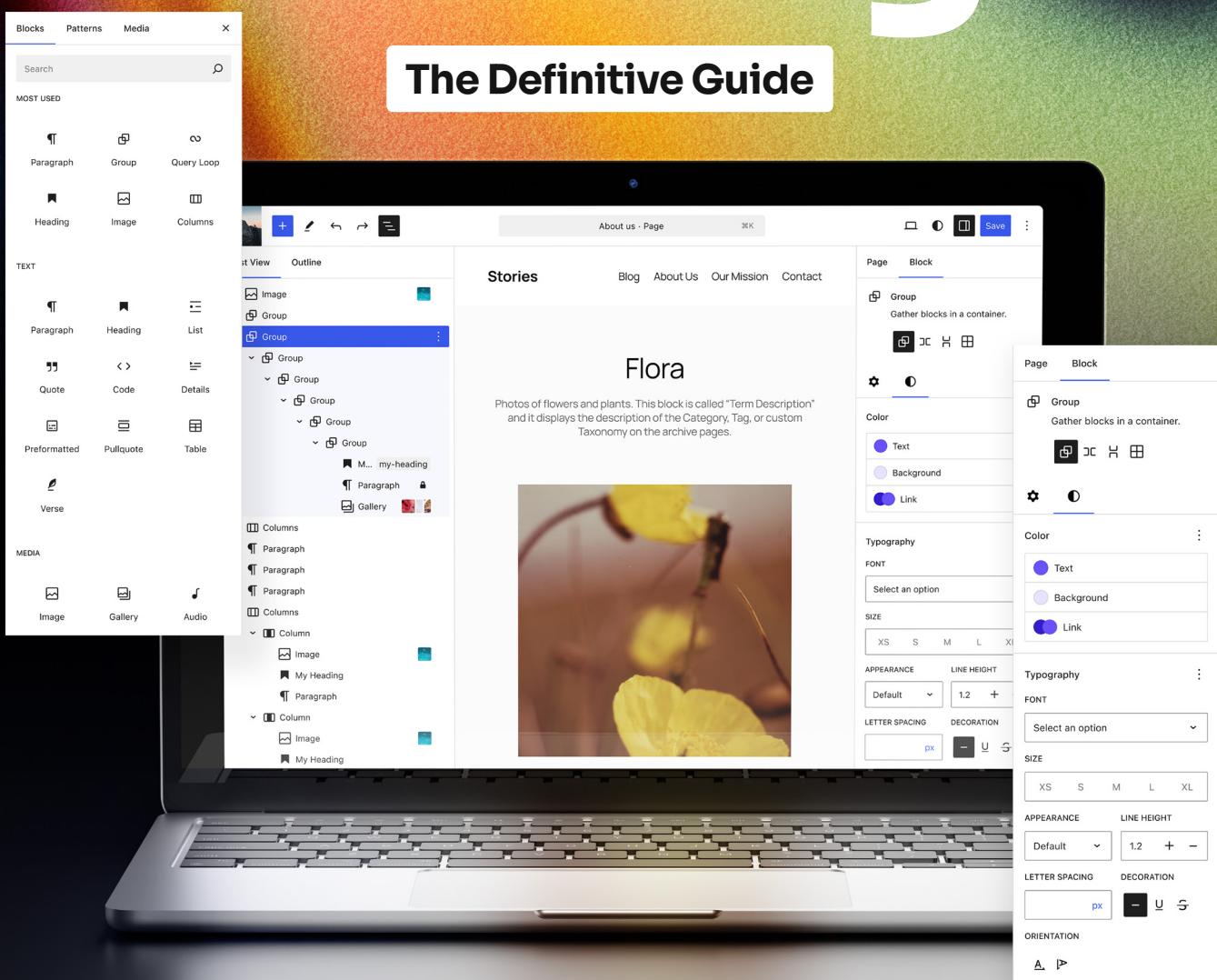


Human Made

WordPress Full Site Editing

The Definitive Guide



Everything enterprise teams need to plan,
implement, and scale Full Site Editing
with confidence

Contents

01	Introduction	5
02	A brief history of Full Site Editing	6
03	What is WordPress Full Site Editing?	7
04	Why Full Site Editing changes everything	8
	From page builders to platforms	8
	A new model for structured content	9
	Built for governance, scale, and speed	9
	Why this matters now	10
05	Understanding how Full Site Editing works: structure, hierarchy, and naming	11
	The FSE lexicon	12
	<i>Common confusion points</i>	13
06	Key features of Full Site Editing	14
	The Site Editor	14
	Block themes	14
	Global styles ('theme.json')	14
	Patterns and template parts	15
	Block-level control and permissions	15
07	How Full Site Editing solves common enterprise challenges	16
	Accelerating time to market	16
	Improving design consistency	16
	Common confusion points	17
	Reducing developer overhead	18
	Enabling scalable governance	18

Contents

08	Governance in Full Site Editing	19
	Granular control over content and design	20
	<i>Tips for enterprise governance</i>	21
	A flexible, secure foundation	21
09	Industry use cases	22
	Higher education	22
	Financial services	23
	Publishing and media	23
	Enterprise organisations	23
10	Technical and strategic benefits	24
11	Full Site Editing vs page builders: a strategic comparison	25
	<i>Why this matters for enterprise</i>	26
12	When WordPress Full Site Editing isn't the right fit	27
13	Implementation considerations	29
	Assess your existing architecture and workflows	29
	Align with your design system	30
	Plan for governance and permissions	30
	Review plugin and infrastructure compatibility	31
	Train and support your teams	31
	Develop a rollout strategy	32
	Measure success and adapt	32

Contents

14 The future of Full Site Editing: roadmap and opportunities	33
Advancing collaboration in the editor	33
Strengthening multilingual support	34
Extending with APIs and headless architectures	35
Scaling design systems	35
Centralised pattern orchestration and content modelling	36
Improving accessibility and performance	36
15 Common pitfalls to avoid	37
Lack of training	37
Too much flexibility too soon	38
Ignoring plugin compatibility	38
Skipping governance and permissions planning	39
16 Migrating to Full Site Editing	40
Audit your existing themes and content	41
Define your design system and theme structure	41
Build or adopt a block theme	41
Prepare patterns and template parts	42
Test editorial workflows and permissions	42
Roll out in phases	43
Monitor and iterate	43
17 Breaking up with shortcodes: migrating to blocks with confidence	44
Why blocks are a better long-term approach	44
What a shortcode-to-block migration actually looks like	45
From technical debt to modern foundations	45
18 Conclusion	47

1

Introduction

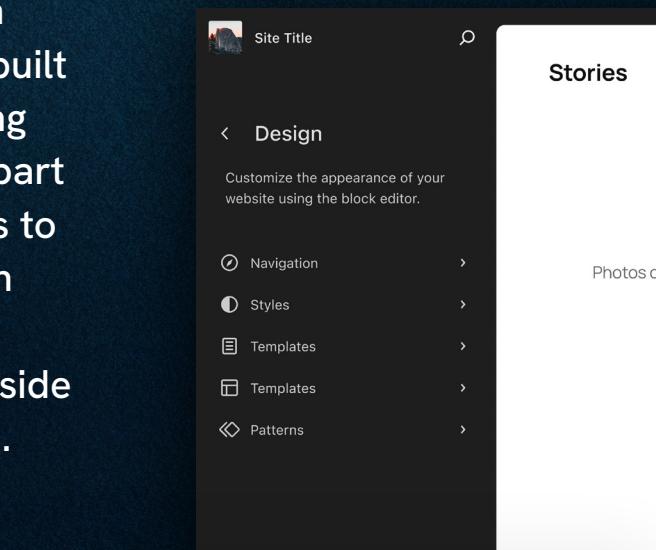
Enterprise websites must do more than publish content. They need to support marketing campaigns, product launches, localisation, compliance, accessibility, and brand consistency, often across multiple regions, teams, and stakeholders.

Traditional theming models can create friction. Design systems can be hard to enforce. Layout changes can require development time. And content teams can be dependent on developer workflows just to make simple structural edits.

Full Site Editing (FSE) represents a major shift in how WordPress sites are built and managed. By extending the block editor to every part of a site, FSE allows teams to create, modify, and govern design and layout using a visual interface directly inside the WordPress dashboard.

For enterprise teams, this opens the door to more agile workflows, better collaboration between content and design, and reduced reliance on developer time, as well as empowering teams to create immersive, high-quality digital experiences. FSE is a more scalable approach to governance and design system implementation.

This guide introduces WordPress Full Site Editing, explains how it works, and helps you understand where it fits within an enterprise-grade digital experience strategy.



All Pages			
Search			
Status	Author	Tag	Category
<input type="checkbox"/> LABEL	LABEL	LABEL	LABEL
<input type="checkbox"/> Hello World	Published	Published	⋮
<input checked="" type="checkbox"/> Hello World	Published	Published	⋮
<input type="checkbox"/> Hello World	Published	Published	⋮
<input type="checkbox"/> Hello World	Published	Published	⋮
<input type="checkbox"/> Hello World	Published	Published	⋮
<input type="checkbox"/> Hello World	Published	Published	⋮
<input type="checkbox"/> Hello World	Published	Published	⋮

WordPress Full Site Editing is the result of a multi-year evolution in how WordPress handles content and presentation. It began with the Gutenberg project in 2017, which introduced the block editor.

Initially focused on post and page content, the block editor allowed users to build structured content layouts using a visual interface, replacing the older TinyMCE editor.

From there, the WordPress community expanded the scope of blocks to include layout components, widgets, navigation, and eventually full site templates. This expansion culminated in the introduction of Full Site Editing in WordPress 5.9, released in January 2022.

With FSE, WordPress moved beyond content editing to offer complete visual control of a site's structure and style. Instead of editing PHP templates or theme files, users could now build headers, footers, and page layouts entirely with blocks. Full Site Editing was built to be modular, extensible, and deeply integrated with modern design and development practices.

● **2017**

Gutenberg project begins

WordPress kicks off the Gutenberg project, laying the foundation for block-based editing.

● **2018**

Block editor introduced

The block editor replaces TinyMCE in WordPress 5.0, enabling users to build content layouts visually.

● **2020**

Scope of blocks expanded

Blocks extend beyond posts and pages to cover widgets, navigation, and layout elements, preparing the ground for full-site control.

● **2022**

Full Site Editing released

WordPress 5.9 introduces Full Site Editing, allowing users to visually build headers, footers, templates, and control global styles using blocks.

● **2023**

Site Editor out of beta

WordPress 6.2 brings a major UI overhaul for the Site Editor, with improved navigation and block management.

● **2025**

The sky's the limit

Planned improvements include real-time collaboration, workflow controls, and more structured editing.

3

What is WordPress Full Site Editing?

WordPress Full Site Editing (FSE) refers to a set of features that enables users to build and manage the entire structure of a WordPress site - including templates, layout, navigation, and global styles - using blocks. It represents a move away from PHP-based templates and hardcoded themes toward a more flexible, declarative system built on HTML, JSON, and visual interfaces.

At the centre of FSE is the Site Editor, a user interface that allows changes to be made directly within WordPress, without editing files. Changes made in the Site Editor can affect the entire site, including templates for posts, pages, archives, and pages like 404 or search results.

FSE works with block themes - a new class of themes built entirely from blocks and configured using a file called '`theme.json`'. This file defines colours, typography, spacing, layout rules, and other style settings. It acts as a single source of truth for the site's visual identity and allows developers to implement design systems directly into the WordPress theming layer.



Site Title

Design

Customize the appearance of your website using the block editor.

 Navigation

 Styles

 Templates

 Templates

 Patterns

4

Why Full Site Editing changes everything

FSE represents a strategic shift in how content, design, and structure come together across platforms. For enterprise organisations rethinking their digital presence, FSE creates a new foundation for delivering content at scale: faster, more consistently, and with far greater control.

From page builders to platforms

In traditional WordPress builds, layout and content were often managed in separate systems. Design lived in PHP templates. Content was entered through the post editor. Branding guidelines were documented externally or enforced manually. Even in page builders, layout control often lived in a plugin interface, detached from the rest of the platform.

Full Site Editing replaces that fragmented approach with a unified system. Templates, blocks, patterns, and global styles now live in one place. Content editors work in the same environment as developers and designers. Everyone sees the same interface, follows the same design rules, and contributes to a platform that is cohesive by default.



This shift is really turning WordPress into a living representation of your design system, and it's the thing that I love most about FSE. It's immensely powerful as a tool for both developers and editorial teams to build within.



Rob O'Rourke,
Product Engineer, Human Made

A new model for structured content

With blocks, content becomes modular, reusable, and portable. Editors can build pages from components that are already aligned with your brand and strategy. Developers can version control those components like code. Designers can define how each element behaves using structured tokens. Every change is traceable, governed, and consistent.

This modular approach supports everything from multilingual publishing and campaign rollout through to headless delivery. Content is no longer locked to a layout or presentation - it becomes structured data that can move across channels and evolve according to need.

Styles



Aa :

[Browse styles](#)



Aa Typography

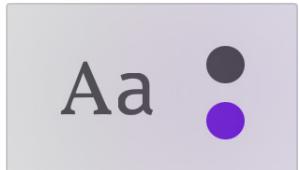
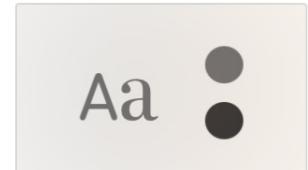
Colors

Background

Shadows

← [Browse styles](#)

Choose a variation to change the look of the site.



Built for governance, scale, and speed

Full Site Editing supports role-based permissions, locked layouts, shared pattern libraries, and global design configuration. These features offer the flexibility to scale content models without losing control. From launching microsites, to managing regional content, or rolling out campaigns across departments, fewer manual touchpoints means it can all be done faster.

FSE also reduces technical debt. Themes are lighter. Layout logic is simplified. Custom design systems can be encoded directly in theme configuration, not rebuilt with every new feature.

Why this matters now

Many enterprise teams are still working with a pre-block editor content model. Others are using WordPress in ways it was never originally intended, or migrating from other CMS platforms entirely.

Full Site Editing offers a clean break from those legacy constraints. It provides a future-ready platform that meets the needs of modern teams: collaborative, modular, governed, and fast.

FSE is not just a redesign tool. It is a complete rethinking of how a digital platform works, who can contribute, and how quickly teams can adapt.

All Pages

Search   1

Status is Published  Author Tag Category Add

 LABEL

<input type="checkbox"/>		Hello World
<input checked="" type="checkbox"/>		Hello World
<input type="checkbox"/>		Hello World
<input type="checkbox"/>		Hello World
<input type="checkbox"/>		Hello World
<input type="checkbox"/>		Hello World
<input type="checkbox"/>		Hello World
<input type="checkbox"/>		Hello World
<input type="checkbox"/>		Hello World
<input type="checkbox"/>		Hello World

 1 ITEM SELECTED   

5

Understanding how Full Site Editing works: structure, hierarchy, and naming

FSE introduces new concepts and terminology to WordPress. Terms like “patterns”, “template parts”, and “reusable blocks” can feel overlapping at first, especially when you’re used to classic themes or page builders.

This section breaks down the building blocks of Full Site Editing, showing how each piece fits into the larger structure. Whether you’re a developer, designer, or content editor, understanding this hierarchy will make FSE feel less abstract and more usable.

All patterns

A list of all patterns from all sources.

Search



Sync status

About the book

This project celebrates the power of a collection of photographs that capture the essence of craftsmanship and culture, reflecting the ingenuity and expression of each artist. Heidegger's evocative imagery strands a groundbreaking modernist approach, and Kōno's meticulous documentation of Japanese folk craftsmanship in a harmonious blend that allows the viewer to fully appreciate the artistry. *The Stories Book* is accompanied by insightful commentary, providing historical context and revealing the stories behind the photographs. The collection artfully evokes a sense of awe, a tribute to the power of technology to preserve and narrate the multifaceted experiences of humanity.



Banner with book description

Not synced



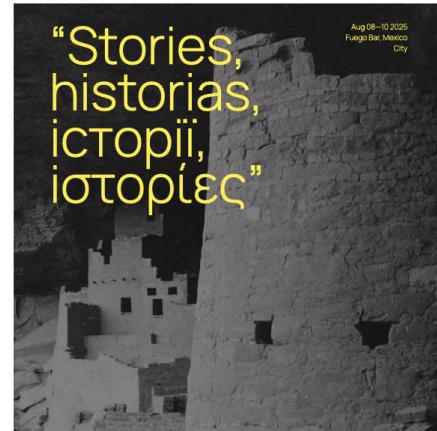
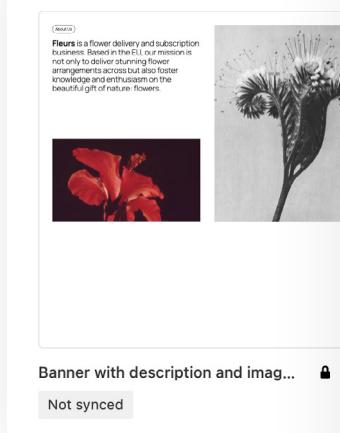
Cover with big heading

Not synced



Short hea

Not sync



The FSE lexicon

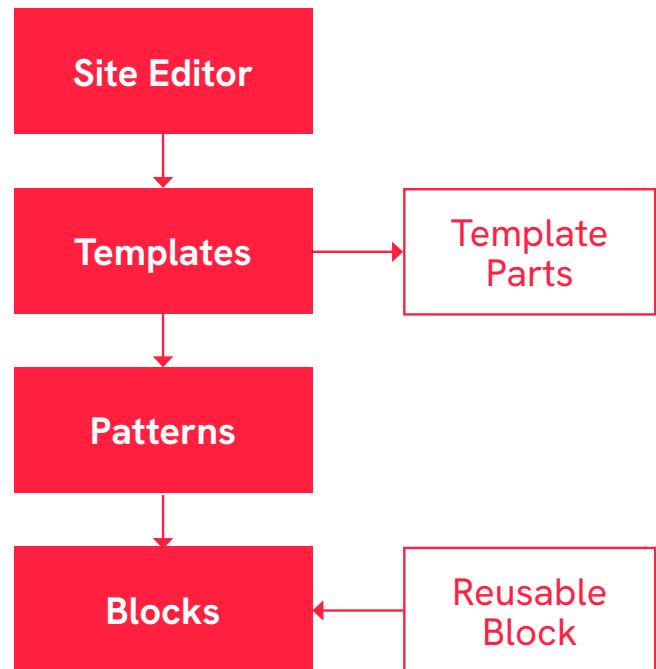
Here's how it all fits together, from the broadest structure to the smallest reusable unit:

Concept	What it is	Who uses it	Why it matters
Site Editor	The visual interface for editing templates and global styles	Editors, designers, developers	Provides a central place to manage structure and styling
Global styles	Controls for colours, fonts, spacing, and more, defined in <code>theme.json</code>	Designers, developers	Keeps branding consistent and design manageable
Templates	Layouts for entire pages, like homepage, single post, or archive	Developers, sometimes editors	Controls page-level layout
Template parts	Sections reused in templates, like headers, footers, or sidebars	Developers	Keeps structure consistent and maintainable
Patterns	Pre-defined layouts made from blocks, insertable anywhere	Editors and designers	Speeds up content creation
Reusable blocks	Specific block content saved for reuse across posts or pages	Editors	Good for CTAs, disclaimers, or snippets
Blocks	The smallest building units: paragraphs, images, buttons, etc.	Everyone	All content and layout is created with blocks

Common confusion points

▪ Patterns vs reusable blocks

Patterns are layout starting points. Once inserted, they can be edited without affecting the original. Reusable blocks are synced. If you change one, it updates everywhere it is used.



▪ Templates vs template parts

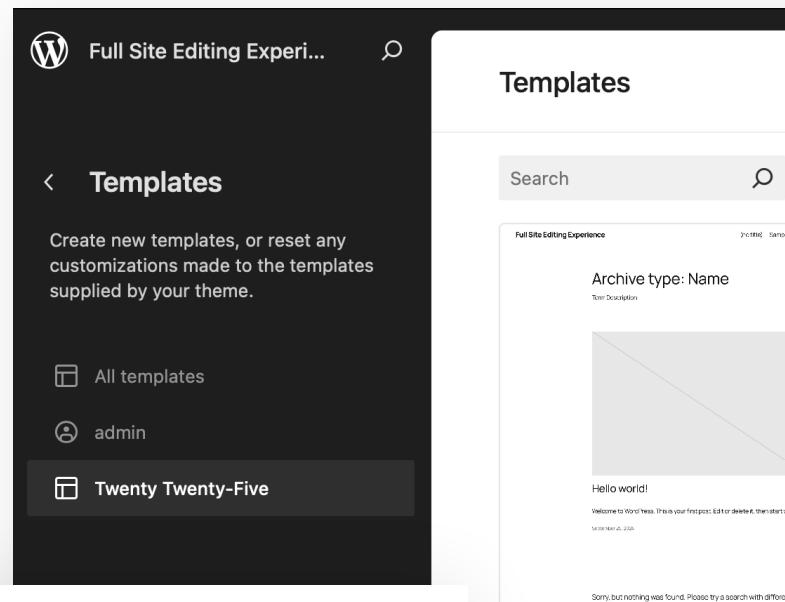
Templates define the layout for an entire page. Template parts are repeatable sections used inside templates.

▪ Global styles vs custom styles

Global styles come from the theme and apply site-wide. Custom styles can be added on a per-block or per-page basis.

▪ Patterns vs template parts

Patterns are editor-facing. Template parts are theme-facing. Use patterns to give editors flexibility, and use template parts to lock in structural elements.



6

Key features of Full Site Editing

The Site Editor

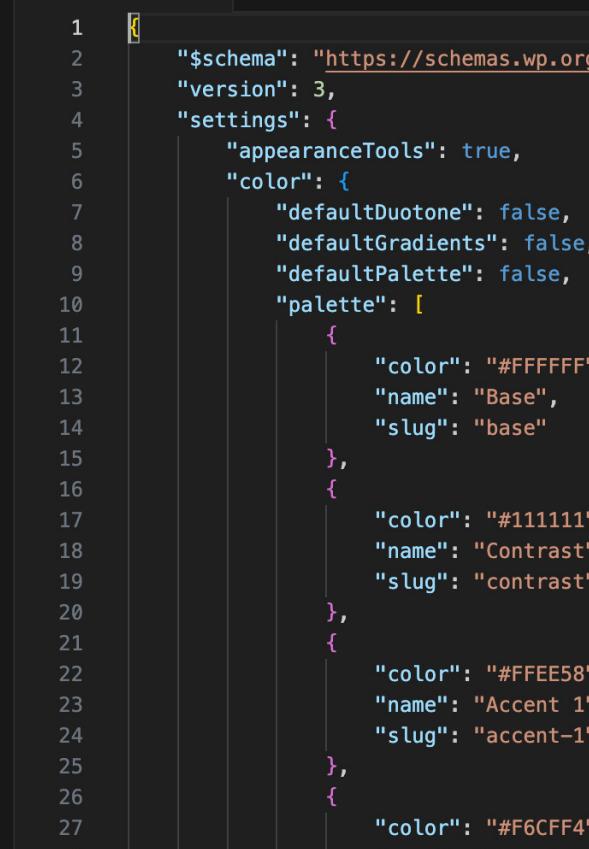
The Site Editor gives content and design teams the ability to manage layout and design from within the WordPress admin. It allows users to build or modify templates and template parts using the same block interface familiar from post and page editing. For example, users can update a homepage layout, create a new landing page template, or edit the global footer, all without touching code or leaving the WordPress dashboard.

Global styles (‘theme.json’)

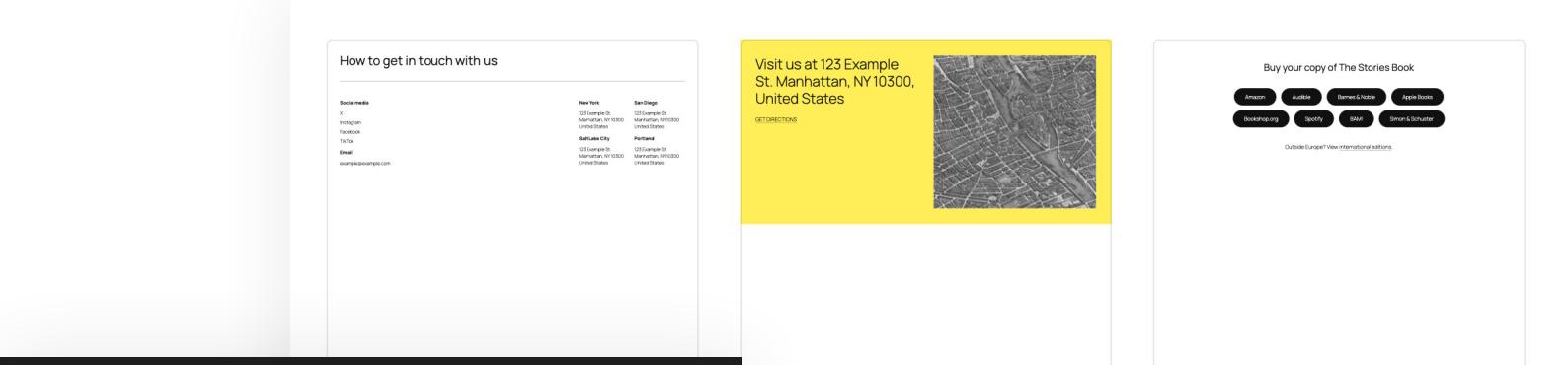
The `'theme.json'` file allows developers to define global design tokens and layout rules in a structured way. Colour palettes, font families, heading sizes, spacing, and more can all be specified, and all in a central location. These settings are applied consistently across blocks and templates and can be updated without editing CSS directly. This enables a scalable, version-controlled design system that works across large or distributed teams.

Block themes

Block themes replace the traditional PHP-based templating model with a more modular and structured approach. These themes are built using HTML files that define templates and template parts, while design and layout settings are controlled via a '`theme.json`' file. This model simplifies development, improves performance, and makes it easier to enforce design consistency across the site or network.



```
1  {
2      "$schema": "https://schemas.wp.org/wp/6.1",
3      "version": 3,
4      "settings": {
5          "appearanceTools": true,
6          "color": {
7              "defaultDuotone": false,
8              "defaultGradients": false,
9              "defaultPalette": false,
10             "palette": [
11                 {
12                     "color": "#FFFFFF",
13                     "name": "Base",
14                     "slug": "base"
15                 },
16                 {
17                     "color": "#111111",
18                     "name": "Contrast",
19                     "slug": "contrast"
20                 },
21                 {
22                     "color": "#FFEE58",
23                     "name": "Accent 1",
24                     "slug": "accent-1"
25                 },
26                 {
27                     "color": "#F6CFF4",
28                     "name": "Accent 2",
29                     "slug": "accent-2"
30                 },
31                 {
32                     "color": "#E8AABC"
33                 }
34             ]
35         }
36     }
37 }
```



Full Site Editing Experience

Patterns

All patterns	71
My patterns	1
About	7
Banners	19
Call to action	13
Contact	3
Featured	16
Footers	5
Gallery	1
Headers	5
Media	2
Pages	11
Post formats	3

✓ Saved

Patterns and template parts

Full Site Editing supports reusable layout components in the form of patterns and template parts. Patterns are predefined groups of blocks - such as a testimonial section or a product card layout - that can be reused throughout the site. Template parts are structural elements like headers and footers that are shared across multiple templates. Both help reduce repetition, speed up content creation, and ensure design consistency.

Block-level control and permissions

FSE supports tools for managing what blocks are available and how they can be used. Organisations can limit which blocks are visible to editors, lock certain parts of a layout, or restrict changes to specific global styles. These features are essential for enterprise governance, allowing teams to balance flexibility with control and enforce brand and compliance standards without micromanaging editorial teams.

7

How Full Site Editing solves common enterprise challenges

Accelerating time to market

Marketing and content teams can be reliant on developers to launch new pages or update layouts. This dependency can delay campaigns and slow down iterations. With Full Site Editing, non-technical users can use pre-approved blocks, patterns, and templates to make structural changes directly in the admin interface. This empowers teams to move faster without compromising design or compliance standards.

Improving design consistency

Maintaining a consistent brand experience across pages, departments, or regions is a common pain point in large organisations. Full Site Editing allows design systems to be implemented in code using the ‘theme.json’ configuration file. This ensures fonts, colours, spacing, and component styles are applied consistently across all parts of the site.

//

Designers and developers can work together in the WordPress design interface, quickly turning ideas into reality without running into implementation issues. Design patterns stay consistent across the site and can be reused wherever they’re needed, without being locked to one page or section.



Paweł Mikolajek,
Designer, Human Made

Common confusion points

However, consistency does not come from configuration alone: it comes from governance. By combining `theme.json` with block locking, pattern curation, and role-based permissions, organisations can control where flexibility is allowed and where it is not. Editors can be empowered to build rich layouts, while still working within a clearly defined system.

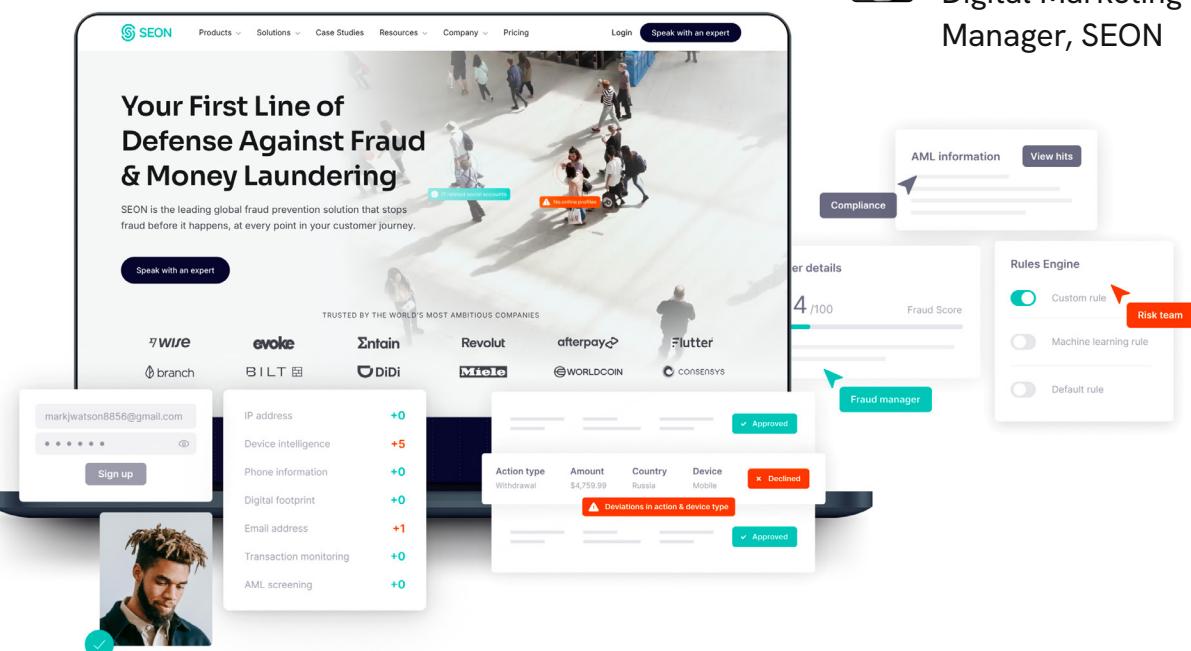
Learn how leading fraud prevention company SEON uses FSE to empower its marketing team.

[Read the case study →](#)

//

Full Site Editing and custom Gutenberg blocks ultimately made the site much more flexible and easier to manage.

 **Jonathan Pryer,**
Digital Marketing
Manager, SEON



Reducing developer overhead

Over time, theme customisation requests and minor layout changes can consume significant developer time. With Full Site Editing, developers create reusable components and design systems editors can use safely and independently, freeing up development resources for strategic projects.



This shift is brilliant for both engineers and clients. We can use development time to prioritise more complex work, and stakeholders don't have to wait for us to make small changes to things like colours.



Joe Bailey-Roberts,
Web Engineer, Human Made

Enabling scalable governance

FSE supports granular permissions, enabling enterprises to delegate content creation and layout editing while maintaining strong governance. You can restrict block use, lock template structures, and define who can change global styles - creating flexibility within boundaries.

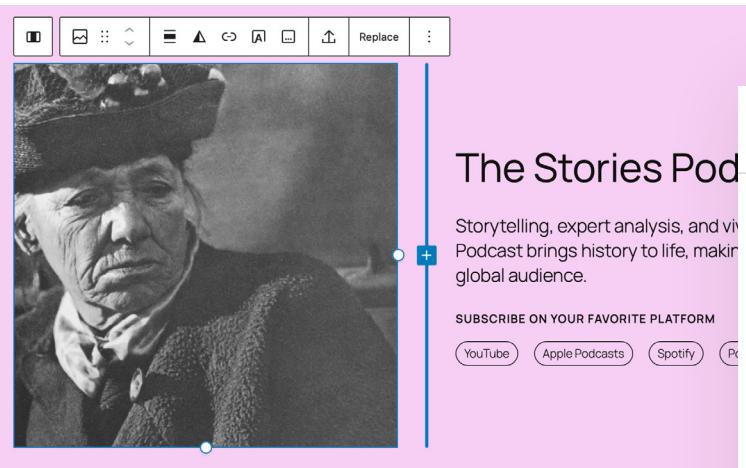
The screenshot shows the WordPress Full Site Editing interface. At the top, there's a toolbar with icons for adding blocks, alignment, and styling. Below that is a rich text editor with a large yellow text overlay containing the text "“Stories, historias, iCTOpīi, iστορίες”" on a grayscale background image of a canyon. The interface includes a color palette on the right side with tabs for "Pattern", "Block", and "X". The "Block" tab is selected, showing a "Color" section with "Text" and "Background" options, each with a circular icon and a color swatch. A note at the bottom right of the palette says: "This color combination may be hard for people to read. Try using a brighter background color and/or a darker text color." The bottom of the interface has a footer with the text "The Definitive Guide to WordPress Full Site Editing".

8

Governance in Full Site Editing

Governance is often the difference between a successful Full Site Editing rollout and a fragmented one. Enterprises need control over what can be changed, by whom, and in which contexts. WordPress offers a flexible permission model that supports fine-grained access control. With FSE, that control now extends to design systems, templates, and editorial layout tools.

This section outlines where and how permissions can be applied in FSE, helping your team strike the right balance between flexibility and oversight.



Styles ⋮ X

Layout ⋮

Dimensions ⋮

Set the width of the main content area.

CONTENT WIDTH

645 px

WIDE WIDTH

1340 px

PADDING

0 px

STYLES

Heading

Code is Poetry

STYLE VARIATIONS

Display

Subtitle

Granular control over content and design

With FSE, governance operates on multiple layers: from content blocks and patterns to templates and global styles. Each of these can be locked, scoped, or restricted based on user roles or custom capabilities.

Governance Area	What it controls	Who can manage it	How to restrict or allow
Blocks	Individual content elements like text, media, buttons	Editors, designers, admins	Use block locking (core) or plugins like Block Visibility
Reusable blocks	Synced blocks used across multiple posts or pages	Editors with <code>edit_reusable_blocks</code>	Limit creation/editing by role or disable for some roles
Patterns	Pre-defined block layouts that can be reused	Editors, designers, developers	Use pattern categories and restrict by user role or CPT visibility
Templates	Full-page layouts used across the site	Admins, developers	Lock templates in the theme, or limit editing via permissions
Template parts	Shared structural elements like headers or footers	Admins, developers	Use file-level locking or restrict via theme configuration
Global styles	Typography, colour palettes, spacing and layout rules	Admins, theme maintainers	Lock specific settings in <code>theme.json</code> , or hide from UI
Theme switching	Ability to change the active theme	Admins only	Core capability — typically locked to super-admins or devs
Custom block registration	Ability to create new blocks or extend existing ones	Developers only	Controlled via theme or plugin code



Pages

All Pages

- Published
- Scheduled
- Drafts
- Pending
- Private
- Trash

Tips for enterprise governance

▪ Use roles wisely:

WordPress' default roles (Admin, Editor, Author, Contributor, Subscriber) can be extended with custom roles for more nuanced control.

▪ Block what needs protecting:

Use block locking and block-level permissions to prevent accidental changes to critical elements like CTAs or brand assets.

▪ Govern at the theme level:

Define design tokens, layout rules, and access restrictions in your theme .json file to enforce consistency across teams and sites.

▪ Use environment-level permissions:

In larger organisations, restrict who can access the Site Editor in production, while allowing more freedom in staging.

Page	Author	Status
Embracing the Present: The Art ...	admin	Draft
Home	admin	Draft
Privacy Policy	admin	Draft
Sample Page	admin	Published
The Benefits of Mindfulness	admin	Published

Embracing the Present: The Art of Mindfulness

In our fast-paced world of constant attention, mindfulness offers a powerful ancient practice that can help us manage modern psychological challenges. By being fully present in the moment without judgment, we can reduce bodily sensations, increase focus, and find balance between stimulus and response. This can lead to reduced stress, improved mood, and a decrease in anxiety and depression. By incorporating mindfulness into our daily routines, we can enhance emotional resilience and overall well-being, allowing us to reclaim the present moment and live more fully.

A flexible, secure foundation

Full Site Editing does not mean giving up control. With the right configuration, enterprises can empower editors to move quickly while protecting core brand, layout, and technical standards. The key is to align governance with your team structure and workflows from day one.

9

Industry use cases

FSE benefits by industry



Higher Ed

- Centrally manage themes & design
- Departments can manage own content
- Improve accessibility & consistency
- Reduce technical maintenance



Financial Services

- Publish faster using locked templates
- Compliant design patterns
- Deploy design updates globally
- Manage site-level content locally



Publishing & media

- Dynamic story layouts
- Reuse editorial components
- More consistency across brands
- Deliver engaging content faster



Enterprise orgs

- Decentralised publishing with centralised framework
- Enforce brand & design standards
- Empower local teams to create content
- Streamline workflows

Higher education

FSE allows universities to provide centrally managed themes and design systems, while enabling departments and research units to manage their own content. Shared layouts and block patterns improve consistency and accessibility, and reduce technical maintenance.

Find out how the Harvard Gazette used Full Site Editing to enhance their users' digital experience

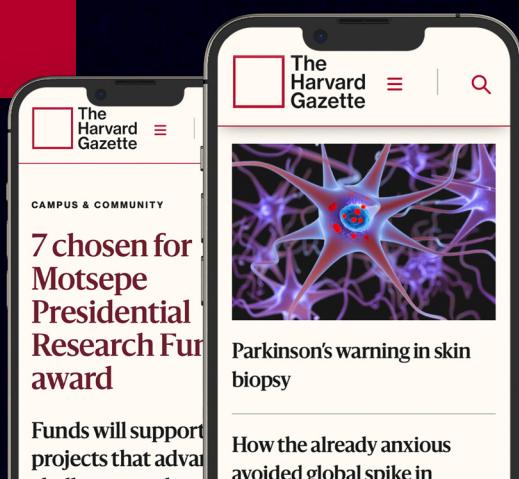
[Read the case study →](#)

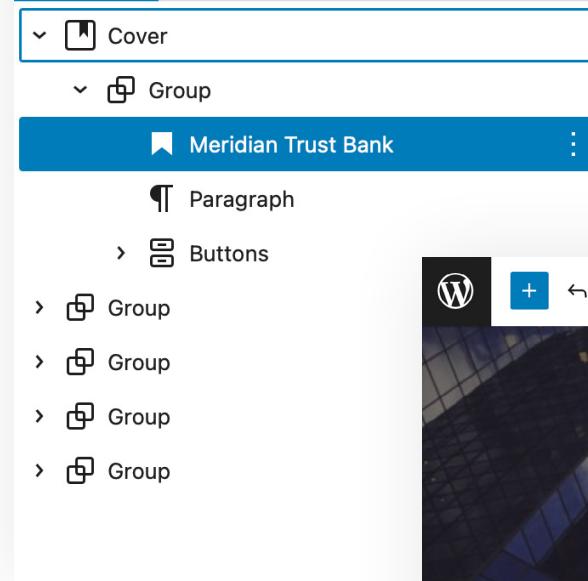


We went all-in on Full Site Editing, using the Gutenberg plugin to get new features that hadn't yet made it to WordPress core. Our goal was for every part of the Harvard Gazette WordPress site to be easy to edit by the internal team, and for the front end to be fully accessible to all users.



Joeleen Kennedy,
Senior Web Engineer, Human
Made, Technical Lead for Harvard
Gazette redesign project





Financial services

In regulated industries, compliance and speed must coexist. FSE allows teams to publish faster using locked templates and compliant design patterns. Design updates can be deployed globally, while site-level content is managed locally.

Publishing and media

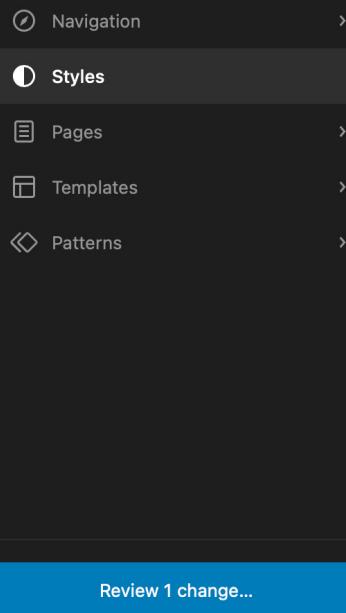
Editorial teams benefit from being able to create dynamic story layouts, reuse editorial components, and quickly spin up new sections. With block themes and patterns, publishers can deliver engaging content faster and with more consistency across brands.

Enterprise organisations

For enterprises operating across multiple teams, regions, or business units, FSE provides the structure needed to support decentralised publishing within a centralised framework. Shared patterns, global styles, and template locking help enforce brand and design standards, while allowing local teams to create content independently. Whether launching campaign microsites or managing internal content portals, FSE helps streamline workflows, reduce duplication, and maintain consistency across the organisation.

Design

Customize the appearance of your website using the block editor.



Styles

Style Book

← Typography

Available fonts, typographic styles, and the application of those styles.

TYPESETS

Aa	Aa	Aa
Aa	Aa	Aa
Aa	Aa	

FONTS

Manrope	1 variant
Roboto Slab	1 variant

ELEMENTS

Headings

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

HEADING 6

10

Technical and strategic benefits

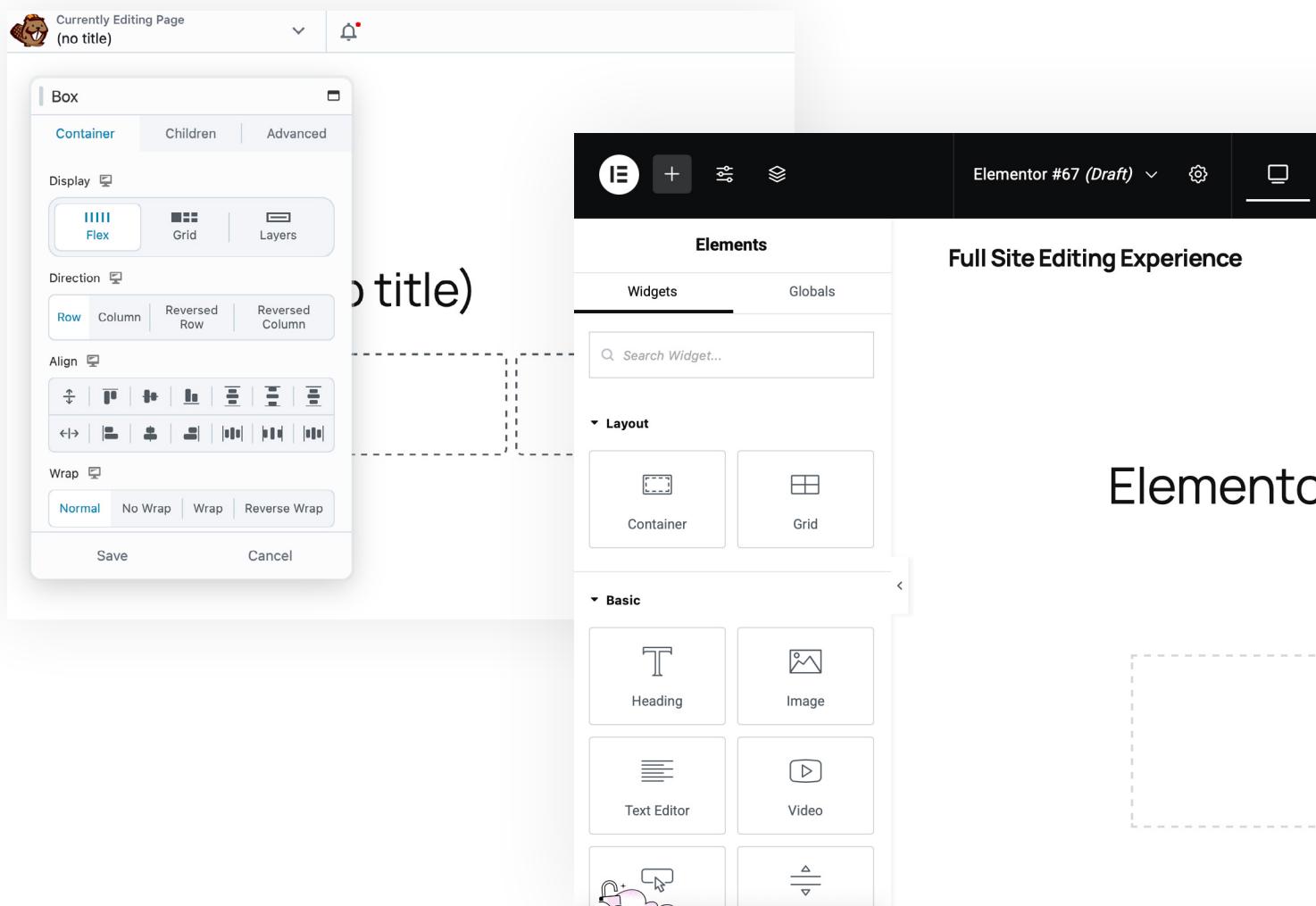
- **Modern theming architecture:** Move from PHP-based templates to structured HTML and JSON.
- **Version-controlled styling:** Store visual configurations in Git for better auditing and change management.
- **Design system enforcement:** Define tokens and styles in 'theme.json' to support consistent branding.
- **Better editorial experience:** Non-technical teams get more control without requiring developer support.
- **Future-ready platform:** FSE is core to WordPress' roadmap and continues to receive active investment.

11

Full Site Editing vs page builders: a strategic comparison

While traditional page builders like Elementor and Beaver Builder have made it easier for non-technical users to build pages in WordPress, they also come with limitations, especially at scale.

FSE is part of WordPress core. It is designed to work with modern theming practices, structured content, and native governance. For enterprise teams, this means fewer plugin dependencies, better performance, and more control over the editorial experience.



Here's how FSE compares to the leading page builders:

Feature / Capability	Full Site Editing	Elementor	Beaver Builder	WPBakery	Divi
Built into WordPress core	✓ Yes	✗ No	✗ No	✗ No	✗ No
Block-based and standards-aligned	✓ Native block model	⚠ Partial	⚠ Partial	✗ No	⚠ Partial
Global styles via design tokens	✓ theme.json driven	⚠ Limited	⚠ Limited	⚠ Limited	⚠ Basic
Performance-optimised output	✓ Lean markup	⚠ Moderate	⚠ Moderate	⚠ Moderate	⚠ Moderate
Accessibility by default	✓ Core-first focus	⚠ In progress	⚠ Limited	⚠ Limited	⚠ Limited
Works with WordPress Multisite	✓ Natively	⚠ Requires setup	⚠ Requires setup	⚠ Requires setup	⚠ Requires setup
Role-based permissions support	✓ Built in	⚠ Plugin-based	⚠ Plugin-based	⚠ Plugin-based	⚠ Plugin-based
DevOps and version control friendly	✓ Code-based themes	✗ Stored in DB			
Custom blocks and extensibility	✓ Uses WP APIs	⚠ Own API	⚠ Own API	⚠ Limited	⚠ Limited
Open source and vendor-independent	✓ Fully open	✗ Commercial	✗ Commercial	✗ Commercial	✗ Commercial
Roadmap aligned with WordPress future	✓ Yes	✗ No	✗ No	✗ No	✗ No

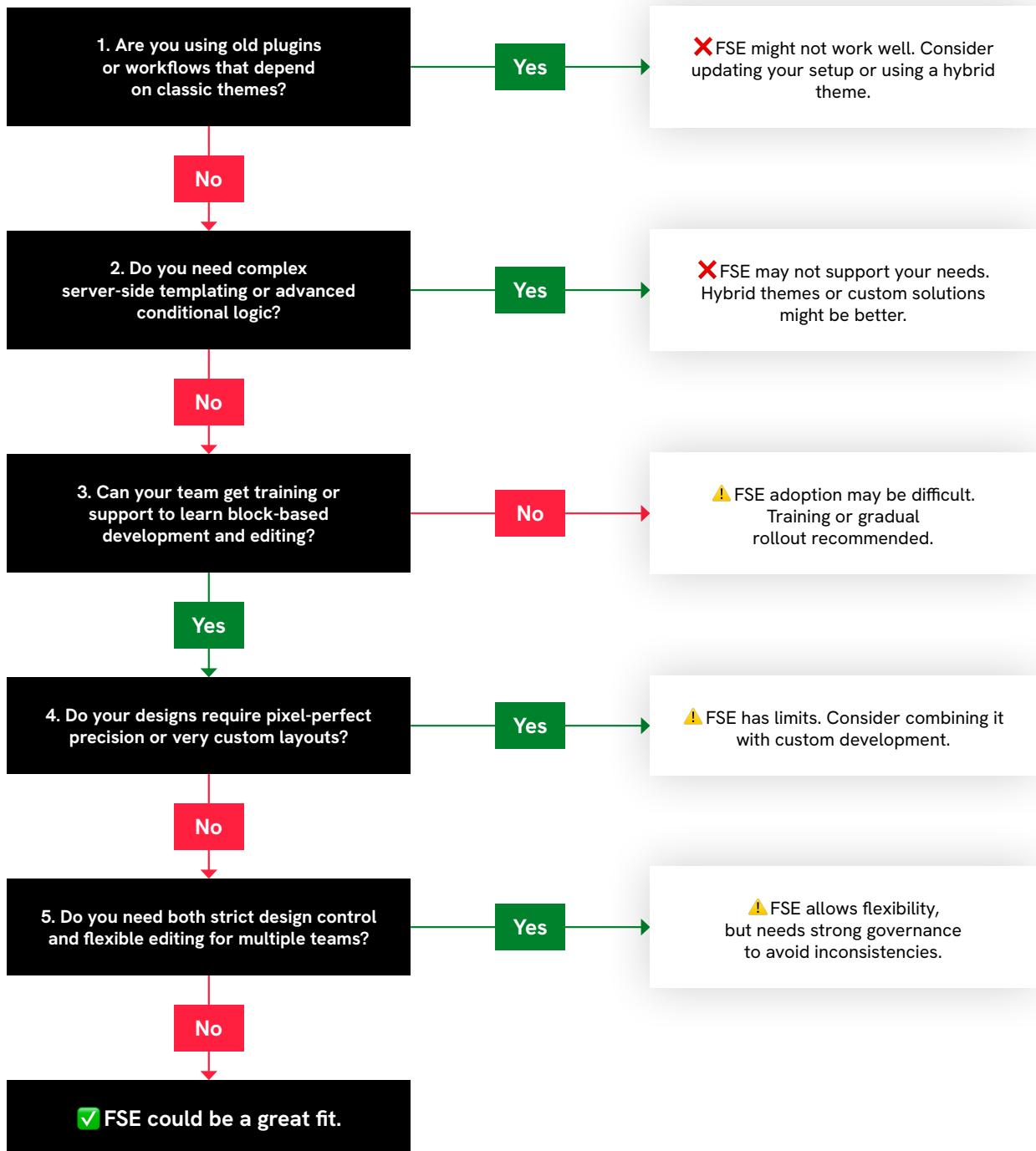
Why this matters for enterprise

FSE is being developed as the long-term future of WordPress. It is standards-driven, open source, and fully integrated into WordPress' roadmap. By contrast, page builders are third-party solutions that may limit flexibility, create performance bottlenecks, and introduce challenges around governance and maintainability at scale.

If you need enterprise-grade workflows, robust design systems, and scalable governance, Full Site Editing is the path forward.

12

When WordPress Full Site Editing isn't the right fit



The results

If you mostly answered **No** ➔ FSE is likely to be the right choice for you!

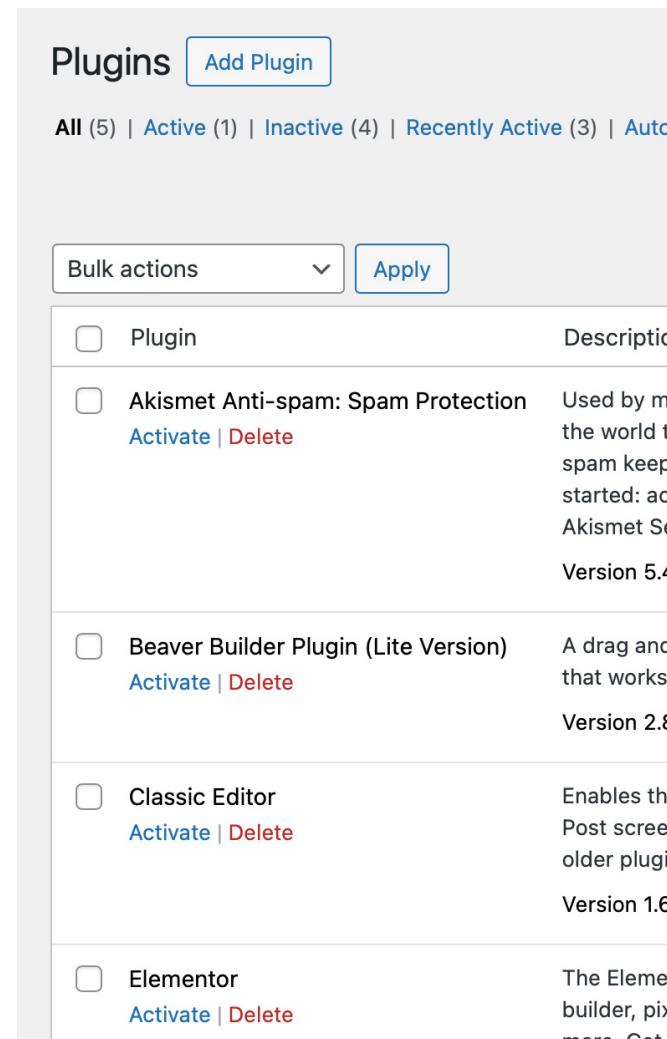
If you answered **Yes** a few times ➔ FSE might still work with the right planning.

If you answered **Yes** to most ➔ A hybrid theme could be a better choice for now.

Full Site Editing may not suit every scenario. It might not be the right solution if:

- Your site uses **outdated plugins** or workflows that depend on classic themes.
- You rely on **advanced server-side templating** or conditional logic not supported in block templates.
- You don't have the **necessary training resources** available to help teams adopt block-based development and editorial practices.
- You require **extreme visual precision** or creative layouts not currently achievable through FSE's constraints.
- You need **strict visual governance**, but your teams require a **high level of flexibility** in the editor. Without clear roles, controls, and design system enforcement, FSE's flexibility can introduce inconsistency across teams or regions.

In these cases, hybrid approaches - such as using classic themes with the block editor - may offer a good transitional path.



The screenshot shows the WordPress Plugins dashboard. At the top, there are buttons for 'All (5)', 'Active (1)', 'Inactive (4)', 'Recently Active (3)', and 'Automatic'. Below this is a 'Bulk actions' dropdown and an 'Apply' button. The main area lists five plugins: 'Akismet Anti-spam: Spam Protection' (version 5.4.1), 'Beaver Builder Plugin (Lite Version)' (version 2.8.1), 'Classic Editor' (version 1.6.1), and 'Elementor' (version 3.5.1). Each plugin entry includes an 'Activate' and 'Delete' link.

Edit Themes

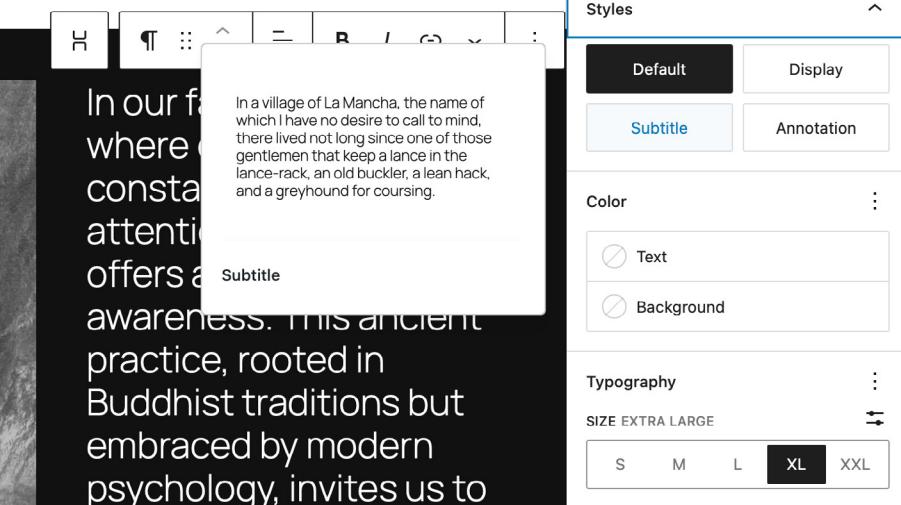
Editing Twenty Twenty-Five (active)

File: style.css

Selected file content:

```
1 /*  
2 Theme Name: Twenty Twenty-Five  
3 Theme URI: https://wordpress.org/themes/twentytwentyfive  
4 Author: the WordPress team  
5 Author URI: https://wordpress.org  
6 Description: Twenty Twenty-Five emphasizes simplicity and a flexible design options, supported by a variety of pattern types, such as services and landing pages, making it ideal for blogs, professional portfolios, online magazines, or business templates cater to various blog styles, from text-focused to image-rich. Additionally, it supports international typography and ensures accessibility and customization for users worldwide.  
7 Requires at least: 6.7  
8 Tested up to: 6.8  
9 Requires PHP: 7.2  
10 Version: 1.2  
11 License: GNU General Public License v2 or later  
12 License URI: http://www.gnu.org/licenses/gpl-2.0.html  
13 Text Domain: twentytwentyfive  
14 Tags: one-column, custom-colors, custom-menu, custom-logo, images, full-site-editing, block-patterns, rtl-language-threaded-comments, translation-ready, wide-blocks, block
```

Present: The Art of



Add template

Select what the new template should apply to:

A sidebar titled "Add template" showing options for applying a new template. It includes icons and labels for "Front Page", "Pages", "Category Archives", "Date Archives", and "Single item: Post".

13

Implementation considerations

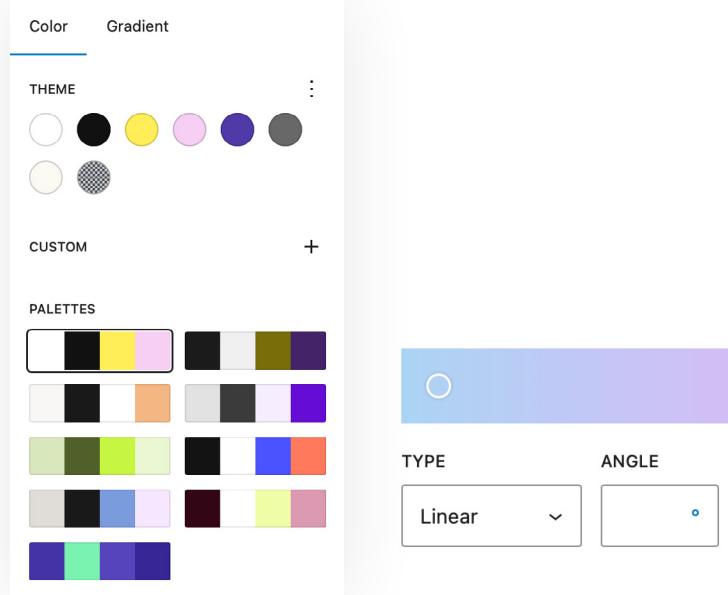
Implementing WordPress Full Site Editing is not just a technical upgrade. It is a significant shift in how your organisation approaches site building, theming, and content management.

Like any platform-level change, success depends on planning, internal alignment, and choosing the right approach for your team structure and strategic goals. Here are the key areas to consider before making the move.

Assess your existing architecture and workflows

Begin with an audit of your current WordPress implementation. Review how your themes are structured, what technologies are in play, and how content is currently created and managed.

If you rely heavily on hardcoded templates or visual page builders, those patterns will influence your migration approach. Pay attention to how developers and content teams interact. Understanding these workflows will help you identify friction points that Full Site Editing could alleviate and areas where the new system may require adaptation.



Align with your design system

If you already have a formal design system in tools like Figma, your next step is translating it into WordPress. Full Site Editing enables you to define visual rules such as colours, typography, and spacing within `theme.json`, meaning design tokens can be enforced at the platform level.

If you do not yet have a design system, FSE offers a practical reason to create one. Start with the foundational elements and grow from there. The more structured your design system, the easier it will be to maintain consistency across templates, blocks, and patterns.

Contrast

#111111

THEME

✓

Clear

Plan for governance and permissions

Full Site Editing gives content teams more control, but that freedom must be guided by thoughtful governance. You will need to define which user roles are allowed to make structural changes, edit templates, or adjust global styles.

WordPress offers role-based capabilities out of the box, and these can be extended further with plugins or custom logic. Plan how governance will work in practice, not just on paper. Who can create patterns? Who is responsible for enforcing design rules? Establishing this early will prevent misuse and reduce the risk of fragmentation.

The Definitive Guide to WordPress Full Site Editing

30

A commitment to innovation and sustainability

Etudes is a pioneering firm that seamlessly merges creativity and functionality to redefine architecture excellence.

Learn More Stories

Blog About Us Our Mission Contact

Active: Twenty Twenty-Five Customize

Review plugin and infrastructure compatibility

Many legacy plugins were built with classic themes in mind and may not yet fully support block themes or the Site Editor. Before committing to FSE, test your plugin stack in a staging environment running a block theme.

Pay particular attention to any plugins that affect layout or design output. You should also review your infrastructure setup. Ensure your deployment processes, CI/CD pipelines, and staging environments are ready to handle the version-controlled nature of modern WordPress theming. A well-aligned infrastructure will help prevent bottlenecks during rollout.

Train and support your teams

Even the most elegant system will fail without proper onboarding. Content editors, designers, and developers will all interact with Full Site Editing differently and need tailored support. Provide documentation that explains not just the how, but the why behind your patterns and governance decisions. Host workshops, offer Q&A sessions, and make space for hands-on experimentation.

The more confident your teams are in using the system, the smoother the adoption process will be. Consider building a network of internal champions who can help field questions and encourage best practice.

Edit Plugins

Editing Secure Custom Fields (active)

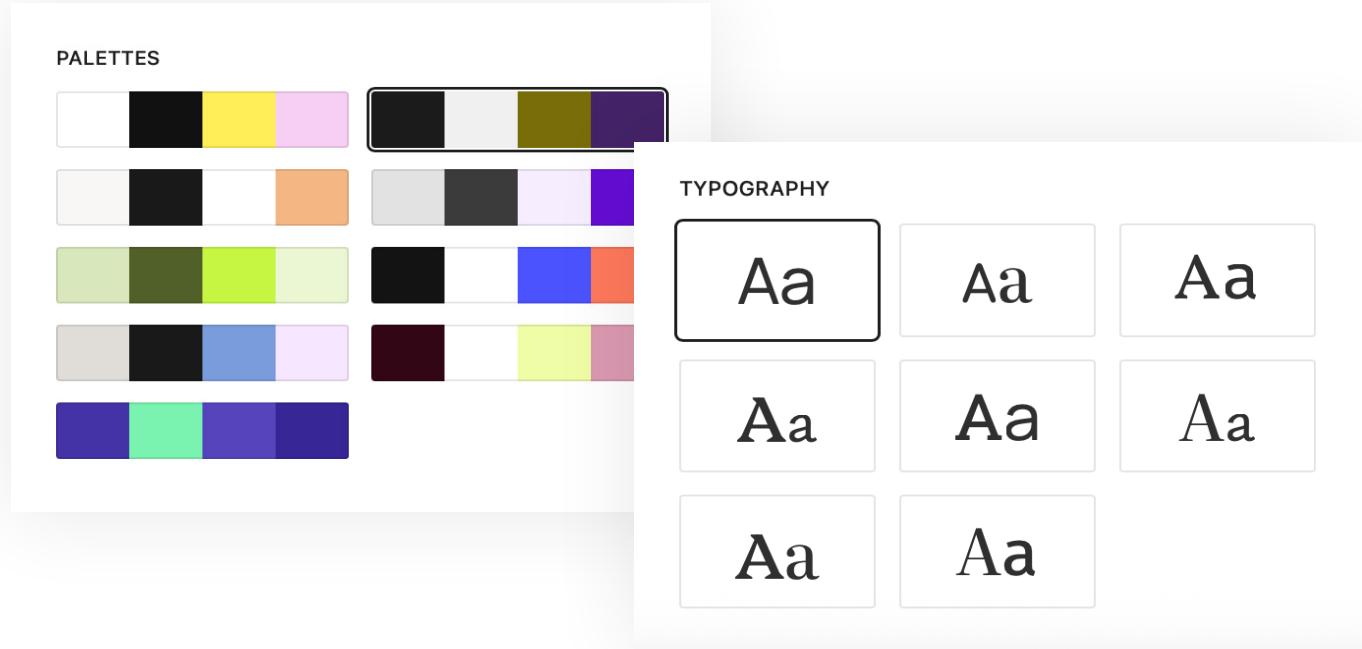
File: `secure-custom-fields/secure-custom-fields.php`

Selected file content:

```

6  * Plugin Name:      Secure Custom Fields
7  * Plugin URI:     https://developer.wordpress.org/
8  * Description:     Secure Custom Fields (SCF) offers developers to enhance WordPress content management by providing options without coding requirements.
9  * Version:         6.4.2
10 * Author:          WordPress.org
11 * Author URI:     https://wordpress.org/
12 * Text Domain:    secure-custom-fields
13 * Domain Path:   /lang
14 * Requires PHP:   7.4
15 * Requires at least: 6.0
16 *
17 * @package wordpress/secure-custom-fields
18 */
19
20 if ( ! defined( 'ABSPATH' ) ) {
21     exit; // Exit if accessed directly.
22 }
23
24 if ( ! class_exists( 'ACF' ) ) {

```



Develop a rollout strategy

Full Site Editing does not need to be implemented across your entire digital estate all at once. In fact, a phased approach is often more successful. Start with a pilot site or microsite where you can test your theme structure, governance model, and editorial workflows.

Use the insights from this pilot to refine your approach, then gradually expand to other sites, regions, or business units, depending on your organisation's structure. This method reduces risk, allows for iteration, and provides early success stories to build momentum internally.

Measure success and adapt

Finally, define what success looks like and build in mechanisms to measure it. This could include time-to-publish metrics, user satisfaction, brand consistency, or development velocity.

Gather feedback regularly from all user types. Are content teams empowered or confused? Are developers being pulled into fewer minor changes? Use this data to iterate. Full Site Editing is not a one-off implementation. It is a new way of working that will continue to evolve, and your platform should evolve with it.

14

The future of Full Site Editing: roadmap and opportunities

Full Site Editing has already redefined how WordPress themes are designed, structured, and managed. But this is only the beginning. The WordPress project continues to evolve FSE, with a development roadmap focused on enterprise-level priorities including better collaboration, multilingual publishing, design system enforcement, and extensibility.

Advancing collaboration in the editor

A key area of development is collaborative editing, and work is already underway to improve this experience. Future iterations are expected to include features such as user presence indicators, non-blocking editing, and version history that will allow teams to collaborate more effectively.

These improvements will bring WordPress closer to modern editorial tools and make the platform even more suitable for distributed teams working together at scale.

Active Projects

Features that are current areas of focus. You are encouraged to participate in them.

ACTIVE ongoing

[Site Editor/Templates Experience \[Phase 2\]](#)

The Site Editor/Template experience allows you to edit all parts of your site with blocks by offering access to various templates, a Styles system, navigation features, and more. The largest amount of remaining work is around clarifying the user experience of editing global concepts (styles, templates, navigation) from specific site content, [aligning page editing features in the Site Editor and Post Editor](#), and ensuring simple site setup actions are properly surfaced.

ACTIVE ongoing

[Block Themes \[Phase 2\]](#)

Strengthening multilingual support

Multilingual content is critical for global organisations. While native multilingual support is not yet part of WordPress Core, the FSE roadmap includes foundational changes to improve compatibility with translation workflows, content variation, and language-aware design systems.

FSE already works effectively in multilingual setups when paired with [WordPress Multisite](#). Each language can be served on a separate site, with fully localised templates, content, and team access. This provides clarity and control at a regional level while supporting global consistency.

Tools like [MultilingualPress](#) allow content relationships to be maintained across languages without forcing strict mirroring. As WordPress evolves, we expect deeper support for multilingual content at the template and block level, making it easier for global teams to manage localisation within a unified system.

MultilingualPress

MultilingualPress: The right way to build a multilingual WordPress site.

Run each language in a separate site of your WordPress multisite and connect the content in a lightweight user interface.

Get it now!



MultilingualPress

Scalable

MultilingualPress integrates with WordPress Multisite, ensuring scalability. It optimizes the core by using the powerful WordPress core.



Seamless SEO for Multilingual Sites

Boost multilingual SEO with automatic hreflang support and language redirection, perfectly aligned with Google's standards.

```
POST /wp-json/wp/v2/posts/42 HTTP/1.1
Host: example.com
X-HTTP-Method-Override: DELETE
```

```
HTTP/1.1 200 OK
```

```
{
  "status": 302,
  "headers": {
    "Location": "http://example.com/wp-json/wp/v2/us
  },
  "body": {
    "id": 42
  }
}
```

Extending with APIs and headless architectures

FSE introduces a more structured and predictable approach to theming. This structure can be exposed and used programmatically through the WordPress REST API or with GraphQL. Enterprise teams that need headless delivery, multi-channel publishing, or frontend frameworks can build custom interfaces while still benefiting from FSE's editorial workflow and design system integration. Patterns, templates, and theme configuration are increasingly stored in standardised formats such as JSON, which can be consumed by other systems.

The future of FSE includes more programmatic access to these components. This will make it easier to manage theme structures as part of a broader digital experience platform and support use cases that extend far beyond traditional websites.

Scaling design systems

Enterprises rely on design systems to ensure consistency across brands, products, and markets. FSE offers a direct way to implement these systems in WordPress using the `theme.json` configuration file. This file acts as a single source of truth for global styles, layout rules, and design tokens.

The FSE roadmap includes continued enhancements to how design rules are declared and enforced. This will support more advanced visual standards, responsive behaviour, and layout logic. These improvements will make it easier to translate design systems from tools like Figma into working WordPress themes without sacrificing control or scalability.

For multi-brand and multi-region organisations, this helps maintain visual consistency across a large estate of content and websites while still allowing room for local flexibility.

The combination of colors used across the site and in color pickers.

Color Gradient

THEME



:

Centralised pattern orchestration and content modelling

Content teams increasingly rely on reusable patterns to move quickly and maintain consistency. Today, patterns can be created and used across templates and pages, but managing them at scale requires custom workflows.

Future versions of FSE are expected to introduce more advanced tools for organising, categorising, and orchestrating patterns across entire websites or networks. This means teams will be able to define a pattern once and deploy it everywhere, or update a layout structure globally without touching every page manually.

This shift toward content modelling, where layouts are treated as structured data rather than static templates, will enable greater reuse, faster iteration, and safer collaboration. For large-scale publishing operations, this becomes a critical advantage.

Improving accessibility and performance

Enterprises in regulated industries must meet strict accessibility and performance standards. WordPress core contributors are working to improve FSE's ability to support these requirements out of the box. Updates to how blocks are rendered, how CSS is generated, and how templates are constructed will help sites meet [WCAG](#) compliance and achieve higher performance scores.

Future improvements may include accessibility validation within the Site Editor itself, more semantic block markup, and better tooling for visual regression testing. These changes will allow teams to deliver faster and more inclusive digital experiences without relying on third-party plugins or custom overrides.



Full Site Editing Experi...



Patterns

Manage what patterns are available when editing the site.

◀ All template parts

7

Header

3

Footer

3

General

1

As the WordPress project continues to mature FSE, its value to enterprises will only increase. Teams that adopt it today are positioning themselves for faster workflows, stronger brand control, and deeper integration with their wider digital ecosystem.

15

Common pitfalls to avoid

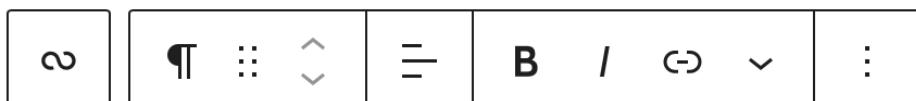
While WordPress Full Site Editing introduces a modern and more efficient way to build and manage enterprise websites, its successful adoption depends on more than just activating a block theme.

Many of the most common issues stem from misunderstanding the scope of change FSE introduces, or underestimating the strategic planning required. Here are some of the pitfalls we recommend avoiding, based on real-world enterprise implementations.

Lack of training

One of the biggest risks during an FSE rollout is assuming that existing WordPress users will intuitively adapt to the new editing experience. The Site Editor, block interface, and global styles require a different mindset and workflow than classic theming.

Without structured onboarding, users may become frustrated, rely on outdated processes, or make changes that disrupt design consistency. It's essential to provide documentation, hands-on training, and clear guidance for both technical and editorial users to ensure adoption and consistency.



Sorry, but nothing was found. Please try a search with different keywords.

Too much flexibility too soon

FSE offers powerful tools for visual editing and layout customisation, but without clear guardrails, teams may inadvertently introduce inconsistencies or bypass design system rules. Unrestricted access to all blocks, patterns, and style options can quickly lead to brand fragmentation and technical debt.

To avoid this, we recommend starting with a tightly scoped block set, pre-approved patterns, and locked templates. This helps teams learn the system within safe boundaries, while governance models mature over time.

TYPOGRAPHY

Size	✓
Font	✓
Appearance	✓
Line height	✓
Letter spacing	✓
Decoration	✓
Orientation	✓
Letter case	✓
Drop cap	✓

Ignoring plugin compatibility

Not all plugins are fully compatible with block themes or the Site Editor. Many legacy plugins were built for classic themes and may not render correctly or provide a seamless experience within block-based templates.

Before migrating to FSE, conduct a thorough audit of your plugin stack. Test key functionality in a staging environment using your block theme to identify any gaps, and seek alternatives where needed. Incompatibility issues caught post-migration can be costly to fix and may result in functionality loss or visual issues on live pages.

Skipping governance and permissions planning

Just like with WordPress Multisite, governance is critical to scaling safely with FSE. Without clearly defined roles, responsibilities, and editorial workflows, you risk inconsistent experiences across the network and increased maintenance overhead.

Establishing who can create or modify patterns, update global styles, or publish new templates is essential. Use role-based access controls, block permissions, and documented processes to make governance part of your platform from the outset: not something that's added in later under pressure.

<input type="checkbox"/>	Username 
<input type="checkbox"/>	 admin
<input type="checkbox"/>	 ehane
<input type="checkbox"/>	 haag.lavern
<input type="checkbox"/>	 huel.garrison
<input type="checkbox"/>	 jesus.daugherty
<input type="checkbox"/>	 legros.jonatan
<input type="checkbox"/>	 lind.deanna

16

Migrating to Full Site Editing

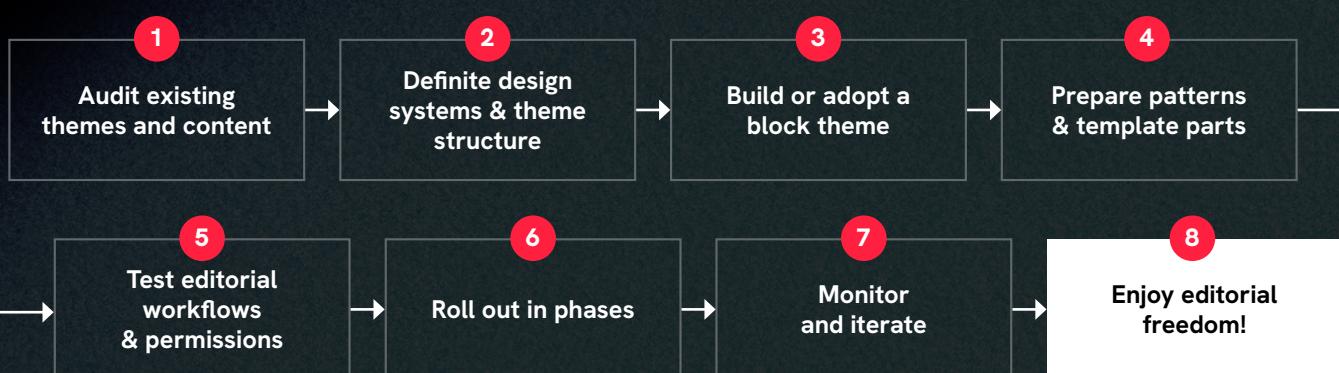
Migrating to Full Site Editing is a strategic opportunity to modernise your website architecture, improve design system adoption, and streamline content workflows. But like any major platform transition, it requires careful planning, cross-functional collaboration, and phased execution. The most successful migrations treat FSE not just as a new feature, but as a foundation for long-term transformation.

Audit your existing themes and content

Start by auditing your current WordPress themes and templates. Identify how your pages are constructed today: what templates are in use, where content structure varies, and which components are reused across the site.

Pay close attention to legacy code, custom shortcodes, and plugins that manipulate layout or content output. These will be the areas that may require the most rework or replacement. This stage is also a good time to assess accessibility, performance, and branding consistency across your estate.

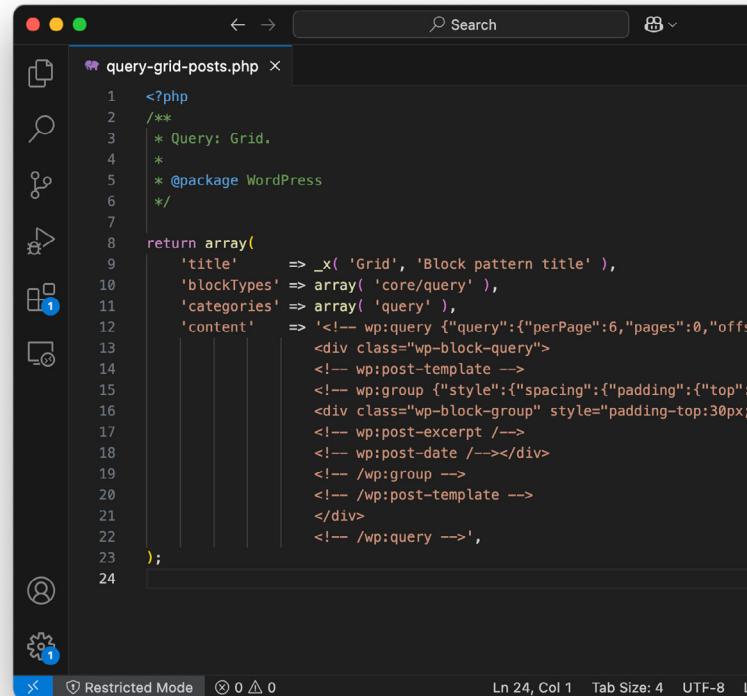
8 steps to migrate to FSE



Define your design system and theme structure

Full Site Editing works best when built around a structured design system. Before writing any code, define your visual foundation: colours, typography, spacing, breakpoints, and layout behaviours. Document these rules and translate them into a `theme.json` configuration file.

Decide what templates and template parts your theme will need - for example, homepage, landing page, archive, 404, and single post templates - and how much flexibility individual editors will have to modify them. Think of this step as translating your brand guidelines into code.



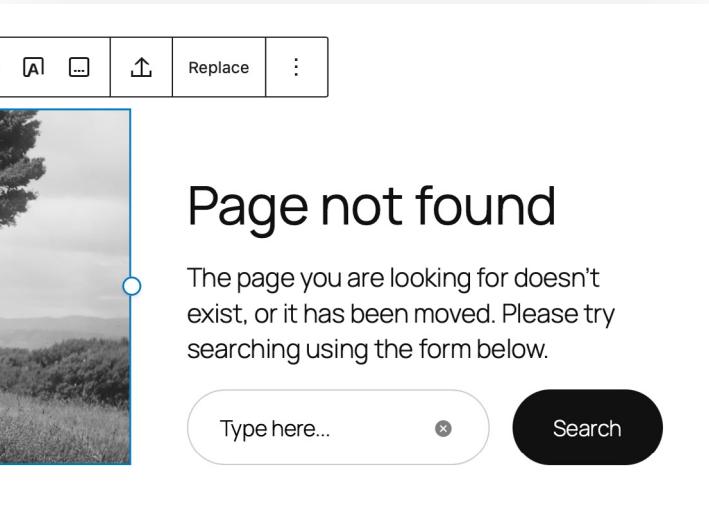
```
query-grid-posts.php x

1  <?php
2  /**
3   * Query: Grid.
4   *
5   * @package WordPress
6   */
7
8  return array(
9      'title'      => _x( 'Grid', 'Block pattern title' ),
10     'blockTypes' => array( 'core/query' ),
11     'categories' => array( 'query' ),
12     'content'    => '<!-- wp:query {"query":{"perPage":6,"pages":0,"offset":0,"order":"desc","orderby":"date","category":0,"showposts":6,"tax_query":[]}} -->
13     <div class="wp-block-query">
14         <!-- wp:post-template -->
15         <!-- wp:group {"style":{"spacing":{"padding":{"top":30,"bottom":30}}}>
16             <div class="wp-block-group" style="padding-top:30px; padding-bottom:30px">
17                 <!-- wp:post-excerpt /-->
18                 <!-- wp:post-date /--></div>
19                 <!-- /wp:group -->
20                 <!-- /wp:post-template -->
21             </div>
22         <!-- /wp:query -->',
23     );
24 );
```

Build or adopt a block theme

With your design system and template needs defined, you can begin development of your block theme. You may choose to build a custom theme from scratch, fork an existing open-source block theme, or work with an agency partner.

Use semantic HTML templates, and structure your `theme.json` for scalability and clarity. Reusable patterns and locked template parts should be included early in the process to help guide the editorial experience. Keep your theme lightweight, modular, and ready to evolve over time.





Blocks

Patterns

Media

x

Search



All

My patterns

Starter content

About

Banners

Call to action

Contact

Featured

Footers

Gallery

Headers

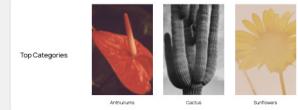
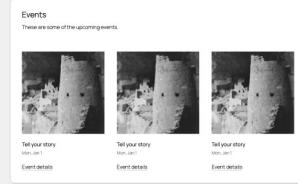
Media

Pages

Banners

≡

Drag and drop patterns into the canvas.

**Full Site Editing Experience****Archive type: Name**

Term Description



Hello world!

Prepare patterns and template parts

Reusable block patterns are key to helping content teams work quickly and consistently. Start by identifying common layout needs, such as testimonials, contact forms, CTAs, or feature blocks, and develop a pattern library that meets those requirements.

At the same time, create template parts like headers, footers, and sidebars that can be shared across templates. These should reflect your global structure and help ensure consistency across pages and sites. Patterns and parts should be tested both technically and editorially.

Test editorial workflows and permissions

Before going live, involve content editors and stakeholders in testing the editorial experience. Are they able to create pages quickly and intuitively? Can they find and use the right patterns? Do role-based permissions work as expected?

Use this time to gather feedback, iterate on training materials, and fine-tune what can and can't be changed by different roles. This testing phase helps ensure high adoption rates and avoids unnecessary support overhead post-launch.

Roll out in phases

We recommend taking a phased approach to migration. Begin with a smaller site, campaign, or department to validate your implementation before scaling further. This pilot group should include a mix of editorial, design, and development users.

Use the learnings from this first rollout to refine governance rules, theme structure, and content migration processes. Once proven, expand your rollout in stages, ideally by content type, site, or region, to maintain control and minimise disruption.

Design



Full Site Editing Experience

Sample Page The Benefits of Mindfulness

Full Site Editing Experience

Sample Page The Benefits of Mindfulness

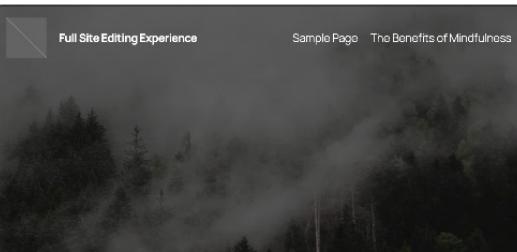


Full Site Editing Experience

Sample Page The Benefits of Mindfulness

Full Site Editing Experience

Sample Page The Benefits of Mindfulness



Full Site Editing Experience

Sample Page The Benefits of Mindfulness

Front Page ·

Full Site Editing Experience

Header with large title

No results found.

Monitor and iterate

Successfully launching your first FSE-powered site doesn't mean the work is done. Monitor adoption, performance, and support requests. Evaluate how your patterns are being used and whether any require adjustment. Gather analytics and editor feedback to refine your templates, patterns, and workflows.

Full Site Editing provides an evolving toolkit, and your implementation should evolve alongside it. Commit to a continuous improvement cycle to ensure your platform remains efficient, effective, and enjoyable to use.

17

Breaking up with shortcodes: migrating to blocks with confidence

One of the most common questions during Full Site Editing migrations is: what happens to all our shortcodes? For many organisations, shortcodes have been used for years to insert content, components, or layouts: everything from contact forms to buttons to custom CTAs.

While shortcodes have served their purpose, they are a legacy pattern that doesn't align with the structured, visual approach of FSE. Migrating to blocks not only modernises your site, but also improves maintainability, accessibility, and the editorial experience.

The good news? It's absolutely possible to move from shortcodes to blocks without disrupting your live content or overwhelming your teams.

Version	Jazz Musician	Release Date
5.2	Jaco Pastorius	May 7, 2019
5.1	Betty Carter	February 21, 2019
5.0	Bebo Valdés	December 6, 2018

Table
Create structured content in rows and columns to display information.

One of the hardest things to do in technology is disrupt yourself.

Matt Mullenweg

Pullquote

Give special visual emphasis to a quote from your text.

Why blocks are a better long-term approach

Blocks are the future of WordPress. Unlike shortcodes, which are essentially placeholders for PHP functions, blocks provide a visual, structured, and reusable way to insert content. Blocks are easier for non-technical editors to work with, provide better accessibility out of the box, and are fully compatible with theme .json, patterns, and the Site Editor.

Replacing shortcodes with blocks improves consistency across your content and gives your teams more intuitive control over how content appears.

Search



Group



Paragraph



Image



Heading



Gallery



List

Browse all

What a shortcode-to-block migration actually looks like

Migrating shortcodes doesn't mean rewriting every page manually. With the right planning, most of the heavy lifting can be automated or handled in batches. Here's how it usually works:

1 Audit your shortcodes

Start by identifying which shortcodes are in use, how often they appear, and where they are located (e.g. pages, posts, widgets, reusable blocks). Prioritise the most widely used or business-critical ones.

2 Create equivalent blocks

Developers can create custom blocks to replicate the functionality and appearance of your shortcodes. In many cases, there are already block-based replacements for popular shortcode use cases (such as buttons, galleries, or embeds).

3 Use scripted replacements where possible

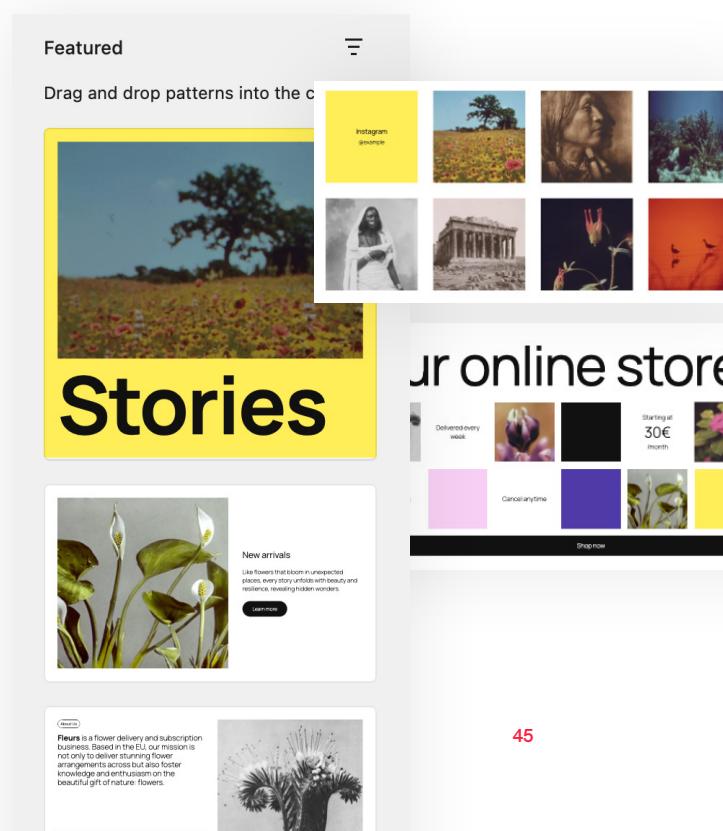
Shortcodes can often be replaced in bulk using database scripts or WP-CLI commands. This ensures consistency and reduces manual work, especially when migrating content across dozens or hundreds of pages.

4 Fallbacks for legacy content

In some cases, shortcodes can be preserved and rendered inside a Classic block during the transition period. This ensures nothing breaks while you phase them out gradually.

5 Empower your content teams

Once your blocks are in place, provide training and documentation for editors so they understand how to use the new system. Block-based content is easier to manage visually and allows for more flexible, reusable layouts.

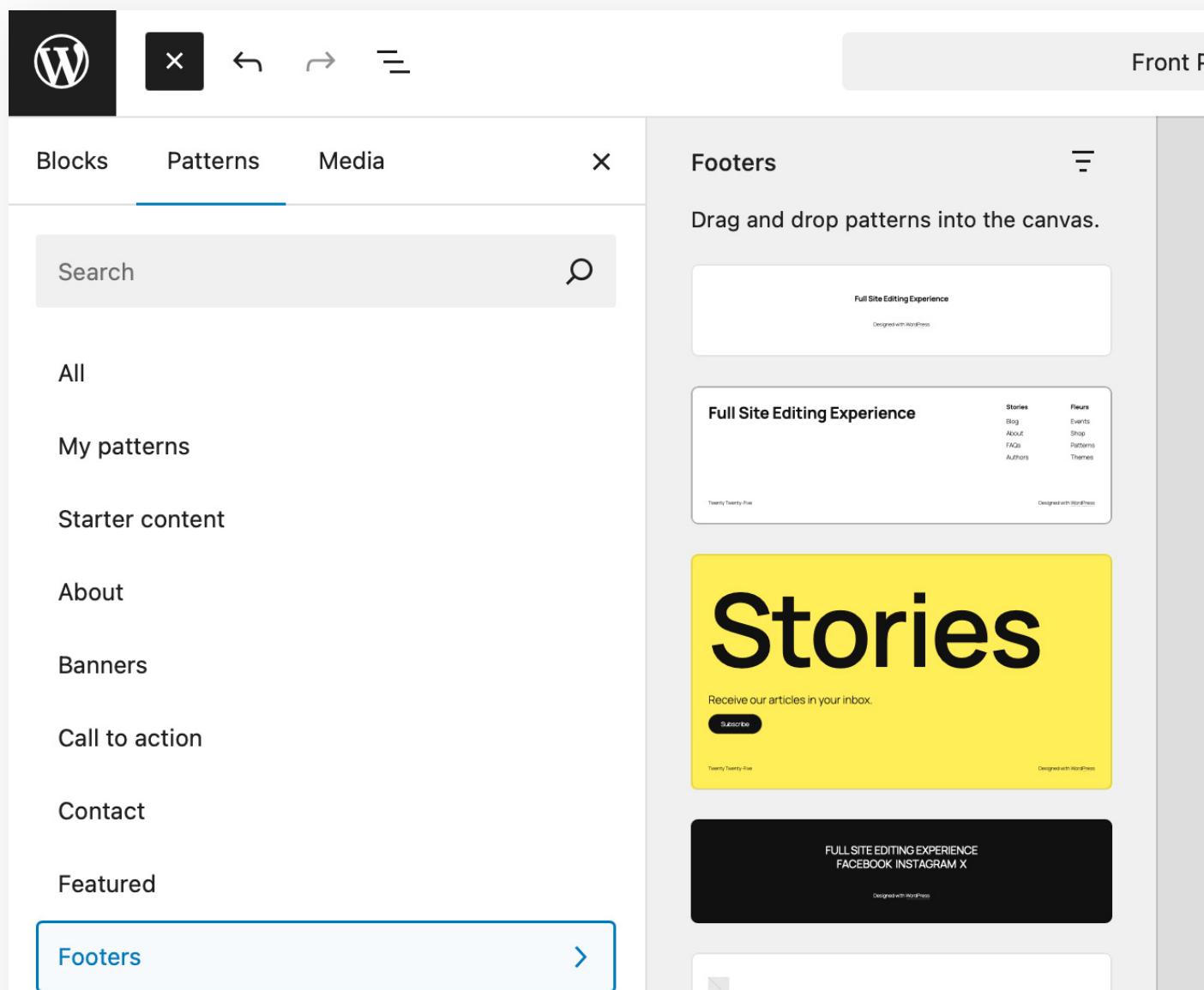


From technical debt to modern foundations

Shortcodes often represent hidden technical debt, but they don't have to hold you back. With a clear audit and migration plan, they can be converted into cleaner, more structured components that fit seamlessly into your FSE workflow.

We've helped enterprise teams move from legacy shortcode-heavy builds to block-first architectures that are faster, easier to manage, and more aligned with the future of WordPress.

Shortcodes may be legacy code, but they're not a blocker. Migrating to blocks is a clear, achievable path forward, and a great opportunity to modernise.



The screenshot shows the WordPress Block Editor interface. At the top, there's a toolbar with a logo, a close button, and navigation icons. Below the toolbar, a navigation bar has 'Blocks' selected, followed by 'Patterns' and 'Media' tabs, and a search bar. The main content area is titled 'All' and lists categories: 'My patterns', 'Starter content', 'About', 'Banners', 'Call to action', 'Contact', and 'Featured'. On the right, a sidebar titled 'Footers' shows a preview of a pattern titled 'Full Site Editing Experience' with a yellow 'Stories' section and a 'Subscribe' button. Other preview cards for 'Stories' and 'Full Site Editing Experience' are also visible.

WordPress Full Site Editing is more than just a visual builder. It is a new architecture for how enterprise websites are structured, designed, and governed. For organisations looking to streamline operations, improve brand consistency, and empower content teams, FSE offers a powerful and future-ready solution.

At Human Made, we help enterprise teams implement Full Site Editing with confidence. Whether you're starting from scratch or planning a theme migration, we can guide you through every step.

Get in touch to find out how Full Site Editing can support your organisation's digital transformation.

Get in touch

By moving to block themes, embracing structured design systems, and adopting modular content patterns, enterprises can modernise their digital presence without sacrificing control or performance.

