

# **COMPLETE REPORT & DEVELOPMENT PLAN: Jannah News Child Theme Redesign for Teznevisan3.com**

**Version:** 1.0 – February 21, 2026

**Goal:** Turn the current bloated, WP-Code-dependent Jannah News installation into a **clean, ultra-lightweight, 100/100 PageSpeed, perfect CWV, fully maintainable service-provider theme** using only the official child theme + Jannah's native shortcodes, Page Builder, and minimal custom code.

---

## **1. Executive Summary**

Your site is currently suffering from **severe technical debt** caused by:

- 9+ heavy PHP snippets ( $\approx$ 2,500+ lines) injected via WP Code
- Duplicate H1s, render-blocking inline CSS/JS, conflicting schemas, FA CDN + local duplication
- Service pages bypassing the theme entirely (full HTML render)
- Custom classes ( `.tez-*` ) overriding Jannah's responsive system

**Result:** PageSpeed <70, failing CWV, SEO damage, fragile updates, poor UX on mobile.

**Solution:** We migrate **everything** into the official `jannah-child` theme you already have. We keep **only** what Jannah cannot do natively (advanced ToC with meta box, poll system, visual sitemap, tag hub, 404 hub, service-page CSS system).

Everything else is replaced by Jannah shortcodes + Page Builder + Custom Content Blocks.

**Expected outcome after completion:**

- 100/100 PageSpeed (mobile + desktop)
  - Perfect Core Web Vitals (LCP <1.8s, CLS = 0)
  - Zero console errors
  - Duplicate H1s eliminated
  - Service pages built 100% inside Jannah Page Builder
  - Full RTL/Persian/dark/sepia support preserved
  - Future Jannah updates safe
- 

## 2. All Identified Problems (One by One)

### Performance & Render-Blocking

1. Inline `<style>` and `<script>` at `wp_head` / `wp_footer` priority 9999 (Typography, Single Post CSS/JS, Service Page, Star Rating, Remove Titles, Critical CSS) → blocks rendering.
2. Duplicated Font Awesome (local FA7 in Typography + CDN in Star Rating).
3. Service Page Template renders full page HTML outside theme → 200+ KB inline CSS/JS per page.
4. Multiple `setTimeout` + repeated `DOMContentLoaded` in JS.
5. No `defer` / `async` on custom JS.

### SEO & Indexing

1. **Duplicate H1s** everywhere (Jannah `.entry-title` + custom code).
2. Overlapping schemas (RankMath/Yoast + manual Article/AggregateRating from Star Rating + Service schema).
3. Service pages disable SEO plugins but inject own schema → validation errors.
4. Custom taxonomies (`tez_service_*`) not auto-included in Jannah sitemap.
5. Tracking-parameter URLs still leaking (even after your htaccess work).

### UI/UX & Theme Conflicts

1. .tez-\* classes break Jannah responsive grid and mobile menu.
2. Service pages have no Jannah header/footer consistency, no sticky elements, no mega menu.
3. Heavy !important + hardcoded colors break dark/sepia modes.
4. Custom back-to-top, breadcrumbs, reading progress duplicate Jannah features.
5. FAQ accordion, TOC, rating duplicate Jannah toggles/shortcodes.

## Maintainability & Security

1. All logic in WP Code → no version control, update conflicts, impossible debugging.
2. Service form AJAX has basic nonce but weak rate-limiting; file uploads to open folder.
3. Hardcoded emails in PHP.
4. No caching headers on custom assets.

## Redundancies vs Jannah Natives

1. Custom TOC → replaceable by Jannah [content\_index] + minor CSS.
  2. Star rating → replaceable by Jannah review shortcode.
  3. Many boxes/tabs/buttons → direct Jannah shortcodes.
  4. Page layouts → Jannah Page Builder + Custom Content Blocks + your existing Complete-Page-Templates-CSS.css .
- 

## 3. Current Assets Inventory (What We Keep / Adapt)

Keep & Integrate into Child Theme:

- Complete Page Templates CSS (design tokens + all .tez-service-\* , .tez-pricing-\* , etc.)
- Beautiful Table of Contents (meta box + auto-insert + advanced features)
- Complete Poll System (DB table + meta box + shortcode)
- Beautiful Visual Sitemap Page
- Beautiful Tag Hub Page Template
- Beautiful 404 Page Hub
- About/Contact Page Templates
- SEO URL Cleanup (tracking params)
- Irancell local fonts + FA7 local
- Service form AJAX handler (lightly hardened)

**Replace with Jannah Natives:**

- Remove Titles/Breadcrumbs → Jannah filters + post settings
  - Single Post enhancements → Jannah single layouts + shortcodes
  - Star Rating → Jannah review shortcode
  - Basic TOC/FAQ → Jannah [content\_index] + Toggle Boxes
  - Buttons/Boxes/Columns → Jannah shortcodes
  - Reading progress → simple CSS + one tiny JS
- 

## **4. Full Migration & Redesign Roadmap (Phased – 7 Days Max)**

### **Phase 0: Preparation (Day 1 morning)**

- Staging site ready (clone live)
- All WP Code snippets **deactivated**
- Jannah Extensions active
- WP Rocket / caching cleared
- Child theme activated (you already have the skeleton)

## Phase 1: Child Theme Skeleton & Core Setup (Day 1)

Create this exact structure:

```

text

jannah-child/
├── style.css
├── functions.php
├── js/
│   └── scripts.js
└── inc/
    ├── complete-page-templates.css
    ├── beautiful-toc.php
    ├── complete-poll.php
    ├── visual-sitemap.php
    ├── tag-hub.php
    ├── 404-hub.php
    ├── about-template.php
    ├── contact-template.php
    ├── seo-url-cleanup.php
    └── service-form-handler.php
└── templates/
    └── page-service.php (optional, only if needed)
└── screenshot.png (optional)

```

## Phase 2: Consolidate PHP (Day 1–2)

Move every PHP module into `inc/` and `require_once` in `functions.php` in this order:

1. seo-url-cleanup.php
2. complete-page-templates.css enqueue
3. beautiful-toc.php
4. complete-poll.php
5. visual-sitemap.php + tag-hub.php + 404-hub.php
6. about/contact templates
7. service form handler

### Phase 3: CSS & Design System (Day 2)

- @import or enqueue complete-page-templates.css in child style.css
- Align Jannah Theme Options → Primary color = #2563eb
- Typography → Irancell (local) + system stack (already in your CSS)
- Dark/Sepia via data-theme attribute (your CSS already supports)

### Phase 4: Service Pages & Layouts (Day 3)

- Create pages with **Jannah Page Builder**
- Use **Custom Content Blocks** + your .tez-\* classes
- Hero → full-width section + background image + Custom Content Block with H1 + USP boxes (Jannah Boxes shortcode)
- Features/Process → Columns + Boxes shortcode
- Pricing/FAQ/Testimonials → Toggle Boxes + Pricing cards from your CSS
- Form → drop your existing form shortcode inside a .tez-form-card

### Phase 5: Single Post & Content (Day 4)

- Override `single.php` in child if needed (rare)
- Use Jannah single layouts
- Reading progress → 4-line CSS + one JS snippet
- TOC → `[content_index]` shortcode (or keep your advanced one)
- FAQ → Jannah Toggle Boxes
- Rating → Jannah review shortcode

## Phase 6: Performance Lockdown (Day 5)

- WP Rocket: Combine CSS/JS, Delay JS, Remove unused CSS
- Critical CSS: only 2–3 KB (hero + navigation)
- All custom JS → defer via `scripts.js`
- Images → ShortPixel + WebP
- Target: 100/100 mobile + desktop

## Phase 7: Testing & Launch (Day 6–7)

- Full crawl (Screaming Frog)
  - GSC request indexing for all cleaned pages
  - AI crawler test (ChatGPT, Claude, Perplexity)
  - 7-day monitoring
- 

## 5. Exact Next Steps (Start Today)

### Step 1 (now):

Reply with “**Phase 1 ready**” after you have created the folder structure above and deactivated all WP Code snippets on staging.

### Step 2: I will give you the **exact code** for:

- Updated functions.php
- style.css (with imports)
- First inc/ file (seo-url-cleanup.php moved)

We proceed file-by-file, you paste → test → confirm → next.

**Step 3:** When all PHP is in, we build the first service page together using Page Builder + your CSS classes.

This is the cleanest, most maintainable path. No more WP Code hell. Full Jannah power + your beautiful service layouts.

Ready? Reply “**Phase 1 ready**” and we begin immediately.