SELECTED TECHNICAL WORK

WORDPRESS ENGINEERING PORTFOLIO

Custom Theme Architecture + React Property Platform

Matthew Moncada

WordPress Developer | Full-Stack Web Engineer | Platform Operations

WordPress | PHP | React | Redux | Google Maps | SCSS | npm

2	81	4,600+	2
FLAGSHIP PROJECTS	FIRST-PARTY FILES	NONBLANK LINES	ARCHITECTURE MODELS

PORTFOLIO THESIS

Two custom-coded WordPress systems that demonstrate engineering beyond page builders: a distribution-oriented classic theme built around core WordPress APIs, and a hybrid React application that connects external property data, centralized state, Google Maps, WordPress AJAX, custom post types, and media ingestion.

CONTACT

Colorado-based | (210) 322-4787 | matthewmoncada90@gmail.com

linkedin.com/in/matthew-moncada | github.com/mattcodingsensei

MM

PROFESSIONAL POSITIONING

A custom-code WordPress developer with evidence across themes, React, APIs, data modeling, and deployment workflows.

These projects show direct implementation of WordPress internals rather than template-only configuration. The work spans PHP theme architecture, template hierarchy, reusable components, accessibility, internationalization, React state management, external API integration, geospatial interfaces, server-side persistence, asset compilation, and local development workflows.

01 Core Theme Engineering

Built a classic theme from first principles using WordPress setup hooks, template hierarchy, reusable template parts, custom navigation logic, asset enqueuing, comments, archives, search, and custom page templates.

02 Hybrid Application Architecture

Embedded a React and Redux application inside WordPress, routing user searches into a map interface while synchronizing external listing data with WordPress content storage.

03 Quality and Distribution

Prepared GPL licensing, theme metadata, translation resources, documentation, optimized assets, local dependencies, semantic markup, and accessibility-oriented controls.

04 Platform and Delivery

Worked across MAMP/local WordPress environments, npm builds, Sass, Webpack, Gulp, PHP, browser APIs, CMS data models, and production-oriented deployment concerns.

ENGINEERING SIGNAL

The strongest evidence is architectural range: one project works within the native WordPress rendering model, while the second treats WordPress as an application platform and persistence layer for a React-driven product experience.

matthewmoncada90@gmail.com | github.com/mattcodingsensei

Project Inventory

Native theme engineering and full-stack application integration.

MoncadaPress

Distribution-oriented classic WordPress theme. Implements native template resolution, reusable PHP partials, accessibility controls, translation support, local assets, and a documented optimization workflow.

Property Discovery Platform

Zillow-inspired prototype combining React, Redux Toolkit, React Router, Axios, Google Maps, WordPress AJAX, a custom property post type, metadata, and remote media ingestion.

ARCHITECTURE COMPARISON

PROJECT	RUNTIME	ARCHITECTURE	CORE SYSTEMS
MoncadaPress	WordPress/PHP	Classic theme + templates	Theme APIs, hierarchy, partials, i18n, accessible navigation, asset pipeline
Property Platform	WordPress + browser	React SPA embedded in theme	Routing, Redux state, API data, Maps markers, AJAX persistence, CPT/media

SHARED ENGINEERING CONCERNS

REQUEST
WordPress route / user query

STATE
CMS content / Redux store

LOGIC
PHP hooks / React effects

OUTPUT
HTML templates / map UI

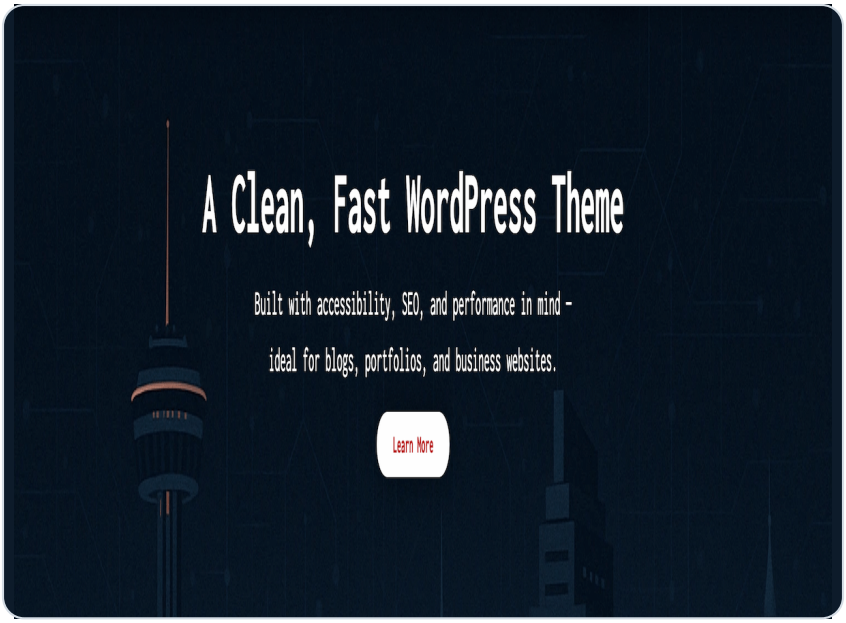
SOURCE REVIEW SNAPSHOT

81 first-party files 4,619 nonblank lines PHP + JS + CSS/SCSS

matthewmoncada90@gmail.com | github.com/mattcodingsensei

MoncadaPress Classic Theme

A complete classic-theme architecture built for responsive publishing, accessibility, SEO, and distribution.



APPLICATION SCOPE

3,983 nonblank lines

- 28 PHP files covering core templates, custom page templates, template parts, and a navigation walker.
- 28 modular CSS files plus compiled production CSS.
- JavaScript interactions for carousel behavior, filters, FAQ controls, mobile navigation, scroll effects, and back-to-top UX.
- Translation catalog, GPL license, credits, theme metadata, screenshot, and distribution documentation.

TEMPLATE ARCHITECTURE

front-page.php
Composes six reusable homepage sections

home.php
Blog index

single.php
Single post

archive.php
Archive views

page.php
Standard pages

comments.php
Comment workflow

front-page.php - component-oriented composition

```
get_template_part( 'template-parts/front-page/hero' );
get_template_part( 'template-parts/front-page/workflow' );
get_template_part( 'template-parts/front-page/wordpress-benefits' );
get_template_part( 'template-parts/front-page/services' );
get_template_part( 'template-parts/front-page/capabilities' );
```


matthewmoncada90@gmail.com | github.com/mattcodingsensei

Theme Standards and Distribution

WordPress-native implementation with quality, accessibility, and maintainability built into the theme structure.

WordPress Core Integration

Theme setup uses add_theme_support(), register_nav_menus(), wp_enqueue_style(), wp_enqueue_script(), load_theme_textdomain(), wp_nav_menu(), get_template_part(), WP_Query, pagination, comments, and core template functions.

Accessibility-Oriented UI

Includes skip navigation, semantic landmarks, ARIA labels, aria-expanded FAQ controls, screen-reader text, accessible menu attributes, keyboard-aware carousel links, responsive navigation, and focus-state handling.

Internationalization and Portability

Uses a dedicated text domain, translation and escaping functions, a /languages directory, generated POT catalog, relative theme paths, versioned asset enqueues, and WordPress template conventions.

Build and Distribution Workflow

Includes source CSS modules, minified CSS/JS outputs, WebP assets, locally hosted fonts and libraries, CLI-based image optimization, documented npm commands, GPL licensing, theme metadata, and deployment instructions.

REPRESENTATIVE SOURCE

functions.php - theme setup and platform features

```
add_theme_support( 'title-tag' );
add_theme_support( 'post-thumbnails' );
add_theme_support( 'html5', array(...));
add_theme_support( 'custom-logo' );
load_theme_textdomain( 'moncadapress', ... );
register_nav_menus( array( 'primary' => ... ));
```

Custom walker - accessible dropdown metadata

```
if ( $has_children ) {
  $atts['aria-haspopup'] = 'true';
  $atts['aria-expanded'] = 'false';
}
$atts = apply_filters(
  'nav_menu_link_attributes', ... );
```


matthewmoncada90@gmail.com | github.com/mattcodingsensei

Property Discovery Platform

A Zillow-inspired property search prototype connecting React, geospatial UI, external data, and WordPress persistence.

APPLICATION SCOPE

636 nonblank lines

- React application mounted into WordPress template containers.
- Redux Toolkit store for shared property data.
- React Router and query-string driven location searches.
- Google Maps advanced markers and information windows.
- Synchronized map selection, highlighted cards, recentering, and smooth sidebar scrolling.
- WordPress custom post type, post metadata, AJAX ingestion, and remote featured-image handling.

END-TO-END DATA FLOW

SEARCH	API	STATE	MAP	AJAX	CMS
Address / city / ZIP	External listing results	Redux property store	Markers + cards	JSON to WordPress	Property posts + media

ENGINEERING VALUE

The project demonstrates a complete data path across browser interaction, asynchronous API retrieval, state normalization, map rendering, event synchronization, server-side processing, CMS content modeling, and media management.

matthewmoncada90@gmail.com | github.com/mattcodingsensei

React, PHP, and Delivery Pipeline

Frontend state, geospatial interaction, backend persistence, and build tooling in one WordPress application.

React Application Layer

Uses hooks, useEffect, useRef, route-aware query parsing, controlled filters, property normalization, conditional rendering, and map/list synchronization. Redux Toolkit centralizes the listing collection for application-wide access.

WordPress Persistence Layer

Registers a public property custom post type, accepts serialized listing data, sanitizes fields, inserts posts with structured metadata, returns JSON responses, and sideloads remote property images as featured media.

React - derived state and synchronized interaction

```
const filtered = properties.filter((property) => {
  const price = normalizePrice(property.price);
  return price >= min && price <= max;
});
setCenter({ lat: property.latitude,
            lng: property.longitude });
propertyRefs.current[id].scrollIntoView(...);
```

PHP - custom content persistence

```
$post_id = wp_insert_post(array(
  'post_type' => 'property',
  'post_title' => $street_address,
  'post_status' => 'publish',
  'meta_input' => array(
     'price' => $price, 'city' => $city
  ),
));
```

BUILD AND ENVIRONMENT

@wordpress/scripts
Webpack/Babel production bundle

npm
Dependency and script management

Sass + Gulp
Modular styles and CSS generation

MAMP
Local Apache/MySQL/PHP WordPress environment

PRODUCTION HARDENING AWARENESS

A production version would move third-party credentials and listing requests behind a server-side proxy, use environment configuration, add nonce and capability checks to write operations, validate remote hosts and schemas, deduplicate listings, and version assets from the generated WordPress dependency manifest. This reflects the ability to distinguish prototype delivery from production security requirements.

matthewmoncada90@gmail.com | github.com/mattcodingsensei

WordPress Engineering Competency Matrix

Evidence of theme engineering, application development, platform integration, and operational delivery.

DEMONSTRATED DIRECTLY IN THE UPLOADED SOURCE

Theme Architecture

Template hierarchy, hooks, setup, enqueueing, reusable PHP parts

PHP and CMS Data

Custom post types, WP_Query, post metadata, AJAX, media ingestion

React Engineering

Hooks, routing, Redux Toolkit, derived state, component mounting

API Integration

Axios, external listing data, query parameters, asynchronous flows

Geospatial UI

Google Maps, advanced markers, info windows, map/list synchronization

Accessibility

Semantic landmarks, skip links, ARIA, keyboard and focus behavior

Internationalization

Text domains, translation functions, POT catalog, portable templates

Build Tooling

npm, Webpack/Babel, @wordpress/scripts, Sass, Gulp, minification

SUPPORTING WORDPRESS PLATFORM TOOLKIT

FRONTEND

JavaScript, React, HTML5, CSS/SCSS, Tailwind CSS, Astro, responsive UI

WORDPRESS

Classic themes, Gutenberg, Advanced Custom Fields, custom templates, plugins

LOCAL DEVELOPMENT

MAMP, VVV, Docker, npm, Git/GitHub, Nano, Linux command line

WEB SERVERS

Apache, Nginx, PHP environments, local-to-hosted deployment differences

HOSTING AND DNS

Domains, DNS records, SSL, MX email records, migrations, backups, security

ANALYTICS

GA4, Google Tag Manager, Search Console, event tracking, KPI validation

ENGINEERING VALUE PROPOSITION

Matthew brings the practical depth of a WordPress specialist and the architectural range of a software engineer. These projects show the ability to work inside the CMS conventions, extend the platform with custom PHP, embed modern React experiences, connect external services, model persistent content, build and optimize assets, and reason about accessibility, security, and deployment as one system.

matthewmoncada90@gmail.com | github.com/mattcodingsensei