ROOTED DEV GUIDES

WordPress Theme Build Checklist

A practical build-and-launch checklist for creating a WordPress theme that is structured, responsive, accessible, fast, and maintainable.

FREE COMPANION RESOURCE

Companion to Book 4 - Building WordPress Themes From the Root Up

+ Plan the theme before designing individual templates.
+ Review responsive, accessibility, performance, and WordPress integration basics.
+ Use the final launch gate before releasing or handing off a theme.

How to use this: Work through the pages, mark what applies to you, and keep the PDF as a practical reference. This resource is intentionally concise; the full book goes deeper into the why, how, and real-world implementation.

Learn WordPress from the root up.

1. Theme Foundation

Start with the content and template system, not isolated screenshots.

Project definition

[ ] The site goals and primary visitor actions are documented.

[ ] Required content types, navigation areas, archives, search, and special templates are known.

[ ] The project has a defined minimum WordPress and PHP version.

[ ] The theme name, slug, text domain, and namespace/prefix choices are consistent.

[ ] Reusable design tokens are defined for spacing, typography, colors, borders, and breakpoints.

Theme structure

[ ] style.css contains valid theme metadata.

[ ] functions.php/bootstrap code is kept organized and focused.

[ ] Templates follow the WordPress template hierarchy intentionally.

[ ] Reusable template parts are used instead of copying large markup blocks.

[ ] Theme assets are organized predictably and are not loaded globally without reason.

2. WordPress Integration

A theme should cooperate with WordPress rather than hard-code what core already knows how to manage.

Core behavior

[ ] Theme supports appropriate WordPress features such as title-tag, thumbnails, HTML5, and editor styles.

[ ] Navigation menus or block theme navigation are configured intentionally.

[ ] Widgets/sidebars are used only where the design genuinely needs them.

[ ] Dynamic URLs use WordPress functions rather than hard-coded domains or paths.

[ ] User-facing text is translation-ready.

Content safety

[ ] Content that belongs to the site, not the design, is not trapped inside the theme.

[ ] Plugin territory features are kept out of the theme where practical.

[ ] Theme settings have safe defaults.

[ ] Escaping is applied to dynamic output according to context.

[ ] Template code handles missing images, empty content, and unusual titles gracefully.

3. Responsive & Accessible UI

A theme is not finished when the desktop screenshot looks right.

Responsive checks

[ ] Header/navigation works at narrow phone widths.

[ ] Typography remains readable without horizontal scrolling.

[ ] Images, embeds, tables, and code blocks do not overflow their containers.

[ ] Buttons and interactive controls have comfortable tap targets.

[ ] Layouts adapt cleanly at intermediate widths, not only at one mobile breakpoint.

Accessibility checks

[ ] The page has a logical heading hierarchy.

[ ] Keyboard users can reach and operate navigation and interactive controls.

[ ] Focus states are visible.

[ ] Color contrast is sufficient for text and important UI.

[ ] Form fields have associated labels and useful error states.

[ ] Images that convey meaning have appropriate alternative text supplied through WordPress content.

[ ] Landmarks and semantic HTML are used appropriately.

4. Performance & Assets

Themes should not make every page pay for assets it does not use.

Front-end performance

[ ] CSS and JavaScript are enqueued through WordPress.

[ ] Large libraries are not included when a small native solution is enough.

[ ] Fonts are limited to the weights/styles actually used.

[ ] Images are sized appropriately and responsive image behavior is preserved.

[ ] Third-party scripts are minimized and documented.

Quality checks

[ ] No PHP warnings or notices appear with WP_DEBUG enabled.

[ ] Browser console is free of unexplained errors.

[ ] HTML is structurally sound on representative templates.

[ ] The theme works with the WordPress admin bar visible.

[ ] The design remains usable with unusually long titles, menus, and content.

5. Theme Launch Gate

Run this list before handing the theme to a client, publishing it, or calling the build complete.

Template coverage

[ ] Homepage/front page is tested.

[ ] Single post and page templates are tested.

[ ] Category/tag/archive and author templates are tested where relevant.

[ ] Search results are tested.

[ ] 404 page is tested.

[ ] Comments are tested if supported.

Real-world states

[ ] Logged-in and logged-out views are tested.

[ ] Empty states and no-results states are readable.

[ ] Menu with many items does not break the header.

[ ] Featured image present and absent are both handled.

[ ] Very short and very long content both look acceptable.

Packaging & handoff

[ ] Development-only files and secrets are excluded from the release package.

[ ] Theme version is updated.

[ ] Changelog or release notes are current.

[ ] Setup instructions explain required plugins and configuration.

[ ] A clean installation test has been completed.

Release rule
Ship - No critical unchecked items, no PHP/JS errors, and the theme has been tested on real content at multiple screen sizes.
Hold - If navigation, forms, accessibility, template coverage, or upgrade/install behavior is still uncertain, keep testing.

GO DEEPER WITH THE FULL GUIDE

Book 4 - Building WordPress Themes From the Root Up

The full guide goes beyond this checklist into theme architecture, the template hierarchy, modern WordPress integration, responsive design, accessibility, performance, debugging, and a maintainable build workflow.

rooteddevguides.com