

In the interest of employer confidentiality and non-disclosure agreements with prior freelance clients, much of my professional work cannot be made publicly-accessible via my website, nor can I post code to a public GitHub repository.

Please see below for a sampling of my recent professional work, as well as links to private GitHub repos provided via GitFront.io. **Please note that linked code examples only contain relevant files that I worked on, rather than the entire codebase, and have been edited to remove any sensitive data. As such, they will not run on a standalone basis.**

PROFESSIONAL PORTFOLIO

GravityForms Custom Integration with Salesforce MarketingCloud

In order to support the marketing department's transition to a new back-end CRM platform, I built out a custom solution that allows existing 'request-for-information' forms on our website to properly integrate with Salesforce MarketingCloud (SFMC), along with other various platforms.

The forms on our website are considered "mission critical" to the marketing department, and run on a highly-customized version of WordPress GravityForms. While our prior CRM platform integrated seamlessly with this, the integration with SFMC posed two major challenges: First, SFMC's API requires authentication via a dynamically-generated access token which expires every 20 minutes. Second, SFMC's API requires payload data be structured in a very specific format of nested JSON, far beyond what GravityForms is capable of on its own.

To solve these challenges, I developed a standalone custom WordPress plugin that extends GravityForms via a series of action hooks and filters. This plugin handles access token generation automatically, storing the token in the database and only generating a new one if needed to complete the submission. The plugin also handles structuring the API payload correctly, and in a manner that is scalable and easily configurable on a form-by-form basis.

 [Code Example](#)

Wisepops Custom Integration with GravityForms

As a follow-on project to the above, engineered a custom back-end solution that allows form responses captured by a third-party platform (Wisepops) to submit to our website, undergo standard data processing we run for normal form submissions, and then route to our third-party CRM platform. The main challenge here was to get 3 separate platform APIs to play nicely with each other, in a manner that still satisfied broader business process requirements.

 [Code Example](#)

SMS Integration with RFI forms

Worked with university enrollment & marketing teams to implement an automated series of SMS text message follow-ups for certain 'request-for-information' forms on our site. This feature is conditionally-enabled, and only happens if the prospective student indicates interest in either a business- or psychology-related degree program within the form. SMS #1 is sent to the prospective student immediately upon form submission, and SMS #2 is then sent approximately 30 minutes later and contains a link to schedule an appointment with an enrollment advisor. Additionally, I adjusted

existing data integrations with numerous third-party platforms used by our enrollment & marketing teams, to automatically flag SMS-enabled form entries as such.

This feature was built as a standalone WordPress plugin, and has a couple major moving parts: It involves heavy integration with Gravity Forms (our existing RFI forms have a high degree of customizations/modifications built on top of Gravity Forms); usage of a third-party API to send automated text messages; custom database entries and the WP-Cron Job API to schedule and send the delayed SMS #2; creation of a custom settings page within wp-admin; and various hooks & filters to conditionally-modify data that integrates with various external data platforms.

 [Code Example](#)

Trustpilot Encrypted Data Link API

Worked with our internal Email Marketing team to create a custom API endpoint on our site that allows the team to send email links that allow students to submit a review on trustpilot.com in a “frictionless” manner, in that it won’t require them to sign in or create an account. For this, Trustpilot requires that certain personal information for each submission is included as encrypted data in a URL parameter.

To achieve this, the initial email link points to our site with the student data included as URL parameter, auto-generated by the email marketing software. Our API will then parse the URL parameters and encrypt them in accordance with Trustpilot’s developer documentation, and then immediately forward the user to the Trustpilot site with the encrypted URL parameter.

 [Code Example](#)

Custom Metabox Database Converter

In an effort to migrate multiple WordPress sites from a legacy ‘custom field’ plugin to a more modern solution, I developed a custom ‘migration’ plugin to ensure no existing data or content would be lost in the transition. Plugin scans the WordPress database for legacy custom fields, converts the data into a structure that’s compatible with the new custom field solution, and then deletes the legacy data. Plugin was built to be easily-extendable & reusable across all sites (more than a dozen) that we manage.

 [Code Example](#)

WordPress Multisite Management Plugin

For an existing WordPress multisite installation, created a network-level plugin to centralize certain data & functions that were previously replicated across all network sites. Plugin allows a non-technical site admin to easily manage & customize various third-party integrations (such as Google Tag Manager) on a site-by-site basis.

 [Code Example](#)

PROFESSIONAL PORTFOLIO – CONTRACT WORK

E-Commerce Software API Version Updates

Worked with an industrial parts supply company that uses a software platform that was custom-developed for them over 10 years ago. This platform was using outdated/deprecated versions of various third-party APIs, to the extent that it was causing significant delays in fulfilling product orders. I successfully updated their platform to use a current version of certain “mission critical” APIs, in a way that avoided having to re-write a significant amount of their legacy code.

WordPress Dynamic Post Filter & Search via Custom REST API

For multiple existing WordPress sites with hundreds of pre-existing posts stored across numerous custom post types and with numerous custom meta fields & taxonomies, I built out functionality to allow site visitors to dynamically search & filter results across multiple parameters. I accomplished this by building out a series of custom REST API endpoints on the WordPress backend, and then implementing an AJAX-based filter on the front-end. Solution was built to scale, with the ability to handle multiple different post types, meta fields, as well as account for future client additions.



[Live Example #1](#)



[Live Example #2](#)



[Live Example #3](#)



[Code Example](#)

Toyota Career Expo – Applicant Progress Tracker

For Toyota’s 2021 “Drive to Employment” career expo microsite, I built out a basic progress-tracking system for registered users. The broader site was pre-existing and built in WordPress, utilizing a third-party “user-registration/membership” plugin. I built in extra functionality on top of this, to track and save user progress as they advance through various sections of the site. Additionally I also added functionality to allow registered users to ‘bookmark’ pages, posts, and videos to their profile page for later viewing.



[Live Site](#)



[Code Example](#)

Digital Advertisement Scraper App

Assisted in building a full-stack JavaScript app that takes a user-entered URL and scrapes that site for certain advertising-related content. For each result found within the scraped site, various metadata about the advertisement is saved, along with a screenshot and a timestamp. Results are saved to a database which can then be viewed, edited, or deleted by the admin user. Uses Strapi.io and Node.js on the back-end, SQLite as the database, and React.js on the front-end.



[Code Example](#)

PROFESSIONAL PORTFOLIO – CONTRACT WORK (cont.)

ErosSTX

Took an existing WordPress site which had been previously built by another developer with the Divi page builder and converted it to a custom-built solution, while still maintaining the exact same look-and-feel of the site. Due to the media-intensive nature of the site, this conversion resulted in a significantly improved page speed and performance of the site.



[Live Site](#)

Sultre

Built out a brand-new e-commerce site for an upscale brick-and-mortar boutique fashion retailer. Site was built upon the BigCommerce back-end platform and included numerous integrations with third-party services, including inventory management systems and the store's physical point-of-sale system. Site is currently generating over \$1,000 in average daily sales.

(note: **potentially** not-work-safe – women's fashion retailer that sometimes features swimsuit models)



[Live Site](#)

Various Custom WordPress Builds

From 2020 through 2022, I completed dozens of custom WordPress builds for various agency clients. All sites feature custom-built themes (usually based on either Underscores + Bootstrap.css, or an internal agency boilerplate theme) based on high-fidelity mockup designs, with a general goal of avoiding third-party plugin bloat. Most sites make heavy use of ACF and Custom Post Types, and are designed to allow a non-technical end-client to manage the site content on an ongoing basis. Please see below for a sampling:

(Please note, many of these were developed 2+ years ago at this point. Current versions of these sites may no longer reflect my work)

- [M4BL.org](#)
- [TheRisingMajority.com](#)
- [BrightideasFL.com](#)
- [RootandRebound.org](#)
- [Proesolar.com](#)
- [CareerSourceCentralFlorida.com](#)
- [LarrysGutters.com](#)
- [DefundPolice.org](#)
- [DiscoverGuruNanak.com](#)
- [OurCourtsOurFight.com](#)