

Tasks: Paediatric Nephrology Clinic Website

Feature: 001-pediatric-nephrology-website

Input: Design documents from [/specs/001-pediatric-nephrology-website/](#)

Prerequisites: plan.md , spec.md , research.md , data-model.md , contracts/ , quickstart.md

Tests: Tests are NOT explicitly requested in the specification. Focus on functional implementation with manual testing via Theme Check, accessibility tools, and browser testing.

Organization: Tasks are grouped by user story to enable independent implementation and testing of each story following WordPress theme development best practices.

Format: [ID] [P?] [Story] Description

- **[P]:** Can run in parallel (different files, no dependencies)
- **[Story]:** Which user story this task belongs to (e.g., US1, US2, US7)
- Include exact file paths in descriptions

Path Conventions

- **Theme root:** [renalinfo/](#) (WordPress theme directory)
 - **Assets:** [renalinfo/assets/css/](#), [renalinfo/assets/js/](#)
 - **Templates:** [renalinfo/template-parts/](#)
 - **Includes:** [renalinfo/inc/](#)
 - **Languages:** [renalinfo/languages/](#)
-

Phase 1: Setup (Shared Infrastructure)

Purpose: Project initialization and WordPress theme structure

- T001 Create WordPress theme directory structure per plan.md at [renalinfo/](#)
- T002 Create [renalinfo/style.css](#) with complete theme header (Theme Name, Description, Version, Author, Text Domain: renalinfo, etc.)
- T003 Create [renalinfo/functions.php](#) with theme setup (add_theme_support for html5, title-tag, post-thumbnails, custom-logo, etc.)
- T004 [P] Create [renalinfo/index.php](#) as main template fallback
- T005 [P] Create [renalinfo/header.php](#) with basic HTML5 structure and wp_head()
- T006 [P] Create [renalinfo/footer.php](#) with wp_footer() and closing tags
- T007 [P] Create [renalinfo/screenshot.png](#) (1200x900px theme screenshot)
- T008 [P] Create [renalinfo/readme.txt](#) with theme information and GPL license
- T009 [P] Create [renalinfo/LICENSE](#) file (GPL v2 or later)
- T010 Create [renalinfo/package.json](#) with build tool dependencies (postcss, esbuild, eslint, stylelint per research.md)

- T011 Create `renalinfo/phpcs.xml` for PHP_CodeSniffer with WordPress coding standards configuration
 - T012 [P] Create `renalinfo/.gitignore` (node_modules/, compiled assets, debug.log, etc.)
 - T013 [P] Create `renalinfo/assets/css/src/style.css` (source CSS file)
 - T014 [P] Create `renalinfo/assets/js/src/main.js` (source JavaScript file)
 - T015 Run `npm install` in `renalinfo/` directory to install build dependencies
 - T016 Configure PostCSS with postcss-preset-env, autoprefixer, and cssnano in `renalinfo/postcss.config.js`
 - T017 Add npm scripts to `renalinfo/package.json` (build, watch, lint commands per research.md)
 - T018 Create `renalinfo/.editorconfig` for consistent code formatting
-

Phase 2: Foundational (Blocking Prerequisites)

Purpose: Core theme infrastructure that MUST be complete before ANY user story implementation

⚠ CRITICAL: No user story work can begin until this phase is complete

- T019 Register navigation menus (primary-menu, footer-menu) in `renalinfo/functions.php`
 - T020 Enqueue Google Fonts (Inter, Noto Sans Sinhala, Noto Sans Tamil) with font-display: swap in `renalinfo/functions.php`
 - T021 Enqueue theme CSS with `wp_enqueue_style` in `renalinfo/functions.php`
 - T022 Enqueue theme JavaScript with `wp_enqueue_script` in `renalinfo/functions.php`
 - T023 Implement asset versioning strategy (use `SCRIPT_DEBUG` for development vs production) in `renalinfo/functions.php`
 - T024 Set up AJAX localization with `admin-ajax.php` URL and nonces in `renalinfo/functions.php`
 - T025 Register custom image sizes (article-hero: 1200x600, staff-profile: 400x400, article-thumb: 300x200) in `renalinfo/functions.php`
 - T026 Register 'article' custom post type with proper supports and taxonomies in `renalinfo/inc/custom-post-types.php`
 - T027 [P] Register 'journey' custom post type in `renalinfo/inc/custom-post-types.php`
 - T028 [P] Register 'staff' custom post type in `renalinfo/inc/custom-post-types.php`
 - T029 [P] Register 'medical_term' custom post type in `renalinfo/inc/custom-post-types.php`
 - T029a [P] Create admin columns and quick-edit UI for medical term abbreviations and synonyms in `renalinfo/inc/custom-post-types.php` to support FR-016a glossary management
 - T030 [P] Register 'support_resource' custom post type in `renalinfo/inc/custom-post-types.php`
 - T031 Register 'article_category' taxonomy (hierarchical) in `renalinfo/inc/custom-taxonomies.php`
 - T032 [P] Register 'audience_type' taxonomy in `renalinfo/inc/custom-taxonomies.php`
 - T033 [P] Register 'specialization' taxonomy in `renalinfo/inc/custom-taxonomies.php`
 - T034 [P] Register 'resource_type' taxonomy in `renalinfo/inc/custom-taxonomies.php`
 - T035 Create custom fields framework using CMB2 or ACF (or native) in `renalinfo/inc/custom-fields.php`
-

- T036 Add article custom fields (_article_template, _reading_level, _audience, _medical_review_date, etc.) in [renalinfo/inc/custom-fields.php](#)
- T037 [P] Add journey custom fields (_journey_articles, _journey_audience, _estimated_time) in [renalinfo/inc/custom-fields.php](#)
- T038 [P] Add staff custom fields (_staff_role, _staff_credentials, _staff_personal_bio, etc.) in [renalinfo/inc/custom-fields.php](#)
- T039 [P] Add medical_term custom fields (_term_abbreviation, _term_synonyms, definitions) in [renalinfo/inc/custom-fields.php](#)
- T040 Create template functions file [renalinfo/inc/template-functions.php](#) with helper functions
- T041 Create template tags file [renalinfo/inc/template-tags.php](#) with display functions
- T042 Implement sanitization functions for all custom field inputs in [renalinfo/inc/template-functions.php](#)
- T043 Implement validation functions for article requirements per data-model.md in [renalinfo/inc/template-functions.php](#)
- T044 Set up WordPress revisions configuration (WP_POST_REVISIONS in wp-config.php documentation)
- T045 Create version history display function in [renalinfo/inc/template-tags.php](#)
- T046 Implement base CSS structure (variables, typography, layout grid) in [renalinfo/assets/css/src/style.css](#)
- T047 Add CSS for font families (Latin, Sinhala, Tamil) with system font fallbacks in [renalinfo/assets/css/src/style.css](#)
- T048 Implement color palette CSS variables (calming blues, soft greens, muted teals) in [renalinfo/assets/css/src/style.css](#)
- T049 Create responsive breakpoint mixins/utilities in [renalinfo/assets/css/src/style.css](#)
- T050 Build initial CSS and JavaScript assets with [npm run build](#)

Checkpoint: Foundation ready - user story implementation can now begin in parallel

Phase 3: User Story 1 - Parent Finding Treatment Information (Priority: P1) MVP

Goal: Parents can find and read condition explainer articles within 3 clicks, with grade 8-9 reading level content, medical term definitions, and related treatment links

Independent Test:

1. Create sample "Chronic Kidney Disease" article with condition-explainer template
2. Navigate from homepage → Kidney Conditions category → article (within 3 clicks)
3. Verify article displays FAQs, simple definitions, related treatments
4. Run Flesch-Kincaid readability check (target 8-9 grade level)
5. Verify WCAG 2.1 AA compliance with axe DevTools

Implementation for User Story 1

- T051 [P] [US1] Create homepage template [renalinfo/front-page.php](#) with category entry points and featured content

- T052 [P] [US1] Create archive template [renalinfo/archive.php](#) for article listings with audience filtering
- T053 [US1] Create single article template [renalinfo/single-article.php](#) with template-specific layouts
- T054 [US1] Create condition-explainer template part [renalinfo/template-parts/content/content-condition-explainer.php](#)
- T055 [US1] Implement FAQ display section in condition-explainer template with schema.org markup
- T056 [US1] Add medical term tooltip/glossary integration in [renalinfo/inc/template-tags.php](#)
- T057 [US1] Create related articles widget/section in [renalinfo/template-parts/content/related-articles.php](#)
- T058 [US1] Style condition-explainer template (typography, spacing, colors) in [renalinfo/assets/css/src/components/_condition-explainer.css](#)
- T059 [US1] Add support resource links section to article template in [renalinfo/template-parts/content/support-resources.php](#)
- T060 [US1] Implement breadcrumb navigation for article pages in [renalinfo/inc/template-tags.php](#)
- T061 [US1] Add reading time display function in [renalinfo/inc/template-tags.php](#)
- T062 [US1] Create homepage CSS (hero section, category cards, featured content grid) in [renalinfo/assets/css/src/pages/_home.css](#)
- T063 [US1] Create archive page CSS (article listing, filters, pagination) in [renalinfo/assets/css/src/pages/_archive.css](#)
- T064 [US1] Implement responsive images with WebP support and lazy loading in [renalinfo/inc/template-functions.php](#)
- T065 [US1] Add image optimization helper function (picture element with srcset) in [renalinfo/inc/template-functions.php](#)
- T066 [US1] Create category taxonomy template [renalinfo/taxonomy-article_category.php](#)
- T067 [US1] Style taxonomy archive pages in [renalinfo/assets/css/src/pages/_taxonomy.css](#)

Checkpoint: Parents can navigate from homepage to condition articles, read content with appropriate formatting, see FAQs and related treatments

Phase 4: User Story 7 - Multilingual User Accessing Native Language Content (Priority: P2)

Goal: Sinhala and Tamil speaking parents can access medical information in their native language with proper Unicode rendering, language switcher, and fallback to English when translation unavailable

Independent Test:

1. Install and configure Polylang plugin (3 languages: en, si, ta)
2. Create English article and Sinhala/Tamil translations
3. Use language switcher to change languages
4. Verify fonts render correctly (Noto Sans Sinhala/Tamil)

5. Test fallback behavior when translation missing
6. Verify language persists within browser session only

Implementation for User Story 7

- T068 [US7] Document Polylang installation and configuration steps in [renalinfo/readme.txt](#)
- T069 [US7] Register theme strings for translation with `pll_register_string()` in [renalinfo/functions.php](#)
- T070 [US7] Create language switcher template part with prominent first-visit modal/banner prompt UI for language selection (FR-045) in [renalinfo/template-parts/navigation/language-switcher.php](#)
- T071 [US7] Style language switcher (flags, dropdown, mobile-friendly) and first-visit prompt modal in [renalinfo/assets/css/src/components/_language-switcher.css](#)
- T072 [US7] Add language switcher to header and implement first-visit prompt JavaScript logic in [renalinfo/header.php](#)
- T073 [US7] Implement language-specific font loading logic in [renalinfo/functions.php](#)
- T074 [US7] Add `[lang="si"]` and `[lang="ta"]` CSS rules for Sinhala/Tamil fonts in [renalinfo/assets/css/src/style.css](#)
- T075 [US7] Create fallback notification template part [renalinfo/template-parts/content/translation-notice.php](#)
- T076 [US7] Implement translation availability check function in [renalinfo/inc/template-functions.php](#)
- T077 [US7] Add translation notice display to article template when fallback to English
- T078 [US7] Implement AJAX language selection endpoint (`renalinfo_set_language`) in [renalinfo/inc/ajax-handlers.php](#)
- T079 [US7] Create JavaScript for sessionStorage language persistence in [renalinfo/assets/js/src/language-handler.js](#)
- T080 [US7] Add query parameter language handling in [renalinfo/inc/template-functions.php](#)
- T081 [US7] Modify search to prioritize current language results in [renalinfo/inc/template-functions.php](#)
- T082 [US7] Add language indicator badges to search results in [renalinfo/search.php](#)
- T083 [US7] Create POT file generation documentation in [renalinfo/languages/readme.txt](#)
- T084 [US7] Generate initial POT file with WP-CLI or plugin in [renalinfo/languages/renalinfo.pot](#)
- T085 [US7] Style multilingual content (proper line-height, letter-spacing for Sinhala/Tamil) in [renalinfo/assets/css/src/style.css](#)

Checkpoint: Users can switch between English, Sinhala, and Tamil with proper Unicode rendering, see fallback notices, and have language persist within session

Phase 5: User Story 2 - Referring Physician Accessing Clinical Guidelines (Priority: P2)

Goal: Physicians can quickly find professional-level content with clinical details, citations, referral protocols, and distinguish it from family-focused content

Independent Test:

1. Create sample professional article with citations and clinical details
2. Navigate to professional content section
3. Filter search by "For Professionals" audience
4. Verify professional article template displays citations, author credentials
5. Access referral process information within 2 minutes

Implementation for User Story 2

- T086 [P] [US2] Create professional article template part [renalinfo/template-parts/content/content-professional-article.php](#)
- T087 [US2] Add citations display section with proper academic formatting in professional template
- T088 [US2] Add author credentials display (linked to staff profiles) in professional template
- T089 [US2] Create audience filter widget/component in [renalinfo/inc/widgets.php](#)
- T090 [US2] Add audience filtering to search results in [renalinfo/search.php](#)
- T091 [US2] Style professional article template (formal typography, citation formatting) in [renalinfo/assets/css/src/components/_professional-article.css](#)
- T092 [US2] Create "For Professionals" visual badge/label component in [renalinfo/template-parts/content/audience-badge.php](#)
- T093 [US2] Add audience badge to article cards in archive views
- T094 [US2] Create referral process page template [renalinfo/page-templates/referral-process.php](#)
- T095 [US2] Add referral information to professional article sidebar/footer
- T096 [US2] Style referral process page and sidebar widgets in [renalinfo/assets/css/src/pages/_referral.css](#)
- T097 [US2] Implement professional content navigation menu section
- T098 [US2] Add "For Professionals" filter to homepage category sections

Checkpoint: Physicians can find and distinguish professional content, access referral guidelines, and see properly formatted citations

Phase 6: User Story 3 - Patient Journey Navigation (Priority: P2)

Goal: Families can follow sequential journey modules (e.g., kidney transplant journey) with progress indicators, next/previous navigation, and overview page

Independent Test:

1. Create journey module with 5+ related articles
2. Navigate through journey with next/previous links
3. Verify "Article X of Y" progress display
4. Access journey overview page with all articles listed
5. Test journey navigation works without persistent tracking

Implementation for User Story 3

- T099 [P] [US3] Create journey archive template [renalinfo/archive-journey.php](#)
- T100 [P] [US3] Create single journey template [renalinfo/single-journey.php](#) with article overview
- T101 [US3] Create journey navigation component [renalinfo/template-parts/navigation/journey-navigation.php](#)
- T102 [US3] Implement journey progress indicator (Article X of Y) in [renalinfo/inc/template-tags.php](#)
- T103 [US3] Add next/previous article navigation within journey in journey template
- T104 [US3] Create journey overview list display with article descriptions
- T105 [US3] Add "Part of Journey" badge to articles in journey in [renalinfo/template-parts/content/journey-badge.php](#)
- T106 [US3] Style journey navigation (progress bar, next/prev buttons) in [renalinfo/assets/css/src/components/_journey-nav.css](#)
- T107 [US3] Style journey overview page (article list, estimated time display) in [renalinfo/assets/css/src/pages/_journey.css](#)
- T108 [US3] Add journey module cards to homepage featured section
- T109 [US3] Create journey category/filter for easy discovery
- T110 [US3] Implement journey article order validation in [renalinfo/inc/template-functions.php](#)

Checkpoint: Families can discover journey modules, navigate through sequential articles with clear progress, and access complete overview

Phase 7: User Story 6 - Staff Profile Discovery (Priority: P3)

Goal: Families can view staff profiles with photos, approachable bios, professional credentials, specializations, and links to authored articles

Independent Test:

1. Create 3-5 staff profiles with all required fields
2. Navigate to team/staff archive page
3. View individual staff profile with photo, bio, credentials
4. Click specialization to see related articles
5. Access authored articles from profile

Implementation for User Story 6

- T111 [P] [US6] Create staff archive template [renalinfo/archive-staff.php](#) with grid/card layout
- T112 [P] [US6] Create single staff profile template [renalinfo/single-staff.php](#)
- T113 [US6] Create staff card component [renalinfo/template-parts/content/staff-card.php](#)
- T114 [US6] Add personal bio section with friendly tone styling
- T115 [US6] Add professional bio section with credentials display
- T116 [US6] Create specialization tags/badges display
- T117 [US6] Add authored articles section (query by author) to staff profile

- T118 [US6] Implement contact information display (email, phone, hours)
- T119 [US6] Add languages spoken display with flags/icons
- T120 [US6] Style staff archive (grid layout, responsive cards) in [renalinfo/assets/css/src/pages/_staff-archive.css](#)
- T121 [US6] Style staff profile page (photo, bio sections, credentials) in [renalinfo/assets/css/src/pages/_staff-profile.css](#)
- T122 [US6] Add staff profile images optimization (400x400 size, WebP)
- T123 [US6] Create "Meet Our Team" section for homepage
- T124 [US6] Add author byline with profile link to articles

Checkpoint: Families can discover staff members, view detailed profiles with photos and bios, see specializations, and find their authored articles

Phase 8: User Story 4 - Mobile Parent Seeking Quick Information (Priority: P3)

Goal: Parents on mobile devices can access all features with responsive design, appropriate touch targets (44x44px), tap-to-call, and mobile-optimized navigation

Independent Test:

1. Test site on mobile device or Chrome DevTools device emulation (320px width)
2. Verify navigation menu is touch-friendly
3. Test tap-to-call on phone number in header
4. Verify content is readable without horizontal scrolling
5. Check touch target sizes (minimum 44x44px)
6. Test search and autocomplete on mobile

Implementation for User Story 4

- T125 [P] [US4] Implement responsive navigation menu (hamburger menu) in [renalinfo/template-parts/navigation/primary-menu.php](#)
- T126 [P] [US4] Add tap-to-call functionality for phone number in header
- T127 [US4] Style mobile navigation (hamburger icon, slide-out menu, overlay) in [renalinfo/assets/css/src/components/_mobile-nav.css](#)
- T128 [US4] Implement mobile navigation JavaScript (toggle, accessibility) in [renalinfo/assets/js/src/mobile-nav.js](#)
- T129 [US4] Add responsive images with mobile-optimized sizes
- T130 [US4] Optimize touch target sizes (buttons, links minimum 44x44px) in [renalinfo/assets/css/src/style.css](#)
- T131 [US4] Create mobile-specific header layout with compact logo and CTA
- T132 [US4] Style mobile typography (appropriate font sizes, line heights) in [renalinfo/assets/css/src/style.css](#)
- T133 [US4] Optimize mobile search interface (full-width, large touch targets)
- T134 [US4] Test and fix mobile layout issues (320px-768px breakpoints)
- T135 [US4] Add mobile-friendly footer (collapsible sections if needed)
- T136 [US4] Optimize mobile forms (large inputs, clear labels)

- T137 [US4] Test mobile language switcher (touch-friendly dropdown)

Checkpoint: All features work seamlessly on mobile devices with appropriate touch targets, readable text, and optimized navigation

Phase 9: User Story 5 - Accessibility for Users with Disabilities (Priority: P3)

Goal: Users with disabilities can access all content using assistive technologies with WCAG 2.1 AA compliance, keyboard navigation, screen reader support, and high-contrast mode

Independent Test:

1. Run axe DevTools accessibility scan (target: zero critical issues)
2. Navigate entire site using keyboard only (tab, enter, arrows)
3. Test with screen reader (NVDA or JAWS)
4. Verify focus indicators are visible on all interactive elements
5. Check color contrast ratios (4.5:1 normal, 3:1 large text)
6. Test font scaling to 200% without layout breaking

Implementation for User Story 5

- T138 [P] [US5] Add skip-to-content link at top of header in [renalinfo/header.php](#)
- T139 [P] [US5] Implement proper heading hierarchy (h1-h6) validation across templates
- T140 [US5] Add ARIA labels to all navigation elements in templates
- T141 [US5] Add alt text requirement and validation for featured images in [renalinfo/inc/custom-fields.php](#)
- T142 [US5] Implement keyboard navigation for custom components (language switcher, mobile menu)
- T143 [US5] Add focus indicators (visible outline) for all interactive elements in [renalinfo/assets/css/src/style.css](#)
- T144 [US5] Create high-contrast mode toggle in [renalinfo/inc/customizer.php](#)
- T145 [US5] Add high-contrast CSS variables and styles in [renalinfo/assets/css/src/accessibility/_high-contrast.css](#)
- T146 [US5] Verify all color contrast ratios meet WCAG 2.1 AA standards (4.5:1 normal, 3:1 large) in both normal and high-contrast modes (FR-024)
- T147 [US5] Add proper form labels and fieldset/legend for all forms
- T148 [US5] Implement ARIA live regions for dynamic content updates (search results, language switch)
- T149 [US5] Add schema.org markup for articles, FAQs, and staff profiles
- T150 [US5] Test font scaling to 200% and fix overflow issues
- T151 [US5] Add screen reader text for icon-only buttons/links
- T152 [US5] Create accessibility statement page template [renalinfo/page-templates/accessibility-statement.php](#)
- T153 [US5] Document accessibility features and testing in [renalinfo/readme.txt](#)
- T154 [US5] Set up monthly accessibility audit process: create calendar reminders, document audit checklist with axe DevTools/WAVE tool usage, and define remediation workflow (FR-027) in

Checkpoint: Site passes WCAG 2.1 AA compliance with axe DevTools, works with keyboard only, compatible with screen readers, and supports high-contrast mode

Phase 10: Search Functionality (Cross-Cutting Feature)

Goal: Implement medical-specific search with autocomplete, medical term abbreviation matching, audience filters, and multilingual support

Independent Test:

1. Search for "CKD" and verify it matches "Chronic Kidney Disease"
2. Test autocomplete appears after 3 characters with debouncing
3. Filter results by "For Families" audience
4. Search in Sinhala and verify correct language results
5. Test rate limiting (10 requests per 10 seconds)

Implementation

- T155 [P] Create search page template [renalinfo/search.php](#) with filters
- T156 [P] Create search form component [renalinfo/template-parts/navigation/search-form.php](#)
- T157 Create enhanced search query modification function in [renalinfo/inc/template-functions.php](#)
- T158 Implement medical term abbreviation search (meta_query for glossary) in [renalinfo/inc/template-functions.php](#)
- T159 Add audience type filtering to search query
- T160 Add language-specific search filtering with Polylang
- T161 Implement AJAX search autocomplete endpoint (renalinfo_search_autocomplete) in [renalinfo/inc/ajax-handlers.php](#)
- T162 Add rate limiting for search autocomplete (transients) in AJAX handler
- T163 Create autocomplete JavaScript with debouncing (300ms) in [renalinfo/assets/js/src/search-autocomplete.js](#)
- T164 Style search form and results page in [renalinfo/assets/css/src/components/_search.css](#)
- T165 Style autocomplete dropdown (results, keyboard navigation) in [renalinfo/assets/css/src/components/_autocomplete.css](#)
- T166 Add search results language indicators and badges
- T167 Implement search results relevance scoring (title match > content match)
- T168 Add "no results" state with suggestions
- T169 Test search performance with 500+ articles

Checkpoint: Search works with medical abbreviations, autocomplete, audience filters, language prioritization, and performs well at scale

Phase 11: Header & Footer Components (Cross-Cutting)

Goal: Implement consistent header with logo, navigation, phone number, CTA button, language switcher, and footer with contact info, legal links, social media

Independent Test:

1. Verify header displays on all pages with correct elements
2. Test header phone number tap-to-call on mobile
3. Test "Request Appointment" CTA button links to correct page
4. Verify footer displays on all pages
5. Test footer social media links open in new tabs

Implementation

- T170 [P] Implement complete header structure in [renalinfo/header.php](#) (logo, nav, phone, CTA, language switcher)
- T171 [P] Implement complete footer structure in [renalinfo/footer.php](#) (contact, links, social, disclaimer)
- T172 Create customizer settings for header in [renalinfo/inc/customizer.php](#) (phone number, CTA text/link)
- T173 Create customizer settings for footer in [renalinfo/inc/customizer.php](#) (social media links, copyright text, medical disclaimer textarea)
- T174 Add custom logo support and display in header
- T175 Create primary navigation walker for accessibility in [renalinfo/inc/template-functions.php](#)
- T176 Style header (logo, navigation, CTA button, phone) in [renalinfo/assets/css/src/layout/_header.css](#)
- T177 Style footer (multi-column layout, links, social icons) in [renalinfo/assets/css/src/layout/_footer.css](#)
- T178 Add sticky header functionality (JavaScript) in [renalinfo/assets/js/src/sticky-header.js](#)
- T179 Implement global medical disclaimer display in footer with customizer textarea option (FR-031) and Polylang string translation support for multilingual disclaimer
- T180 Create footer menu walker for proper markup
- T181 Add footer widget areas registration in [renalinfo/functions.php](#)

Checkpoint: Header and footer display consistently across all pages with all required elements and customization options

Phase 12: Optional Feedback Collection (Cross-Cutting)

Goal: Allow anonymous users to provide feedback on article helpfulness with optional comments, rate-limited submissions, and simple analytics

Independent Test:

1. Navigate to article page

2. Click "Was this helpful?" - Yes or No
3. Optionally add comment (max 500 chars)
4. Verify feedback submitted successfully
5. Test rate limiting (5 submissions per hour)
6. Admin: View feedback counts on articles

Implementation

- T182 [P] Create feedback widget component [renalinfo/template-parts/content/feedback-widget.php](#)
- T183 [P] Add feedback widget to article templates
- T184 Implement AJAX feedback submission endpoint (`renalinfo_submit_feedback`) in [renalinfo/inc/ajax-handlers.php](#)
- T185 Add feedback counter meta fields (`_feedback_helpful_count`, `_feedback_not_helpful_count`) to articles
- T186 Implement rate limiting for feedback (5 per hour per IP)
- T187 Add session check to prevent duplicate feedback on same article
- T188 Create feedback JavaScript (submit, UI updates, validation) in [renalinfo/assets/js/src/feedback.js](#)
- T189 Style feedback widget (buttons, comment textarea, success message) in [renalinfo/assets/css/src/components/_feedback.css](#)
- T190 Add feedback comment storage and admin review interface
- T191 Add feedback counts display in admin article list columns
- T192 Sanitize and validate feedback comment input (max 500 chars)

Checkpoint: Users can submit helpful/not helpful feedback with optional comments, rate limiting works, admin can view feedback data

Phase 13: Content Version Control & Update Notices (Cross-Cutting)

Goal: Display "Updated [date]" notices on articles, maintain version history accessible to editors, track medical review dates, and handle content archival with redirects

Independent Test:

1. Edit existing article and add version note
2. Verify "Updated [date]" notice displays prominently on frontend
3. Admin: View revision history for article
4. Test 301 redirect from archived article to replacement
5. Verify version history shows across all language translations

Implementation

- T193 [P] Add update notice display to article templates (all types)
 - T194 [P] Create version history display function in [renalinfo/inc/template-tags.php](#)
 - T195 Add medical review date display with styling
 - T196 Create version note meta box for content editors in [renalinfo/inc/custom-fields.php](#)
-

- T197 Style update notice (prominent, date formatting) in [renalinfo/assets/css/src/components/_update-notice.css](#)
- T198 Implement content archival workflow documentation in [renalinfo/readme.txt](#)
- T199 Create redirect management system for archived content with custom meta field UI for setting redirect URLs in [renalinfo/inc/template-functions.php](#) to support FR-005a
- T200 Add version history link for logged-in editors with edit permissions
- T201 Ensure version history syncs across language translations

Checkpoint: Articles display update dates, editors can add version notes, revision history is accessible, archived content redirects properly

Phase 14: Performance Optimization (Cross-Cutting)

Goal: Achieve PageSpeed Insights score ≥ 90 , LCP $< 2.5s$, FID $< 100ms$, CLS < 0.1 , with lazy loading, asset minification, and caching strategies

Independent Test:

1. Run Google PageSpeed Insights on homepage and article pages
2. Test with Query Monitor plugin for database query optimization
3. Verify WebP images generate and serve correctly
4. Test lazy loading works on images
5. Verify page load $< 3s$ on 5 Mbps connection

Implementation

- T202 [P] Implement WebP image generation on upload in [renalinfo/inc/template-functions.php](#)
- T203 [P] Create responsive image function with picture element in [renalinfo/inc/template-functions.php](#)
- T204 Add lazy loading to all images (loading="lazy" attribute)
- T205 Implement critical CSS extraction and inline for above-fold content
- T206 Add DNS prefetch and preconnect for Google Fonts in header
- T207 Minify CSS and JavaScript with build process (npm run build)
- T208 Implement asset versioning based on file modification time
- T209 Add browser caching headers documentation for .htaccess
- T210 Optimize database queries (remove unnecessary meta_queries, use proper indexing)
- T211 Implement transient caching for expensive queries (taxonomy queries, search results)
- T212 Add conditional loading for scripts (only load search JS on relevant pages)
- T213 Optimize web font loading (only load necessary weights and subsets)
- T214 Test and optimize Largest Contentful Paint (LCP) metric
- T215 Test and optimize Cumulative Layout Shift (CLS) by adding width/height to images
- T216 Document CDN integration options in [renalinfo/readme.txt](#)

Checkpoint: Site meets all performance targets (PageSpeed ≥ 90 , LCP $< 2.5s$, FID $< 100ms$, CLS < 0.1) on both mobile and desktop

Phase 15: Treatment Procedure Template (Additional Content Type)

Goal: Implement treatment procedure template with step-by-step instructions, what-to-expect sections, pre/post-care information, and child-focused experience descriptions

Independent Test:

1. Create sample treatment procedure article (e.g., "Dialysis Procedure")
2. Verify step-by-step procedure display with numbering
3. Check what-to-expect section displays
4. Verify pre-care and post-care sections
5. Test child perspective narrative

Implementation

- T217 [P] Create treatment procedure template part [renalinfo/template-parts/content/content-treatment-procedure.php](#)
- T218 [P] Add procedure steps repeater field display with numbered list
- T219 Add what-to-expect section with child-focused language
- T220 Add pre-care and post-care sections with checklists
- T221 Style treatment procedure template (steps, icons, timelines) in [renalinfo/assets/css/src/components/_treatment-procedure.css](#)
- T222 Add procedure duration and recovery time display
- T223 Add related treatments and conditions links
- T224 Implement procedure illustration/diagram support

Checkpoint: Treatment procedure articles display with clear step-by-step instructions, child-friendly explanations, and care information

Phase 16: Support Resources Integration

Goal: Display support resources (support groups, educational materials, helplines) on relevant pages with categorization and external link management

Independent Test:

1. Create 5-10 support resources with different types
2. View support resources archive page
3. Verify resources appear on related article pages
4. Test external links open in new tabs with proper attributes
5. Check resource filtering by type works

Implementation

- T225 [P] Create support resources archive template [renalinfo/archive-support_resource.php](#)
- T226 [P] Create single support resource template [renalinfo/single-support_resource.php](#)

- T227 Create support resource card component [renalinfo/template-parts/content/resource-card.php](#)
- T228 Add resource type filtering to archive page
- T229 Implement related resources display on article pages
- T230 Add external link icon and rel="noopener noreferrer" to external links
- T231 Style support resources (cards, icons, contact info) in [renalinfo/assets/css/src/pages/_resources.css](#)
- T232 Add resource availability/hours display
- T233 Create resources homepage section with featured resources

Checkpoint: Support resources are discoverable, properly categorized, display on relevant pages, and external links work correctly

Phase 17: Polish & Cross-Cutting Concerns

Purpose: Final improvements, documentation, testing, and launch preparation

- T234 [P] Run PHP_CodeSniffer on all PHP files and fix violations: [./vendor/bin/phpcs](#)
- T235 [P] Run Theme Check plugin and fix all errors/warnings
- T236 [P] Run axe DevTools on all page templates and fix accessibility issues
- T237 [P] Test all templates in Chrome, Firefox, Safari, and Edge (current and previous versions)
- T238 [P] Create sample content for all custom post types (50+ articles, 5 staff, 20 terms)
- T239 [P] Create translations for sample content (English, Sinhala, Tamil)
- T240 Generate final POT file for translators in [renalinfo/languages/renalinfo.pot](#)
- T241 Create theme documentation in [renalinfo/readme.txt](#) (features, setup, customization)
- T242 Document Polylang configuration steps in readme
- T243 Create sample content import documentation
- T244 Test quickstart.md with fresh WordPress installation
- T245 Create theme screenshot with actual content (1200x900px)
- T246 Add inline code documentation (PHPDoc) to all functions
- T247 Create changelog in [renalinfo/readme.txt](#)
- T248 Test 404 error page styling and create custom template if needed
- T249 Test and style WordPress default widgets if used
- T250 Create print stylesheet for article pages in [renalinfo/assets/css/src/print.css](#)
- T251 Add RSS feed customization if needed
- T252 Test theme with popular page builder plugins (compatibility check)
- T253 Test theme with popular caching plugins
- T254 Security review: verify all inputs sanitized, outputs escaped, nonces used
- T255 Performance testing: verify all targets met (PageSpeed ≥90, LCP <2.5s, etc.)
- T256 Final code review and refactoring
- T257 Create deployment checklist documentation
- T258 Verify GPL license in all files
- T259 Test content import/export workflow
- T260 Final build of all assets with [npm run build](#)
- T260a Performance load test with 500+ sample articles to verify FR-005 requirement (no performance degradation at scale) and maintain PageSpeed targets

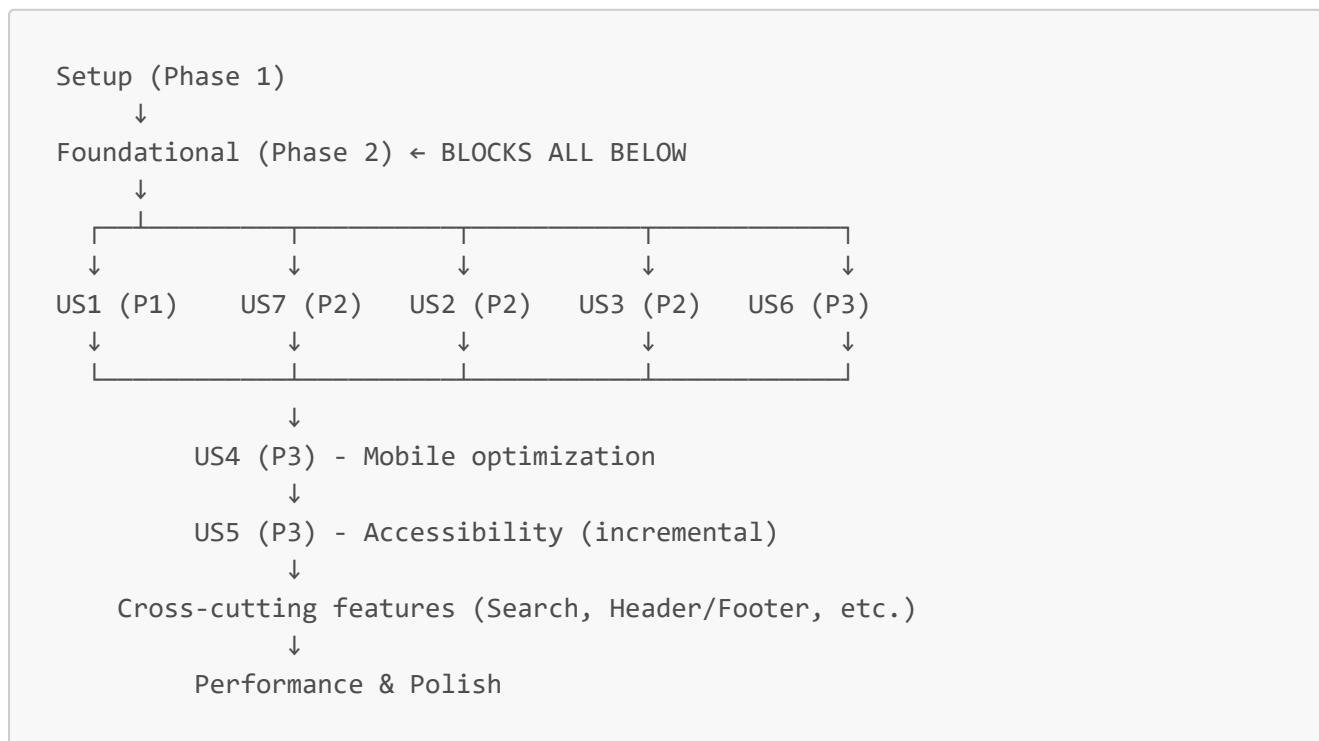
Checkpoint: Theme is production-ready, fully tested, documented, and meets all quality standards

Dependencies & Execution Order

Phase Dependencies

- **Setup (Phase 1):** No dependencies - can start immediately
- **Foundational (Phase 2):** Depends on Setup completion - BLOCKS all user stories
- **User Story 1 (Phase 3):** Depends on Foundational (Phase 2) - P1 priority, MVP target
- **User Story 7 (Phase 4):** Depends on Foundational (Phase 2) - P2 priority, multilingual critical
- **User Story 2 (Phase 5):** Depends on Foundational (Phase 2) and US1 (shared templates) - P2 priority
- **User Story 3 (Phase 6):** Depends on Foundational (Phase 2) and US1 (article templates) - P2 priority
- **User Story 6 (Phase 7):** Depends on Foundational (Phase 2) - P3 priority, independent
- **User Story 4 (Phase 8):** Depends on completion of US1, US2, US7 (requires base templates to optimize) - P3 priority
- **User Story 5 (Phase 9):** Spans all user stories (accessibility is incremental) - P3 priority
- **Search (Phase 10):** Depends on Foundational (Phase 2) - can start early, cross-cutting
- **Header/Footer (Phase 11):** Depends on Foundational (Phase 2) - blocks all user story UI
- **Feedback (Phase 12):** Depends on US1 (article templates) - optional, can be done late
- **Versioning (Phase 13):** Depends on US1 (article templates) - cross-cutting
- **Performance (Phase 14):** Depends on all primary features - optimization phase
- **Treatment Template (Phase 15):** Extends US1 - additional content type
- **Resources (Phase 16):** Depends on Foundational (Phase 2) - independent feature
- **Polish (Phase 17):** Depends on all desired features being complete

User Story Dependencies



Recommended Execution Order

Phase 1: MVP (Minimum Viable Product)

1. Setup (Phase 1) → T001-T018
2. Foundational (Phase 2) → T019-T050 ⚠ CRITICAL BLOCKING PHASE
3. Header/Footer (Phase 11) → T170-T181 (needed for navigation)
4. User Story 1 (Phase 3) → T051-T067 (core parent content access)
5. Stop and validate MVP with stakeholders

Phase 2: Multilingual + Professional Content

1. User Story 7 (Phase 4) → T068-T085 (multilingual support)
2. Search (Phase 10) → T155-T169 (findability)
3. User Story 2 (Phase 5) → T086-T098 (professional content)

Phase 3: Enhanced Experience

1. User Story 3 (Phase 6) → T099-T110 (journey modules)
2. User Story 6 (Phase 7) → T111-T124 (staff profiles)
3. Treatment Template (Phase 15) → T217-T224 (additional content type)
4. Resources (Phase 16) → T225-T233 (support resources)

Phase 4: Mobile & Accessibility

1. User Story 4 (Phase 8) → T125-T137 (mobile optimization)
2. User Story 5 (Phase 9) → T138-T154 (accessibility compliance)

Phase 5: Refinement

1. Feedback (Phase 12) → T182-T192 (optional feedback)
2. Versioning (Phase 13) → T193-T201 (content management)
3. Performance (Phase 14) → T202-T216 (optimization)
4. Polish (Phase 17) → T234-T260 (final QA and launch prep)

Parallel Opportunities Within Phases

Setup Phase (Can run in parallel):

- T004, T005, T006, T007, T008, T009, T012, T013, T014 (different files)

Foundational Phase (Can run in parallel after T019-T025):

- T026-T030 (custom post types - different files)
- T031-T034 (taxonomies - different files)
- T037-T039 (custom fields for different post types)

User Story 1 Implementation:

- T051, T052 (homepage and archive - different files)
- T058, T062, T063, T067 (CSS files - different component files)

User Story 7 Implementation:

- T068, T070, T071, T073, T074, T083, T084, T085 (independent files)

Polish Phase:

- T234-T237 (linting, testing - different tools)
 - T238, T239, T240, T241 (documentation tasks)
-

Task Count Summary

- **Phase 1 (Setup):** 18 tasks (T001-T018)
- **Phase 2 (Foundational):** 33 tasks (T019-T050, T029a) BLOCKING
- **Phase 3 (US1 - P1):** 17 tasks (T051-T067) MVP
- **Phase 4 (US7 - P2):** 18 tasks (T068-T085)
- **Phase 5 (US2 - P2):** 13 tasks (T086-T098)
- **Phase 6 (US3 - P2):** 12 tasks (T099-T110)
- **Phase 7 (US6 - P3):** 14 tasks (T111-T124)
- **Phase 8 (US4 - P3):** 13 tasks (T125-T137)
- **Phase 9 (US5 - P3):** 17 tasks (T138-T154)
- **Phase 10 (Search):** 15 tasks (T155-T169)
- **Phase 11 (Header/Footer):** 12 tasks (T170-T181)
- **Phase 12 (Feedback):** 11 tasks (T182-T192)
- **Phase 13 (Versioning):** 9 tasks (T193-T201)
- **Phase 14 (Performance):** 15 tasks (T202-T216)
- **Phase 15 (Treatment Template):** 8 tasks (T217-T224)
- **Phase 16 (Resources):** 9 tasks (T225-T233)
- **Phase 17 (Polish):** 28 tasks (T234-T260, T260a)

Total Tasks: 263

Parallelizable Tasks: 87 tasks marked [P] (33% can run in parallel with proper team structure)

MVP Scope Recommendation

Minimum Viable Product includes:

- Phase 1: Setup (T001-T018)
- Phase 2: Foundational (T019-T050)
- Phase 11: Header/Footer (T170-T181) - needed for basic navigation
- Phase 3: User Story 1 (T051-T067) - core parent content access
- Basic performance optimization (T202-T207 from Phase 14)
- Basic QA (T234-T237, T254-T255, T260 from Phase 17)

MVP Delivers: Parents can navigate from homepage to condition articles, read family-friendly content with FAQs, see related treatments, with basic responsive design and performance optimization.

Estimated MVP Task Count: 50 highest-priority tasks (~100-200 hours of development)

Post-MVP Priorities (in order):

1. Multilingual support (US7 - Phase 4)
 2. Search functionality (Phase 10)
 3. Professional content (US2 - Phase 5)
 4. Patient journeys (US3 - Phase 6)
 5. Mobile optimization (US4 - Phase 8)
 6. Accessibility compliance (US5 - Phase 9)
 7. Staff profiles (US6 - Phase 7)
 8. Remaining cross-cutting features
-

Implementation Strategy

Incremental Delivery Approach

Sprint 1 (MVP Foundation):

- Complete Setup + Foundational + Header/Footer
- Deliver: Working WordPress theme structure with custom post types

Sprint 2 (MVP Complete):

- Complete User Story 1
- Deliver: Parents can find and read condition articles

Sprint 3 (Multilingual):

- Complete User Story 7 + Search
- Deliver: Sinhala/Tamil support with working search

Sprint 4 (Professional Content):

- Complete User Story 2 + Treatment Template
- Deliver: Professional articles and treatment procedures

Sprint 5 (Enhanced Experience):

- Complete User Story 3 + User Story 6 + Resources
- Deliver: Journey modules, staff profiles, support resources

Sprint 6 (Mobile & Accessibility):

- Complete User Story 4 + User Story 5
- Deliver: Mobile-optimized, WCAG 2.1 AA compliant

Sprint 7 (Polish & Launch):

- Complete Performance + Feedback + Versioning + Polish
- Deliver: Production-ready theme with all features

Testing Strategy

- Run Theme Check plugin after each phase
 - Run accessibility scan (axe DevTools) after each user story
 - Performance testing after Phase 14
 - Cross-browser testing in Phase 17
 - User acceptance testing with sample content in Phase 17
-

Tasks Document Version: 1.0.0

Generated: 2025-10-16

Status: Ready for Implementation

Next Step: Begin Phase 1 (Setup) - Task T001