



Mark Vicuña

Customizing WordPress for Enterprise Cases

Metropolia University of Applied Sciences

Bachelor of Engineering

Degree Programme in Information Technology

Bachelor's Thesis

7 May 2022

Abstract

Author: Mark Vicuña
Title: Customizing WordPress for Enterprise Cases
Number of Pages: 41 pages + 1 appendix
Date: 7 May 2022

Degree: Bachelor of Engineering
Degree Programme: Information Technology
Professional Major: Software Engineering
Supervisors: Janne Salonen, Head of Department
Joni Polvi, Production Leader

The final year project presented in this thesis and its project were carried out in cooperation with Liana Technologies, a company specializing in digital marketing services and software. The goal of the thesis was to propose a solution for web developers at Liana to begin building enterprise-level websites using WordPress, a popular open-source content management system (CMS). This was achieved by developing a custom theme for WordPress, as well as compiling a list of recommended plugins to extend the platform's default functions.

The thesis investigated the current WordPress landscape using a set of websites inherited by the company through site-migration cases. These cases offered an opportunity to analyze the methods used to build contemporary websites with WordPress, and evaluate them alongside the project requirements. These requirements were to create a WordPress site that mirrored the functionality and usability of an enterprise site built on the company's proprietary system, LianaCMS. The project's maintainability and relevance to current development practices at Liana was also considered.

The project resulted in the creation of a demo WordPress site exhibiting the developed theme and plugin selection. The theme adds support for new content types, as well as integrating with WordPress's Gutenberg block-system. In extending WordPress's capabilities to build an enterprise-level site, the plugins that proved most effective were Advanced Custom Fields Pro, Polylang, WooCommerce, and Contact Form 7.

Keywords: WordPress, content management system, web development, themes, plugins

Contents

List of Abbreviations

1	Introduction	1
2	Content Management Systems	2
2.1	LianaCMS	2
2.1.1	Managing Content	3
2.1.2	Templates and Common Elements	5
2.1.3	Multilingual Websites	6
2.2	WordPress	6
2.2.1	Posts and Pages	7
2.2.2	Custom Post Types and Taxonomies	8
2.2.3	Plugins	9
2.2.4	Themes	10
2.2.5	Gutenberg	11
2.2.6	Updates	12
3	Project Scope and Requirements	13
3.1	Functionality	14
3.2	Usability	14
3.3	Maintainability	15
3.4	Learning	15
4	Current State Analysis	15
4.1	Client A	16
4.1.1	Elementor Pro	16
4.1.2	WooCommerce	17
4.1.3	Polylang	18
4.1.4	Advanced Custom Fields	18
4.1.5	Contact Form 7	19
4.2	Client B	20
4.2.1	Custom Theme and ACF Pro	20
4.2.2	Active Plugins	21
4.3	Comparing Strategies	21
4.3.1	Page Builders	21

4.3.2	Leveraging ACF Pro	22
4.3.3	Additional Plugins	23
5	Methods	24
5.1	Tools	24
5.1.1	Local WP	24
5.1.2	VScode	24
5.1.3	Node.js and npm	25
5.2	Theme Repository	25
5.2.1	Wireframe	25
5.2.2	Bootstrap 5	27
5.2.3	WP Pusher	27
5.3	Required Plugins	27
5.3.1	Advanced Custom Fields Pro	27
5.3.2	Font Awesome 4	28
6	Results	29
6.1	Contacts, News, and Events	30
6.2	Lianatech Blocks	32
6.3	Page Settings	33
6.4	Menus and Global Elements	35
6.5	Shop, Contact Forms, and Language Translations	37
7	Conclusion	40
	References	42
	Appendices	
	Appendix 1: WordPress Template Hierarchy	

List of Abbreviations

CMS: Content Management System

HTML: HyperText Markup Language

CSS: Cascading Style Sheets

PHP: HyperText Preprocessor

SaaS: Software as a Service

SEO: Search Engine Optimization

ACF: Advanced Custom Fields

npm: Node Package Manager

JS: JavaScript

SCSS: Sassy Cascading Style Sheets

JSON: JavaScript Object Notation

1 Introduction

Liana Technologies is a European company specializing in digital marketing and communication software. The company was founded in 2005, with the aim of developing a set of tools to facilitate the work activities of marketing and communication professionals. Their technology stack, now used by over 3,500 customers worldwide, includes solutions for managing email marketing, press releases, and online stores to name a few. [1]

In 2006, Liana published their first product, a content management system (CMS) called LianaCMS. [2] The intention behind LianaCMS was to provide a platform for users to manage their website's content through an easy-to-use interface, without prior programming knowledge. [3] Today, the company continues to build custom websites for clients with LianaCMS. The scale of the website can range anywhere from small landing pages using pre-set templates, to larger tailored enterprise-level sites.

As of late 2021, Liana began offering website builds using WordPress, a widely popular open-source CMS. [4] Currently, the company primarily offers WordPress sites built around pre-made themes or templates which are selected by the client, allowing for basic editing of site content and design. In the near future, Liana plans to introduce solutions for larger-scale WordPress builds involving higher degrees of customization, namely "enterprise cases". As such, this thesis examines the current state of WordPress and the different methods employed to build sites; in order to propose an approach for developers at Liana to build enterprise-level WordPress websites.

This thesis is divided into 7 sections. Following the introduction, Section 2 provides an overview of content management systems with a detailed look into LianaCMS and WordPress processes. Section 3 establishes the project scope and requirements, after which Section 4 evaluates current approaches to building WordPress sites. Section 5 presents the methods used to implement

the project and Section 6 describes its results. Lastly, Section 7 offers concluding remarks regarding the contents of the thesis.

2 Content Management Systems

A content management system or CMS, is a piece of software that allows users to create, edit, and publish content on a website without requiring any programming or in-depth technical knowledge. [5] A CMS generally provides a graphical user interface that equips users with a broad set of tools to easily manage the content of a site, as well as control its appearance. Among the most popular content management systems being used with websites today are WordPress, Joomla, and Drupal. [6] All three are open-source platforms, which means they are publicly accessible, and the underlying source code is available to be inspected and extended. [7]

Although managing site content on a CMS does not typically require technical skill, creating the website's structure and appearance is generally done by an experienced developer. The framework for the site is built so that content added by users in the CMS is appropriately reflected in the website's front-end and adheres to its predetermined design. The methods and technologies used to develop a site will vary depending on the content management system in use. Nevertheless, in line with the fundamentals of web development, the process generally involves writing HTML, CSS, and JavaScript code. [5]

2.1 LianaCMS

At Liana Technologies, websites are built for a variety of clients using the company's proprietary content management system, LianaCMS. This work is carried out by a group of front-end developers that make up the Websites team within the Liana SaaS unit. These developers, referred to as "builders", create sites based on specifications outlined by project managers and designs submitted by the design team; all done in accordance with the client's requirements.

2.1.1 Managing Content

With LianaCMS, users can display various types of content on their website depending on the features sold to the client. Contact, event, product, and news items are available to users, each with a set of properties that can be used to add details to them. For example, in the case of an event, this would mean adding its start and end time through an on-screen date picker. These items can then be selected to display individually or collectively on a page with their own markup and style.

The structure of a site page is assembled within LianaCMS using predefined elements developed by the builders. These elements may have custom properties attached to them, where users can add content to the element as well as tweak their appearance. Managing a site's pages inside the CMS is done through the central page view, showcased below in Figure 1.

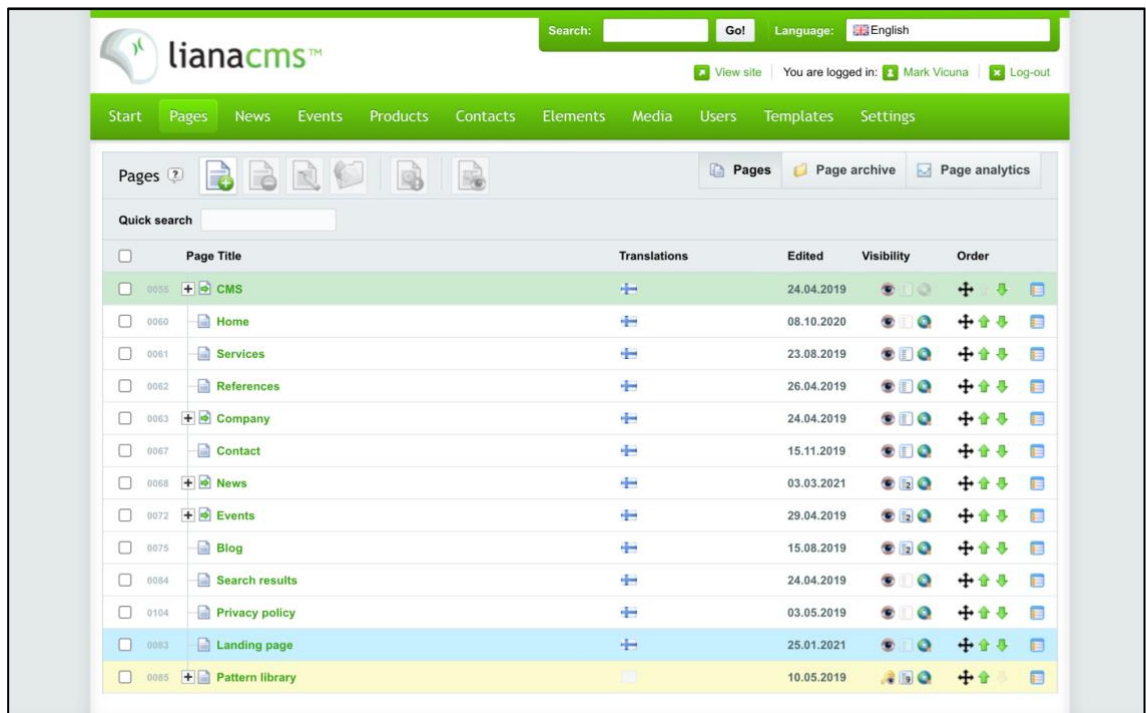


Figure 1. LianaCMS's admin page view.

Within this view, users create a hierarchy of pages and toggle basic settings such as page, menu and SEO visibility. Navigating into a page opens up the page editor view, displayed in Figure 2, where users may edit page settings in more depth and access the interface for adding structural elements to the site.

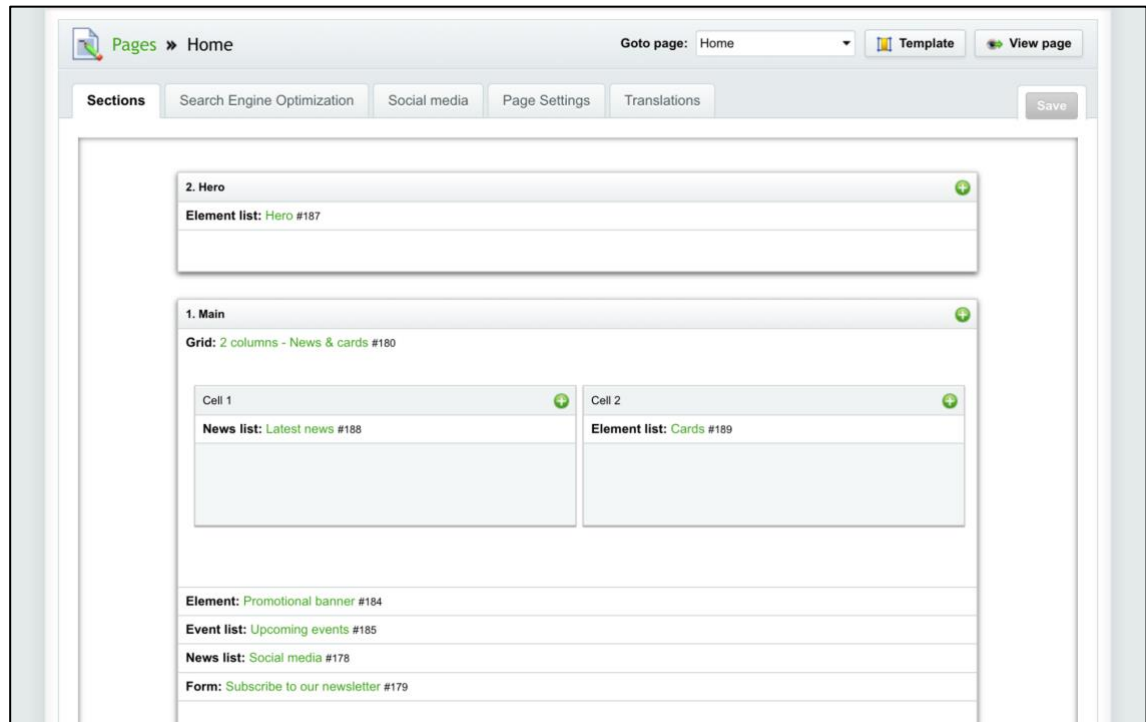


Figure 2. The page editor view in LianaCMS.

This interface, showcased above, allows users to add elements to different sections of the page layout in a sequential manner. As displayed in Figure 2, these elements can be forms, news lists, and banners among others. The sections and elements available to users in this view are informed by the site's design and requirements. The builders set them up and configure the properties associated with them. Figure 3 below depicts a list of properties attached to a promotional banner element on a LianaCMS sample site.

Banner - Full width >> Promotional banner

Name: Promotional banner

Settings

Properties

0044 **Content tag**
Short Text IT consulting 13 chars

0013 **Heading**
Short Text Successful client story 22 chars

0014 **Text**
Text We work with you and carry out research to understand your needs and the needs of your customers. We then identify the best way or ways for these needs to be met and enable you to implement this effectively. 207 chars

0017 **Banner image ***
Image Image size 991x400 pixels rawpixel-250087-unsplash.jpg Open folder

Figure 3. The properties associated with a promotional banner element.

The properties on display in Figure 3 are essentially fields that can be linked to custom elements. LianaCMS has a wide variety of properties available for use, such as text fields, links, media uploads, and multiple-choice selectors. From the user's end, elements and properties form the basis of site content creation in LianaCMS.

2.1.2 Templates and Common Elements

Much of the page structure is defined by the page template in use. A variety of page templates can be developed for users to customize the layout of their content. Within templates, builders create the global site areas such as the header and footer that recur on every page. While these are generally static areas, certain elements can be added dynamically in the code, such as the menus for a site's navigation areas. These global elements are often managed

through a separate hidden page created in the central page view, named “Common elements”.

2.1.3 Multilingual Websites

LianaCMS has integrated support for translating a website into multiple different languages. This process involves registering another language in the CMS’s language settings view, after which pages and elements can be duplicated and manually translated. A translations database page is also accessible in the CMS’s back-end, where builders can register translatable text strings used in various areas of the site. Over in the front-end, switching between languages is typically achieved through a custom dropdown element built into the site’s header.

2.2 WordPress

WordPress is the most widely used content management system to date, estimated to be running on 43% of all websites on the internet. [4] The first version of WordPress was launched in 2003, with the goal of creating an elegant open-source publishing platform built on PHP and MySQL. [8] The CMS was originally intended to be used as a blog-publishing platform, but as its functionality evolved, it allowed for creating websites such as forums, portfolios, and online stores. [9, p. 5-11]

As WordPress is an open-source platform, the general public is encouraged to customize and extend it. [10] A community has grown around WordPress, and due to the high availability of documentation and tutorials, contributions are regularly made to the platform’s core functionality in terms of new features and add-ons. According to Sitecare, the popularity of WordPress is largely due to the support of the community and the ease with which the CMS’s functions can be tailored to meet a range of site requirements. [11]

2.2.1 Posts and Pages

Given WordPress's origins as a blogging platform, the only content types available in default installations of the CMS are "posts" and "pages"; where a post is a single blog post item, and a page is an individual page on the website. [12] A standard WordPress site will therefore be made up of posts and pages created in the back-end of the platform, otherwise known as the WordPress admin area. [9, p. 57] This area or view is showcased in Figure 4.

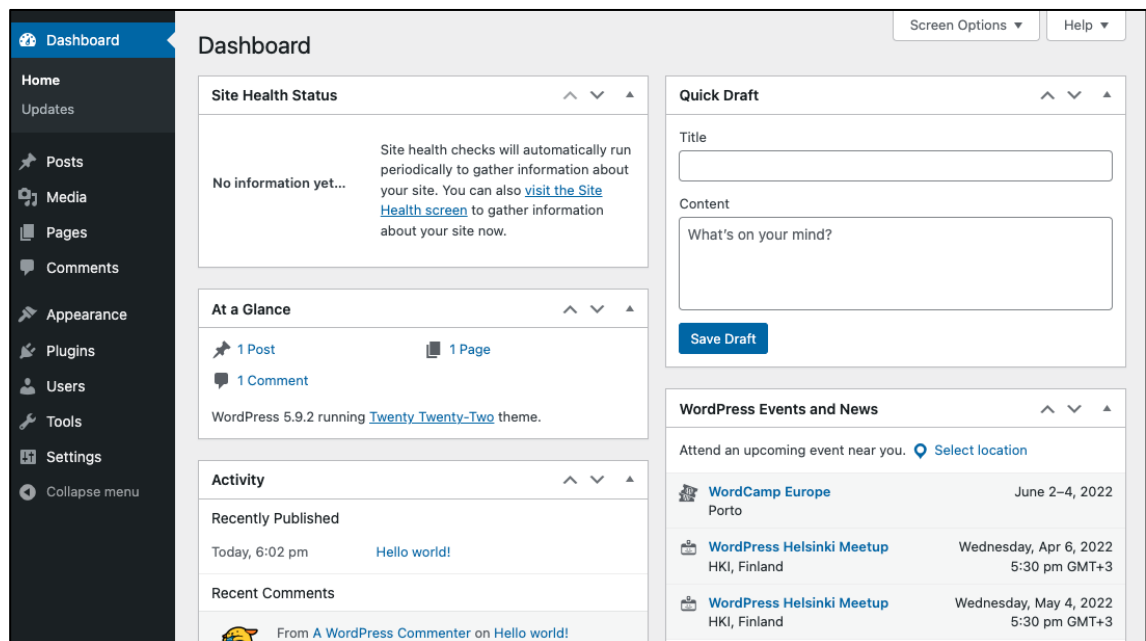


Figure 4. The WordPress admin area on a new installation.

As seen in Figure 4, posts and pages have their own menu tabs in the admin area's sidebar which provide access to various settings and management views. The editor view for posts is presented in Figure 5.

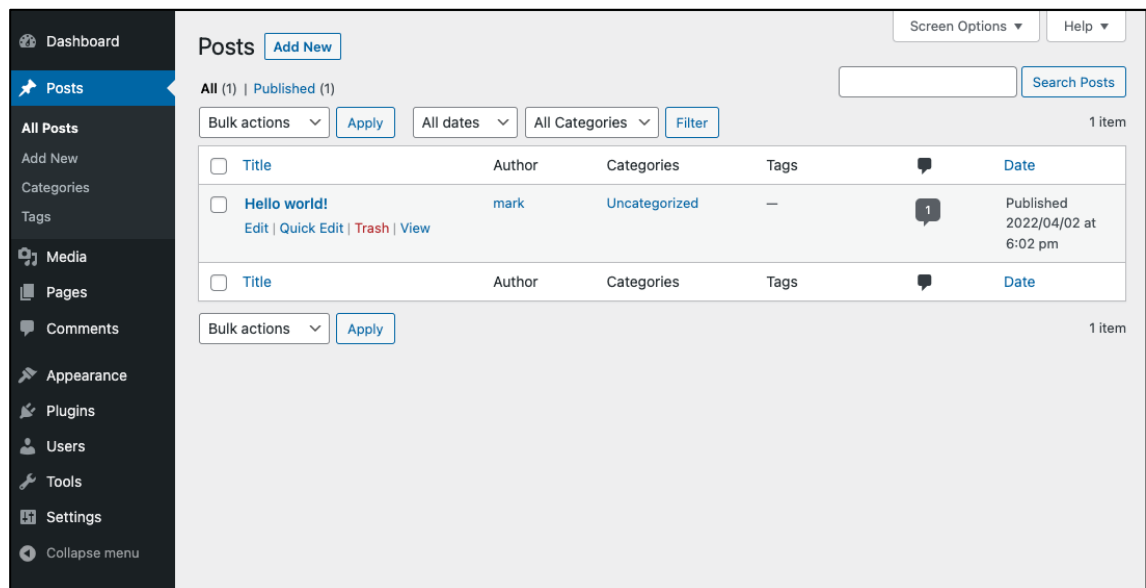


Figure 5. A WordPress post in the post editor view.

The post editor view houses an interface for users to edit and add posts to their site. This view, which is similar for pages, contains information about a post's author, publishing date, and any associated tags or categories. Categories and tags are types of taxonomies available in WordPress that allow users to group and organize content. [13]

2.2.2 Custom Post Types and Taxonomies

In WordPress, a “custom post type” can be registered on an installation in order to house content that is not suited to the default post or page classification. [14] For example, a culinary blog may see fit to add a custom post type of “recipe”; in order to store food recipes on the site. Adding a new post type creates a sidebar menu item akin to those previously displayed in Figure 5, where it can be managed in the same way as posts and pages. Similar to post types, custom taxonomies may also be registered to group content in more appropriate ways. [15] Expanding on the culinary blog example; a “course” taxonomy may prove useful in categorizing posts of type “recipe” into appetizer, main, and dessert groupings.

2.2.3 Plugins

Adding custom post types and taxonomies present one method of extending WordPress's default capabilities to meet site-specific requirements. Although building upon the CMS's functionality requires technical knowledge and an understanding of the platform's inner workings, users have the option to extend WordPress with "plugins". Plugins are ready-made extensions that can be activated on an installation, adding extra functionality to core WordPress operations. [16] Among the most widely used plugins for WordPress are Contact Form 7, a plugin that allows users to build forms on their site; and WooCommerce, which aids in setting up an online store. [17]

Plugins are also managed through the WordPress admin area. A plugin directory view is available, where users can install plugins directly onto their site. This view is showcased in Figure 6.

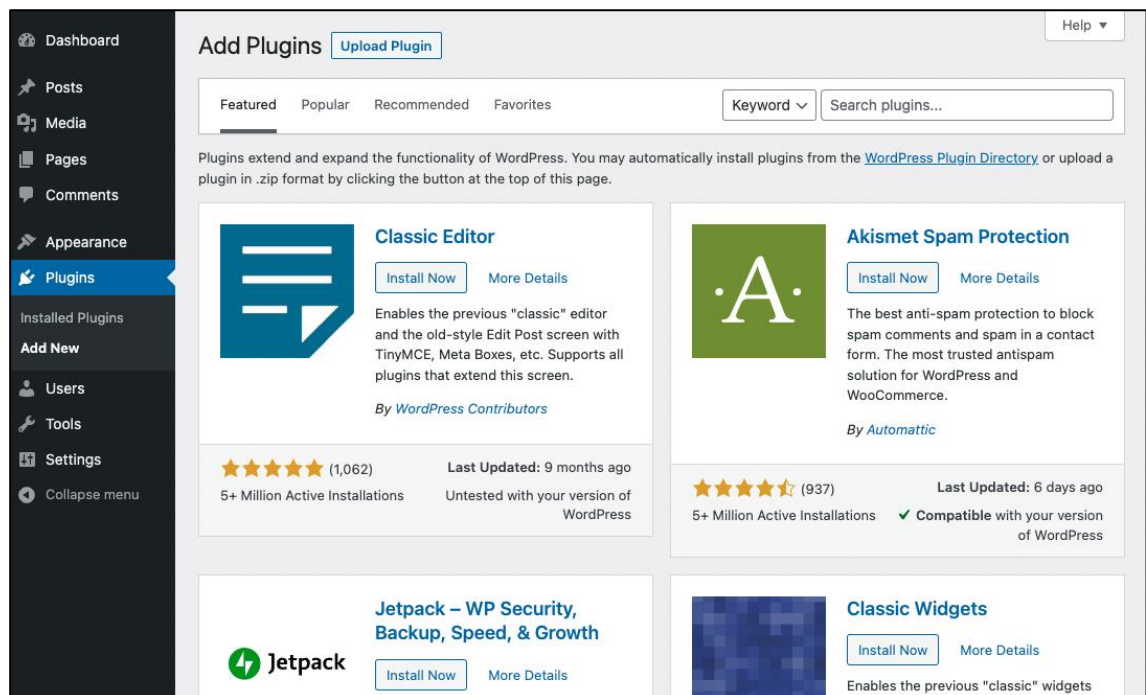


Figure 6. The WordPress plugin directory.

As shown in Figure 6, the plugin directory may be used to search for plugins as well as manually upload plugin files to the CMS. While there is no limit to the number of plugins that can be installed on a site, having an excessive amount of plugins active may impact the website's performance and bring forth security issues. [9, p. 256]

2.2.4 Themes

Whereas plugins extend the functions of a WordPress site, the site's layout and visual style is determined by the active "theme". As with plugins, themes are managed through a dedicated page in the WordPress admin area. Themes can be browsed, installed, or manually uploaded within the theme directory. Only a single theme can be active at once, and users should be able to switch between themes to change the appearance of their site without restricting any functionality. Though themes often include code that adds extra features to the site, it is recommended to separate site appearance from site function by developing these features into a custom plugin. [19]

A wide assortment of themes are openly available for use with WordPress, which are largely created by external parties. Scepter Marketing estimates there are over 30,000 themes in circulation today. [20] While themes fulfil a variety of user needs and site requirements, there are also themes geared towards developers. Such an example is a minimal theme named "Blank Canvas" which has no styling applied to it and only provides a basic structure for the site, lending itself as a starting point for building a custom theme. [21] Approaches such as this one may help to offset the time and effort required to build a theme entirely from scratch.

At its core, a theme is made up of a collection of PHP files and a CSS stylesheet, which define the design and structure of the various areas of the site. Themes are built around WordPress's templating system, which serves these files based on page requests sent to the server hosting the installation. [18] For example, if a site visitor navigates to an archive page, WordPress will

use the template file defined in the theme for displaying archive pages to serve content to the visitor.

The template used to serve a page request is selected by the installation according to the WordPress template hierarchy (Appendix 1). [18] WordPress will use this hierarchy to search for available templates within the theme that suit the request. If no appropriate and more “specific” templates are present in the theme, WordPress will default to using the `index.php` file at the end of the hierarchy to display the requested page.

2.2.5 Gutenberg

Until late 2018, the WordPress content editing experience had remained largely unchanged. In the past, content was edited in the admin area through a text editor embedded into post and page items. Following the release of WordPress version 5.0, the CMS began to ship with “Gutenberg”, a new block-based content editing system. Gutenberg lets users build pages and posts using a series of predefined “blocks, whose styles are defined by the theme in use. [22] Several core blocks are included in default WordPress installations, such as basic paragraph and heading blocks; as well as more complex query and archive blocks. The Gutenberg editor is showcased in Figure 7 below.

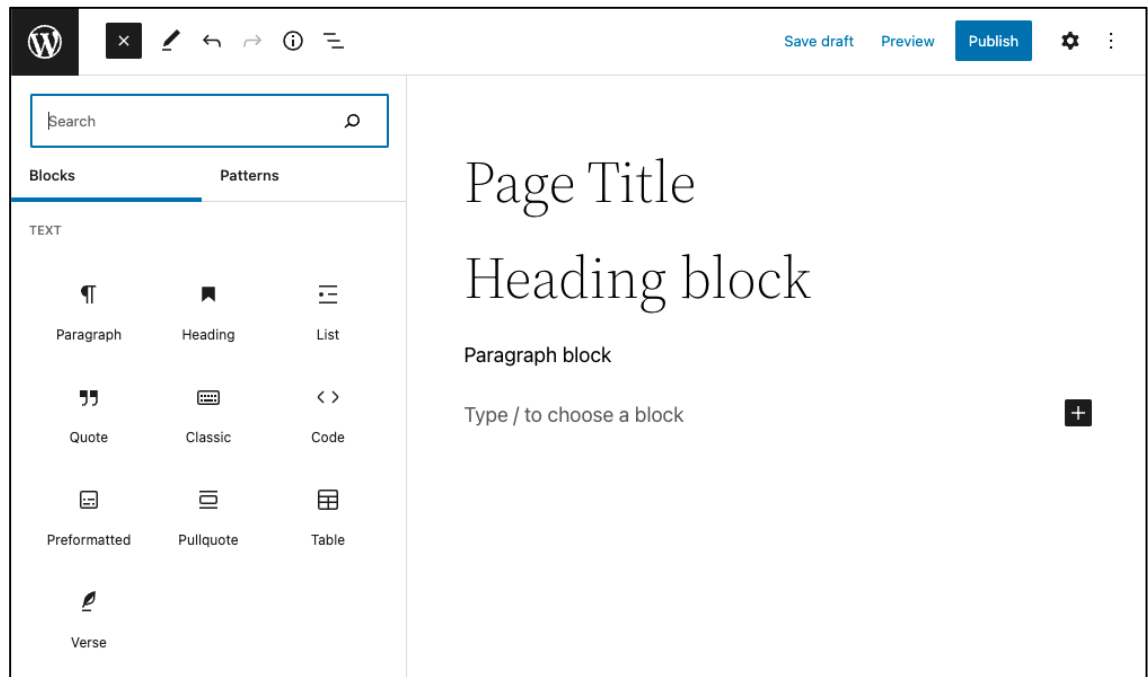


Figure 7. The Gutenberg editor.

Gutenberg is an ongoing, multi-phase project for the WordPress core development team. [23] The idea behind the project is to simplify the page-building process and enable a consistent full-site editing experience that leverages the use of modular building blocks. [24] Over the past few years, WordPress development has focused significantly on the Gutenberg editor and the creation these functional blocks.

2.2.6 Updates

Given the constantly evolving state of WordPress, the CMS is subject to regular updates, which may be set to run automatically. [9, p. 38] The changing nature of WordPress means that sites could require frequent upkeep, as major and minor updates can cause compatibility issues with themes or plugins active on the site. This is an important consideration to make when developing for WordPress, as methods should be employed to mediate the risk of site failure caused by updates. This includes practices such as limiting the number of

active plugins on your installation, disabling automatic updates and creating a development environment where changes can be tested. [25]

3 Project Scope and Requirements

Following a series of meetings with Liana’s Websites team leader and business director, the scope of the project was decided, and preliminary requirements were drafted. The proposed outcome of the project was to build a prototype WordPress site that mirrored the design and functions of LianaCMS’s “Showroom Template”, a pre-made template used to build enterprise websites using the company’s CMS. The homepage of the Showroom Template’s demo site is showcased in Figure 8 below.

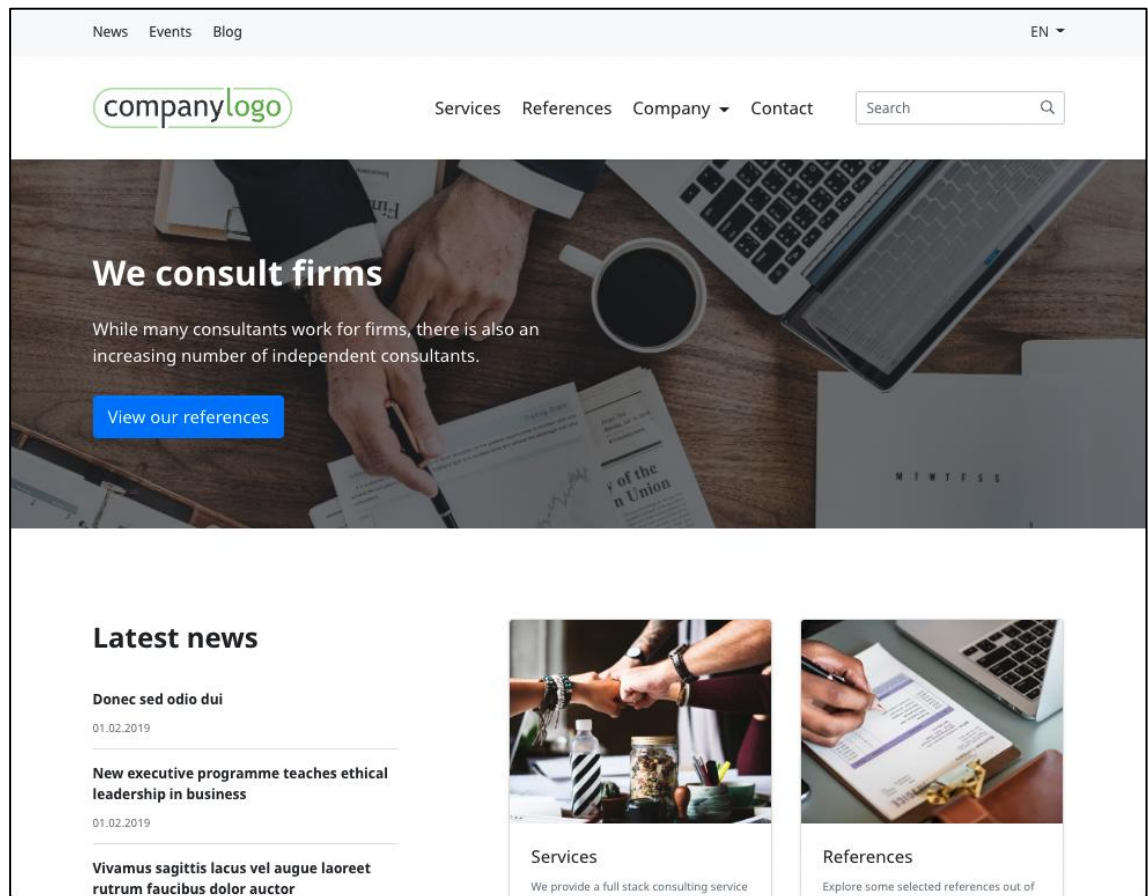


Figure 8. LianaCMS's Showroom Template

Building this prototype site would involve developing a custom WordPress theme and compiling a list of suitable plugins that fulfil the company's requirements for an enterprise-level website. The goal was to use this theme as a starting point for building custom WordPress sites at Liana, as well as to collect knowledge that would inform the Websites team's future practices and processes when using this platform.

The methods used to carry out the project were determined by the project requirements. These requirements can be broken down in terms of functionality, usability, maintainability, and learning.

3.1 Functionality

Functionality requirements represent the features and functions that need to be available for users on the prototype or demo WordPress site. In line with the Showroom Template, the goal was to implement or support the following:

- custom content types such news, events, contacts
- custom content elements to add to pages
- basic store / e-commerce features
- content translation, multilingual capabilities
- contact forms

3.2 Usability

Usability requirements had to be met in the front and back-end of the demo site, emphasizing the ease-of-use when it came to both browsing the site and adding content to it. Since the site's front-end was based on the Showroom Template, that meant it should inherit its layout and structure, as well as its responsiveness to different screen sizes. Though the sites design did not have to fully match the template's, it was important to recreate the following:

- header and footer areas
- menus (primary, secondary, footer, and mobile)
- parent / child page template structure
- essential content elements

In the back-end or admin area of WordPress, the process of adding content to the site needed to be user-friendly; carried out in a consistent and logical manner.

3.3 Maintainability

Maintainability refers to the ease with which the website can be maintained and updated to ensure compatibility with future versions of WordPress. In practice, this required:

- using current best WordPress development practices
- utilizing core features extensively
- keeping number of plugins to a minimum

3.4 Learning

The project also had to consider learning and training requirements. Following the project's conclusion, the Websites team would have to be trained on the project's findings and proposed methods. To facilitate this learning process and make use of the team's expertise, current LianaCMS development practices should be integrated as much as possible.

4 Current State Analysis

In early October 2021, the websites team at Liana received its first set of WordPress migration cases. A WordPress migration case involves moving a client's existing WordPress site over to the company's own servers, handing over hosting and maintenance responsibilities to Liana. The website is copied over exactly as is with the content and design intact, and updates to the site

may follow depending on the agreement made with the client. These migration cases offered an opportunity to observe how clients' WordPress installations were set up and to evaluate them alongside the project requirements.

Two migration clients' websites were investigated before starting the project, as they offered insight into two separate approaches to building multi-faceted WordPress sites. Nevertheless, both approaches shared similarities in plugin selection used to extend WordPress's core capabilities. Below is a summary of these findings and an evaluation of their strategies alongside external sources of information. To preserve anonymity, the names of the clients have been omitted.

4.1 Client A

Client A is a medium-sized business involved in selling electrical components. The client's WordPress site was built by an external agency, serving primarily to showcase the company's product offerings and provide detailed documentation related to them. The website also houses basic information about the company and the industries they operate in; as well as a blog-like news section that is updated regularly.

4.1.1 Elementor Pro

Page content and other areas of the site such as the header and footer have been created using Elementor Pro. Elementor Pro is the paid version of the Elementor WordPress plugin, which allows users to build sites using custom "widgets" inside a drag-and-drop editing interface. [26] This interface can be seen in Figure 9.



Figure 9. The Elementor page builder.

The site's visual style is defined by a custom theme active on the site, which is based on a default theme for Elementor Pro. Elementor Pro offers a minimally styled theme named "Hello Elementor", which may act as a starting point for customizing an Elementor Pro-based theme. [27] However, much of this customization was not done with code, as the plugin offers a method of creating templates and designing site areas using Elementor's drag-and-drop interface.

Elementor Pro can be likened to Gutenberg in that they both simplify the content addition process by employing the use of pre-made modular and reusable components. The level of customization achievable through Elementor Pro's editing interface is much higher than that of Gutenberg, as the plugin offers advanced controls to customize the style and functionality of every element on the site; allowing users to create complex pages.

4.1.2 WooCommerce

As Client A's business practice involves selling goods, a plugin called WooCommerce is installed on the site. WooCommerce extends WordPress by creating a framework for listing and selling products, which is done by

registering a “Products” post type on the site. WooCommerce includes functionality for managing product data such as price and variations; and automatically creates customizable user, account, cart, and shop pages. [28] In Client A’s site, WooCommerce’s features are only used to manage and display the products offered by the company, as the site is not intended to function as an online store.

4.1.3 Polylang

Client A’s site is translated into three different languages: Finnish, English and Estonian. This is done with the aid of Polylang, a plugin that enables multilingual features on WordPress sites. Polylang allows users to assign a default language to the installation and then register additional site languages. Posts, pages, and taxonomies are all translatable; and their translated versions are linked to one another. [29] In Client A’s site, all the pages were built in Finnish first and subsequently duplicated and translated into the remaining languages. In order to switch between language versions in the front-end, a dropdown element provided by Polylang was added to the site’s main navigation.

4.1.4 Advanced Custom Fields

Two custom post types have been manually registered on Client A’s WordPress installation, one of which is “Person”. This post type is used to create a roster of the company’s employees and to display their contact information on the site. This information is added to each Person post using a plugin called Advanced Custom Fields or ACF. ACF allows users to create custom meta boxes or “fields” which can be attached to several locations in the WordPress admin area; such as posts, pages, blocks, and menus. This enables the storing of custom data associated with these items in the installation’s database. [30] Figure 10 below displays an example of a custom field created with ACF.

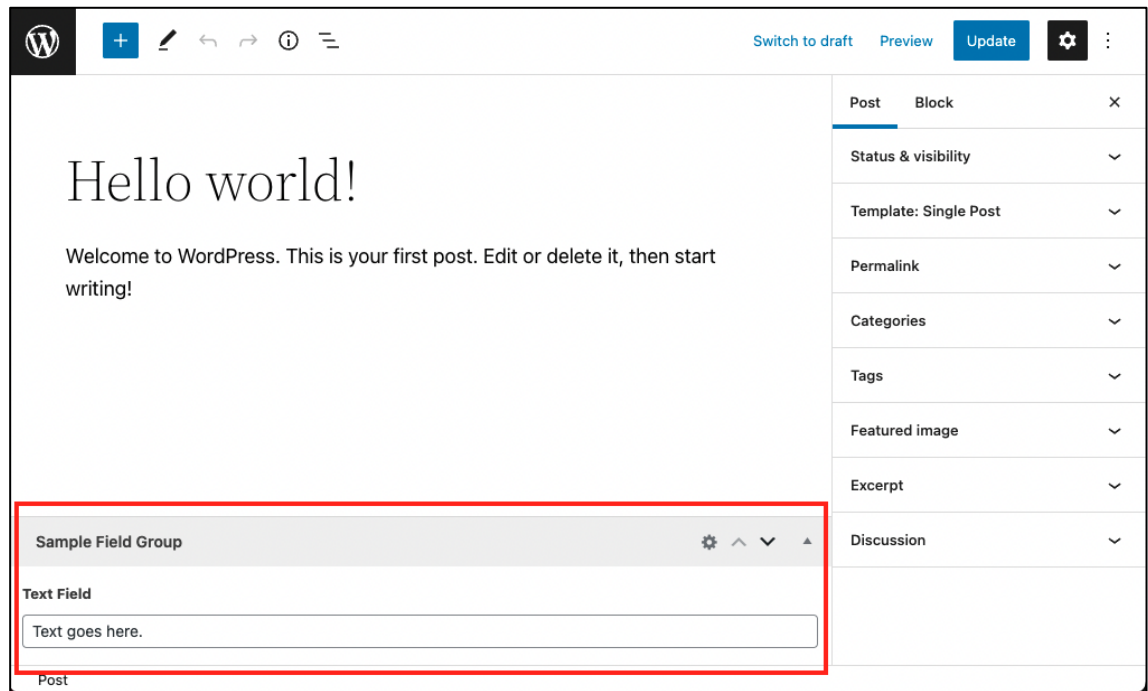


Figure 10. A text input field on a post item.

Multiple types of fields can be created with ACF; including image, text, and choice fields. As an example, Client A's WordPress installation uses ACF to add phone number and email input fields to each Person post corresponding to an employee. These values are then displayed under each employee's profile on the site's contact page. More complex fields such as relational and layout-manipulating fields are also available, though many require the paid Pro-version of the plugin to be installed. [30]

4.1.5 Contact Form 7

Several contact forms appear in different areas of Client A's site, which allow visitors to get in touch with the company regarding their products and services. While form building features are limited by default in WordPress, they can be created with plugins such as Contact Form 7, which is used on this site. [31] Although Elementor Pro has its own form widget available for building forms, all the forms on Client A's site have been set up with Contact Form 7. These forms were created to fulfil different functions, and each one is available in all three

site languages. In the admin area, forms are added to the different pages using shortcodes, which are code snippets generated by the plugin for each form. [32]

4.2 Client B

Client B is a medium-sized insurance brokerage firm which offers insurance-related consultation and services. Their WordPress site, which was also built by an external agency, highlights the company's services and gathers information on several types of insurance. The website includes a basic blog section, as well as pages for viewing company references and contact information.

4.2.1 Custom Theme and ACF Pro

Client B's site uses a custom theme that was developed from scratch, as opposed to building atop another ready-made theme. Content is added to the site's pages using a combination of Gutenberg and Advanced Custom Fields Pro; where basic text content is added inside the Gutenberg editor and custom elements are added with ACF Pro, beneath the editor. The site employs frequent use of the ACF Flexible Content field, which allows users to add content to pages from a list of predefined layouts. Each layout is made up of several fields and represents a section of content, such as a text element or a call-to-action button. [33] These layouts were configured by the theme developers, and template files for each layout are defined within the theme. This method proposes a solution for adding custom elements to pages separately from Gutenberg.

Another Advanced Custom Fields Pro feature being used on Client B's site is custom "options" pages. Options pages are standalone pages that appear on the navigation sidebar in the WordPress admin area, which are often used to manage site settings and other general functionality. ACF Pro adds the ability to register these pages inside the theme and add groups of fields to them. [34] In this case, an options page called "Theme settings" was created where users manage global theme preferences.

Lastly, ACF Pro is also used to add meta information to custom post types. Client B has one custom “Specialist” post type registered, which collects the name and contact information of the company’s brokers. The fields added to Specialist posts include email, phone number, and role; as well as a Boolean field which can be used to hide a broker’s information from the site’s search results.

4.2.2 Active Plugins

Similar to Client A, Client B uses Polylang to translate their website and Contact Form 7 to create forms. There are a total of 30 active plugins on the site, some of which are used extensively and others which perform smaller functions such as adding a Google Maps integration. The site’s header and menu navigation system were built using a third-party plugin called Max Mega Menu, which provides a visual interface for building and customizing menus. [35] Although WordPress includes a default system for creating menus, the appearance and structure must be defined inside the theme. Plugins such as Max Mega Menu aid in offloading some of the work required to set up a theme from scratch.

4.3 Comparing Strategies

Both clients’ sites illustrated some of the methods and tools being used to build contemporary WordPress websites. As Liana began handling more migration cases, a trend in theme-building practices and plugin preferences became evident. In order to propose a method for the company to commence building enterprise-level WordPress sites, these approaches had to be evaluated according to the requirements defined for the project in Section 3.

4.3.1 Page Builders

When building a custom theme for WordPress, it is important to consider which page-building features should be available for the end users. This means deciding on which editor to incorporate into the content management

experience. Currently, the popular options are to build native WordPress themes that leverage the Gutenberg editor, or to build themes for use with Elementor Pro.

Elementor Pro's primary appeal is the degree of page customization attainable through the plugin's drag and drop interface, with little-to-no coding knowledge required to set up. This gives users more control over their WordPress site, though makes it difficult for precise customization to be achieved by theme developers; as control over element functionality and structure is designated to Elementor Pro. In this regard, an Elementor Pro approach may suit cases where users have the flexibility to build their own website, but not an enterprise-level site with set requirements and design.

Creating a native WordPress theme hands developers more control over the site's appearance, while still offering the flexibility for users to arrange content using the Gutenberg editor or a plugin such as ACF Pro. As Gutenberg is part of core WordPress, longevity and long-term compatibility with future iterations of the platform can be expected. Meanwhile, reliance on third-party plugins to establish essential functionality may put a site at risk of deprecation [36]. Nevertheless, plugins are a major convenience, so it is important to select them carefully. The metrics that often indicate the reliability of a plugin are the reviews, number of active installations, and the date the plugin was last updated. [9, p. 251]

4.3.2 Leveraging ACF Pro

Using Advanced Custom Fields with custom post types proves to be an effective method of storing data associated with different types of content on a WordPress site. Nearly all migration cases received by Liana have had ACF active on their websites and used to varying degrees. In the plugin directory, ACF has over 2 million active installations, is consistently updated, and has largely positive reviews which support its reliability. [37] The Pro version of the plugin can be leveraged to build complex custom elements, custom blocks, and

options pages; which open up several possibilities for tailoring the content management process. Additionally, the concept of adding custom fields in WordPress is reminiscent of properties in LianaCMS, which builders are accustomed to working with.

The ability to create custom Gutenberg blocks with ACF Pro was introduced in version 5.8 of the plugin, released in mid-2019. This feature offers a simplified framework for registering blocks that support ACF field groups. [38] Although developing native Gutenberg blocks is possible, the process is relatively complex, and requires a solid understanding of the React JavaScript library. [39] For developers accustomed to working with PHP and ACF, custom ACF blocks are a convenient alternative to building a Gutenberg-ready theme.

4.3.3 Additional Plugins

In terms of adding multilingual capabilities to a WordPress site, Polylang has also seen wide adoption, with favorable reviews and numerous active installations [29]. The translation process with Polylang works similarly to that of LianaCMS, where pages built in one language are duplicated and then manually translated. An alternative to LianaCMS's translations database page is also available, where custom text strings added to a theme are available for translation. This makes Polylang a convenient plugin to include in the project, though the Pro version is required to enable compatibility with additional plugins such as WooCommerce. [29]

For creating simple online stores in WordPress, WooCommerce provides a straightforward solution with minimal setup required to support it in a custom theme. Though the plugin allows for in-depth customization of the ecommerce experience, this is outside the scope of the current project. As for contact forms, although a variety of form plugins were discovered in later migration cases, Liana has decided to focus its efforts on adopting the use of Contact Form 7. In the future, the company plans to integrate this plugin with their own marketing automation and mailer services.

5 Methods

The thesis project was carried out between late autumn of 2021 and early spring of 2022. In this time, a custom theme was developed from scratch for WordPress, and a demo site was set up on the company's servers. Outlined below are the methods used to approach the project and fulfil its requirements.

5.1 Tools

Most of the tools used to develop the custom theme were selected in order to mirror current site-building practices at Liana. These tools will allow the Websites team to begin exploring theme development, and the aim is to refine the selection as builders collect hands-on experience working with WordPress sites.

5.1.1 Local WP

Theme development was carried out in a local environment, where a WordPress installation was set up using Local. Local is an application that simplifies the process for creating local WordPress sites and provides easy access to the installation's file structure and database. [40] Based on the specifications of WordPress sites currently running on the company's servers, a WordPress installation was set up using PHP version 8, nginx 1.16.0 for the web server, and MySQL 8.0.16 for the database.

5.1.2 VScode

Visual Studio Code was the code editor of choice for the project, as it is the preferred editor for builders at Liana. The Websites team also has a list of extensions and configurations set up for use with VScode, which improve workflows and ensure a unified coding standard in the team. Inside VScode, the PHP Intelephense extension was an essential inclusion, as it allowed the editor

to recognize and check the syntax for core WordPress functions being used in the code.

5.1.3 Node.js and npm

As with LianaCMS development, Node.js and npm were used to manage the project's packages and dependencies. This allows for incorporating external software packages to handle processes such as compiling and compressing the theme's scripts and styles. The project was set up using Node version 14.19.1 and an `.nvmrc` file was included in the theme directory, which makes it easier to install the corresponding Node version on any machine using Node Version Manager.

5.2 Theme Repository

The theme's code repository contains all the files required to run and develop the theme. The repo uses Git for version control and is hosted on GitLab, from which it can be cloned to develop in a local environment. Cloning the theme repository inside a WordPress installation's "themes" directory will allow the theme to be recognized and activated in the admin area.

The theme is primarily composed of SCSS, JS and PHP files; where the SCSS and JS files determine the theme's appearance and front-end behavior, and the PHP files dictate most of the back-end functionality. In line with WordPress's template hierarchy, several PHP templates are included that define the HTML markup and structure of the different site pages.

5.2.1 Wireframe

The process for building a website from scratch in LianaCMS involves forking a "wireframe" repository, which has the initial code file structure and minimal configuration defined for getting started with development. The structure of the theme was largely based on this wireframe and tailored to meet WordPress's

own requirements. A screenshot of the theme's file structure is shown on the left-hand side of Figure 11 below.

The screenshot shows a VS Code editor with two panels. The left panel displays the file structure of a theme repository named 'LIANATECH-THEME'. The right panel shows the code for 'gulpfile.js'.

```

LIANATECH-THEME
├── acf-json
├── assets
├── dist
├── inc
├── languages
├── page-templates
├── src
├── template-parts
├── vendor
├── woocommerce
├── .gitignore
├── .nvmrc
├── 404.php
├── archive.php
├── comments.php
├── footer.php
├── functions.php
├── gulpfile.js
├── header.php
├── index.php
├── package-lock.json
├── package.json
├── page.php
├── README.md
├── screenshot.png
├── search.php
├── single-lianatech_contact.php
├── single-lianatech_event.php
├── single-lianatech_news.php
├── single-post.php
├── style.css
└── woocommerce.php

```

```

gulpfile.js > ...
94     } ) )
95     .pipe(dest(`./languages/${config.domain}.pot`));
96   }
97
98   /**
99    * Clean
100   */
101
102   function clean() {
103     return del(`${config.dist}/**/*`);
104   }
105
106   /**
107    * Watch
108   */
109
110   function watchFiles() {
111     global.isWatching = true;
112     watch(`${config.inputDir}/**/*.scss`, series(css));
113     watch(`${config.inputDir}/**/*.js`, series(js));
114   }
115
116   /**
117    * Exports
118   */
119
120   // Can be run with the gulp command e.g. 'gulp sass'
121   exports.sass = sass;
122   exports.js = js;
123   exports.palette = create_color_palette_json;
124   exports.pot = generate_pot_file;
125
126   // Run with 'gulp'
127   exports.default = series(
128     clean,
129     parallel(css, js),
130     parallel(watchFiles),
131   );
132

```

Figure 11. The theme repository open on VScode.

The file structure organizes templates, scripts, and styles into separate directories. To improve legibility and coherence, files are split based on their function into separate parts and components.

Once the theme repository is cloned and the dependencies are installed, a set of development scripts or tasks can be run via the command line. These tasks, which aim to replicate those used with LianaCMS's wireframe, are defined in a Gulp file inside the theme directory. This file is showcased on the right-side column in Figure 11.

5.2.2 Bootstrap 5

The inclusion of Bootstrap 5 in the custom theme is an update to current site-building practices within Liana's Websites team. Bootstrap is a mobile-first CSS framework that facilitates the development of modern and responsive sites [41]. The fifth major release of Bootstrap was made public in mid-2021, though builders in the Websites team have so far been working on projects using Bootstrap 4. Bootstrap 5 supplements its predecessor with numerous features, including support for RTL or right-to-left text, which would benefit projects for Liana's clients in the middle east.

5.2.3 WP Pusher

Although the custom theme was developed locally, a WordPress installation sporting the theme was set up on the company's servers for demonstration purposes. The process of adding the theme to the live site was facilitated by a plugin named WP Pusher. WP Pusher integrates with GitLab and allows for connecting a remote theme repository to a WordPress installation. [42] This allowed any changes made to the theme code base to be deployed automatically to the live demo site, where the theme was active.

5.3 Required Plugins

In line with keeping plugin requirements to a minimum, the custom theme requires only two plugins to be active on a WordPress installation for it to function properly: Advanced Custom Fields Pro and Font Awesome 4.

5.3.1 Advanced Custom Fields Pro

ACF Pro and its functions are used extensively within the theme; notably for adding fields to custom post types, creating custom Gutenberg blocks, and adding controls for customizing page layouts. Figure 12 below showcases the registration of blocks using ACF functions inside the theme repository.


```

1 <?php
2
3 /**
4  * Create custom Gutenberg blocks with ACF Pro
5  *
6  * @link https://www.advancedcustomfields.com/resources/blocks/
7  *
8  * @package Lianatech
9  */
10
11 add_action('acf/init', 'lianatech_register_block_types');
12
13 if (!function_exists('lianatech_register_block_types')) {
14
15     function lianatech_register_block_types()
16     {
17         if (function_exists('acf_register_block_type')) {
18
19             /**
20              * Templates created for the blocks are in 'template-parts/block-templates'
21              *
22              * Learn more about the 'register block' function parameters here:
23              * https://www.advancedcustomfields.com/resources/acf_register_block_type/
24              */
25
26             // Promotional Banner
27             acf_register_block_type(array(
28                 'name' => 'promotional-banner',
29                 'title' => __('Promotional Banner', 'lianatech'),
30                 'description' => __('A custom promotional banner block.', 'lianatech'),
31                 'render_template' => 'template-parts/block-templates/promotional-banner.php',
32                 'category' => 'lianatech-block-category',
33                 'icon' => 'megaphone',
34                 'keywords' => array('promotional', 'banner'),
35                 'supports' => array(
36                     'jsx' => true,
37                 ),
38             ));
39
40             // Contact Cards
41             acf_register_block_type(array(

```

Figure 12. The custom-blocks.php file for registering ACF blocks.

ACF field groups are added to version control and stored in the theme repository with the aid of an ACF Pro feature called Local JSON. Local JSON creates a folder in the root of the theme directory where field configurations are stored in JSON format. This allows ACF to synchronize and use the latest configurations defined by the theme. [43]

5.3.2 Font Awesome 4

Font Awesome is an icon library which facilitates the implementation of graphic icons on a website. [44] Font Awesome is used frequently in LianaCMS projects for purposes such as adding social media links with corresponding brand icons to a site's footer. The WordPress plugin version of this library, called Font Awesome 4, must be activated in order to render the icons that have been added to different parts of the theme.

6 Results

The result of the project is Liana’s first custom theme built for WordPress called “Lianatech”. The theme functions as a ready-made template solution for the company to use with enterprise-level WordPress cases. It integrates LianaCMS content types such as Contacts, News, and Events into WordPress; and includes support for setting up a simple online store, building contact forms, and creating multilingual sites via a set of recommended WordPress plugins. A demo site showcasing the theme was created, the home page of which is displayed in Figure 13.

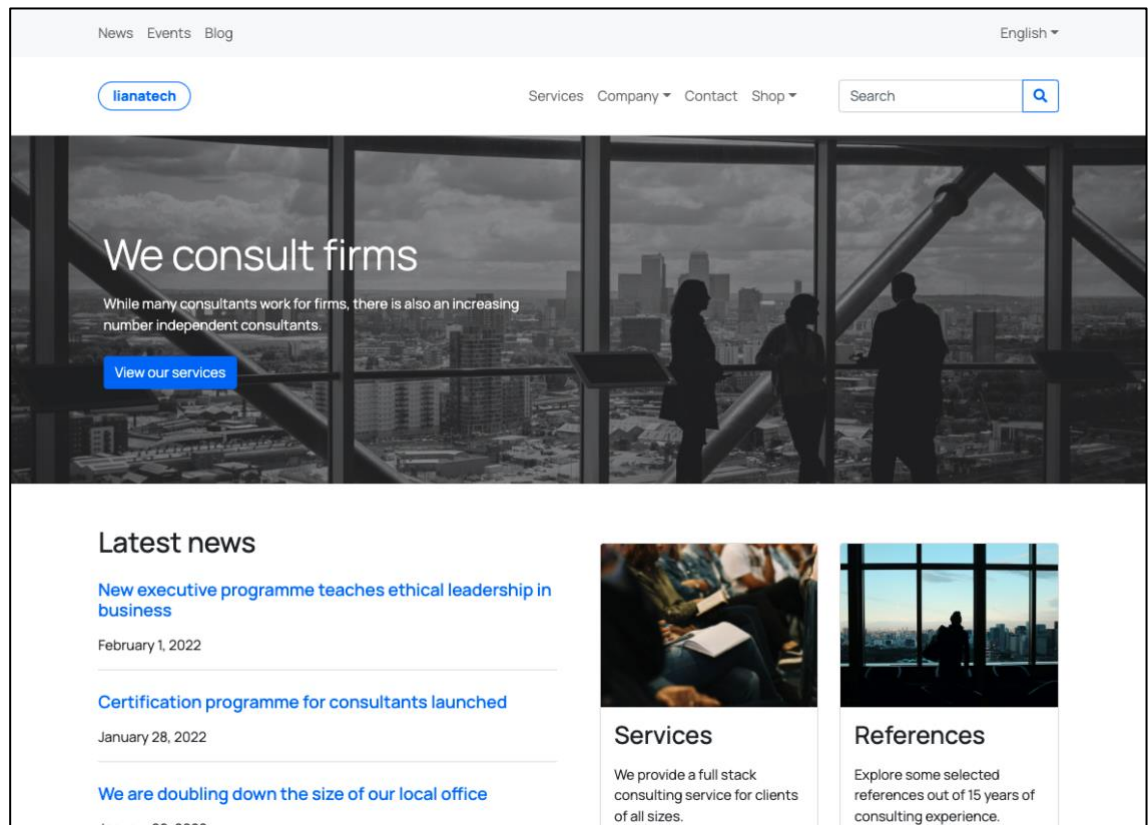


Figure 13. The homepage of the demo site using the Lianatech theme.

Lianatech’s design is contemporary yet plain, which means it can easily be tailored to suit the visual requirements of specific cases. The theme is responsive and has been optimized for browsing site content on desktop and mobile devices.

6.1 Contacts, News, and Events

The theme registers three custom post types to a WordPress installation: Contacts, News, and Events. Similarly, custom category taxonomies are registered and linked to News and Events, in order to create categories that are unique to these post types. Figure 14 below shows a list of custom event posts created in the admin area of the demo site and their associated categories. Each new content type has added its own menu tab to the admin area sidebar.

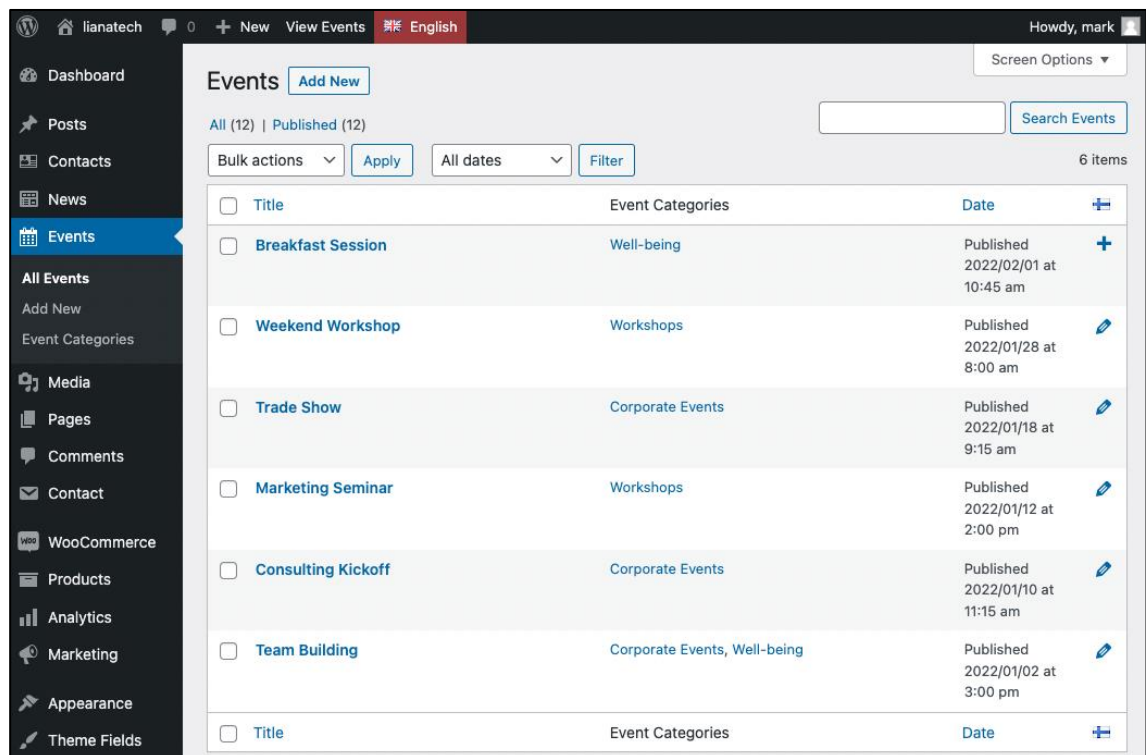


Figure 14. Custom events and their categories.

Where necessary, fields are attached to these custom post types using Advanced Custom Fields Pro in order to store associated data. For instance, each event post has a field group where users may input details such as event timing and location. This field group is showcased below in Figure 15.

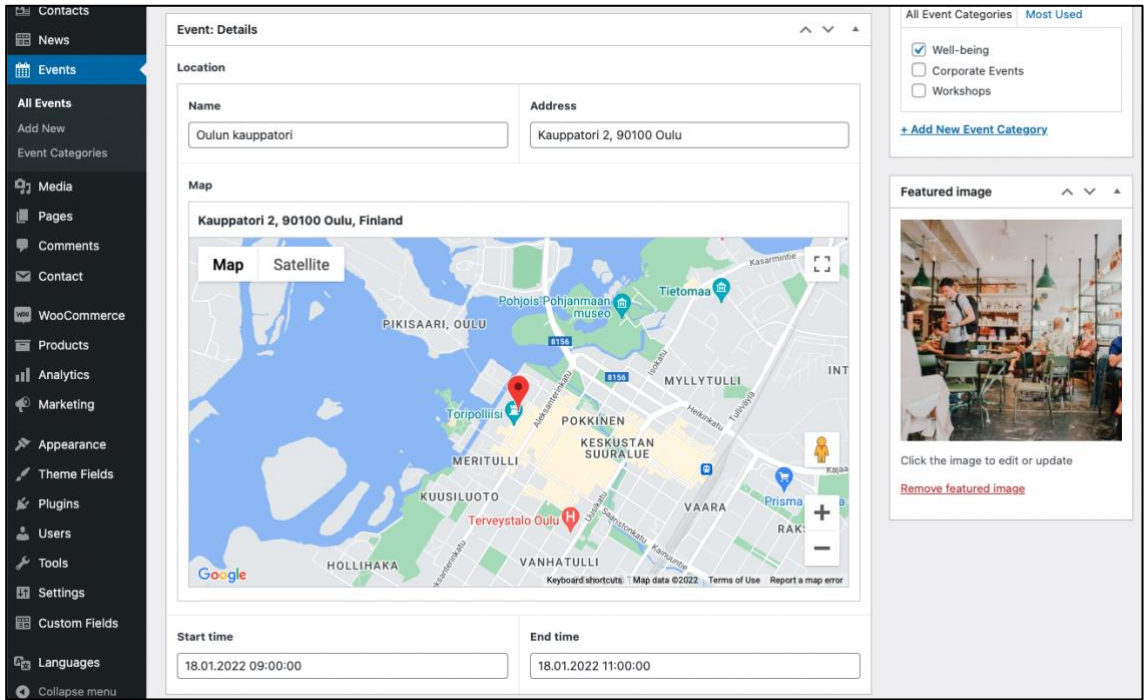


Figure 15. Location and timing fields for an event post.

The page corresponding to the event in the figure above is shown in Figure 16.

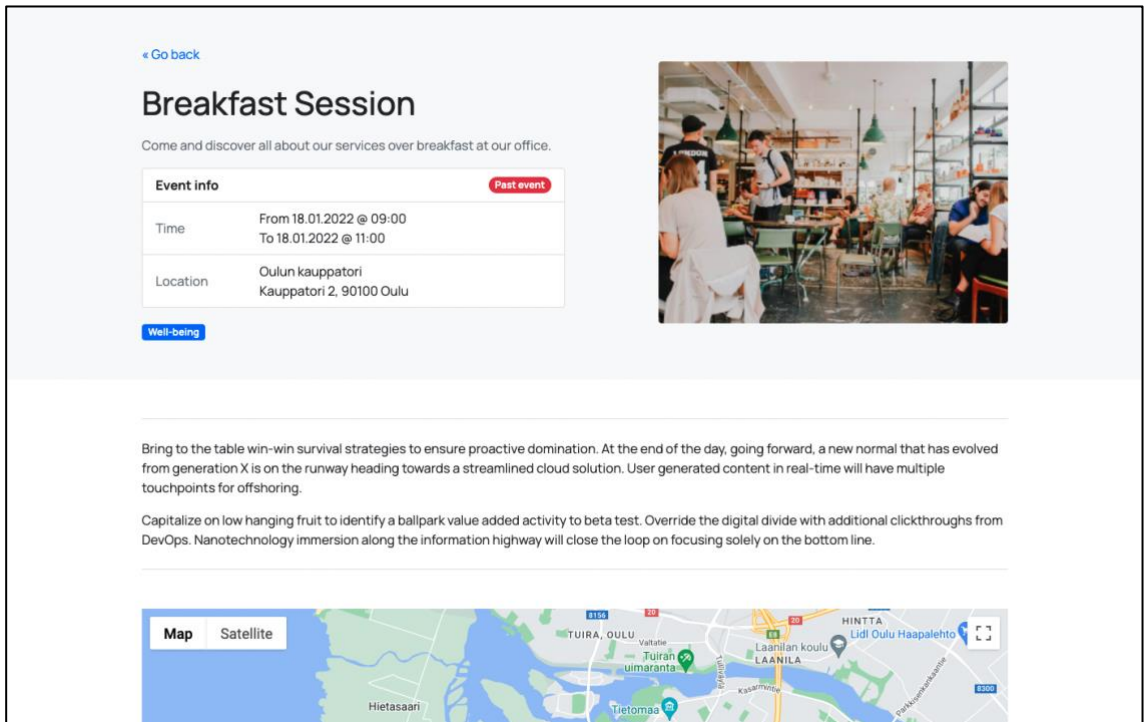


Figure 16. A single event page displaying event details.

The template for displaying each custom post type has been defined within the theme.

6.2 Lianatech Blocks

The theme is Gutenberg-ready, meaning that content can be added to pages through the Gutenberg editor using a combination of native and custom blocks created specifically for the theme. Figure 17 displays the list of custom Gutenberg blocks available for use, as seen in the demo site.

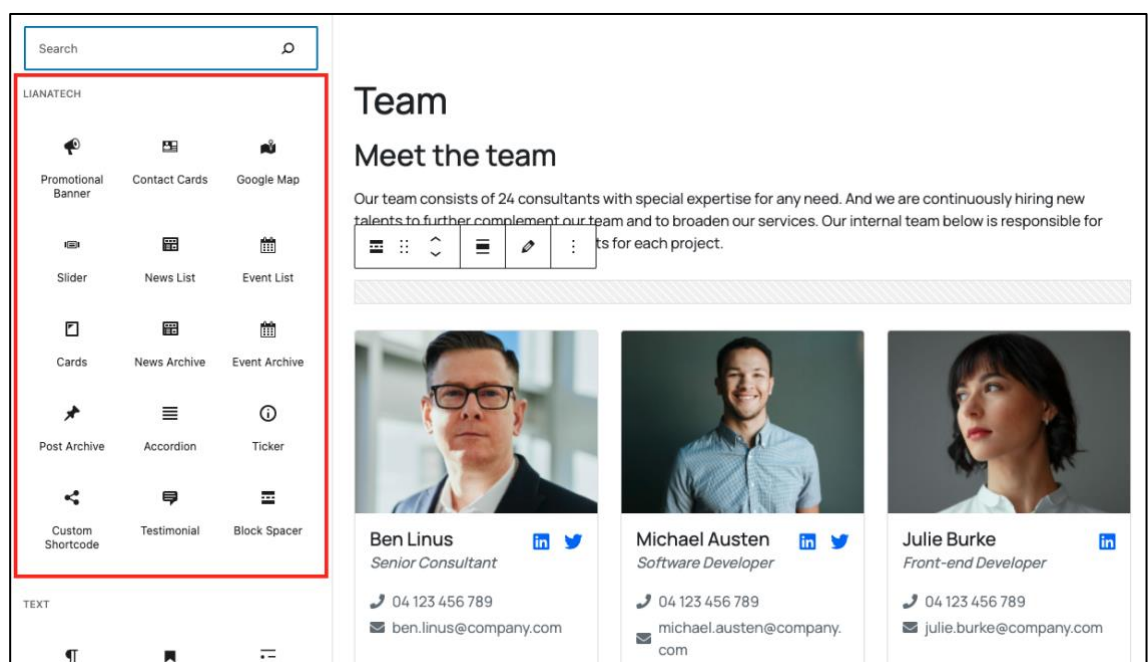


Figure 17. Custom Gutenberg blocks made for Lianatech.

Every custom block was registered with ACF Pro and has custom fields associated with it, where users can edit the block's settings and content. Each block has its own template within the theme that defines the block's structure and design, as well as its operating logic. As an example, the "Event Archive" block and its fields are highlighted in Figure 18.

The screenshot shows the Gutenberg editor interface. At the top left, there's a 'Happening Now' section with a calendar icon and a toolbar. Below it, two event cards are displayed:

- Marketing Seminar:** Come together to learn more about various marketing topics. Dates: 01.01.2022 - 31.12.2025. Location: Company Headquarters.
- Trade Show:** We're showcasing our latest products and services at the trade show! Let's come together to discuss and engage in some healthy competition over the course of a few days. Dates: 01.02.2022 - 01.02.2025. Location: Messukeskus.

The right sidebar shows the settings for the 'Event Archive' block, which is highlighted with a red border. The settings include:

- Event Archive:** Display an archive of events with pagination.
- Categories:** A 'Select' dropdown menu.
- Filter by event category:** A label for the categories filter.
- Time span:** Radio buttons for 'Past', 'Present' (selected), and 'Future'.
- Filter by timing of the event:** A label for the time span filter.
- Posts per page:** A text input field with the value '2'.
- Order:** A dropdown menu set to 'Ascending'.
- Block Styles:** A dropdown menu.
- Advanced:** A dropdown menu.

Figure 18. The Event Archive block and its associated fields.

Any changes made to the block's fields are instantly rendered within the Gutenberg editor, allowing for a convenient live preview of the block's output.

6.3 Page Settings

The Lianatech theme also adds fields to the page editor sidebar which can be used to customize the layout of pages. As can be seen in Figure 19 below, these fields allow users to utilize the page's featured image and excerpt to create a hero element on the page.

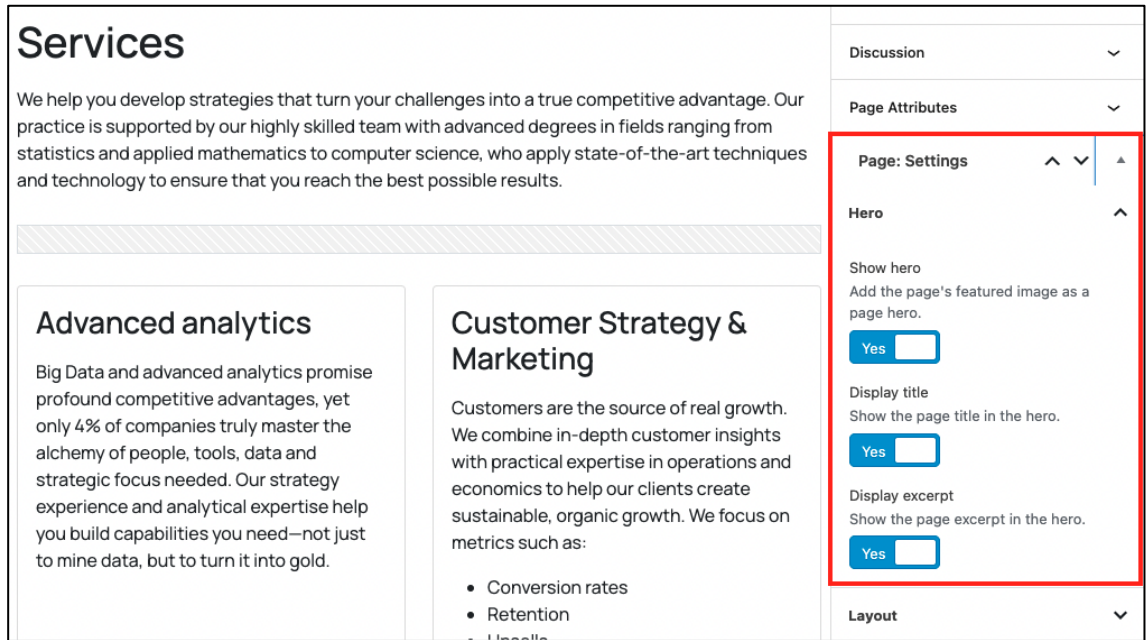


Figure 19. The "Page Settings" field group attached to every page.

Meanwhile, Figure 20 showcases an example of such a page in the front-end.

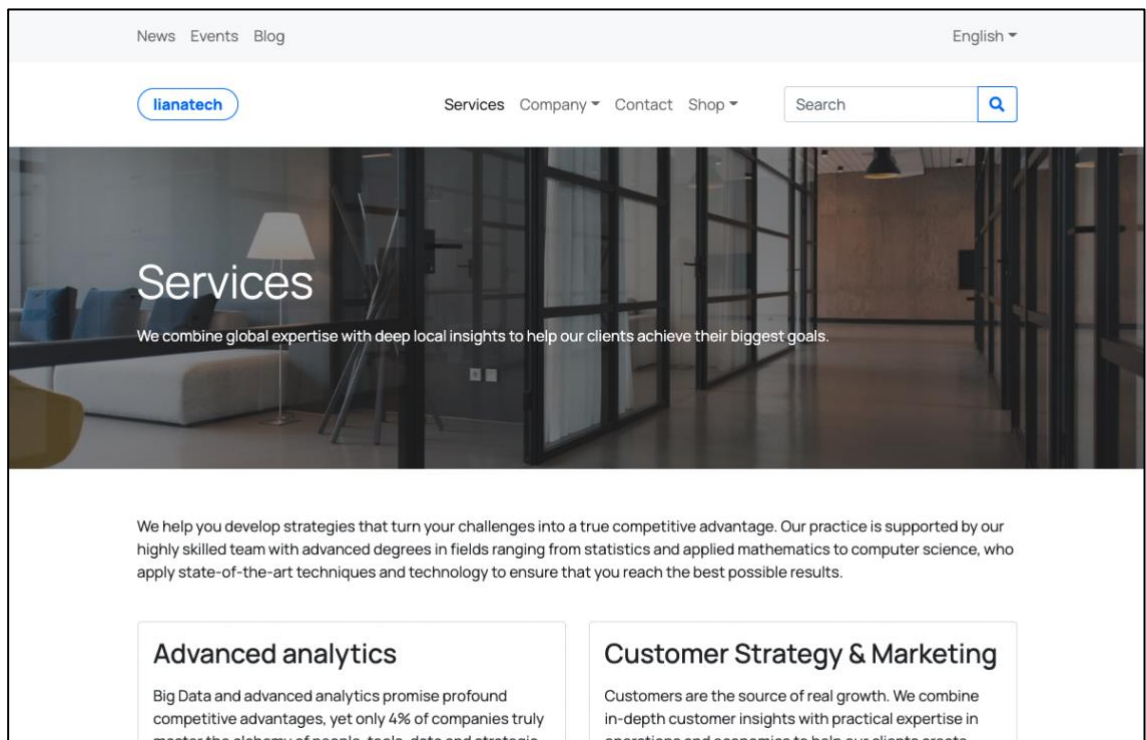


Figure 20. A page with a dynamically-created hero element.

Each page is rendered using one template file within the theme, which conditionally creates the page structure based on the settings enabled on each individual page.

6.4 Menus and Global Elements

The theme registers four menu locations to a WordPress installation: primary, secondary, footer and mobile. In the WordPress backend, menus can be created and attached to these locations. Figure 21 illustrates what the demo site's footer menu looks like in the back-end.

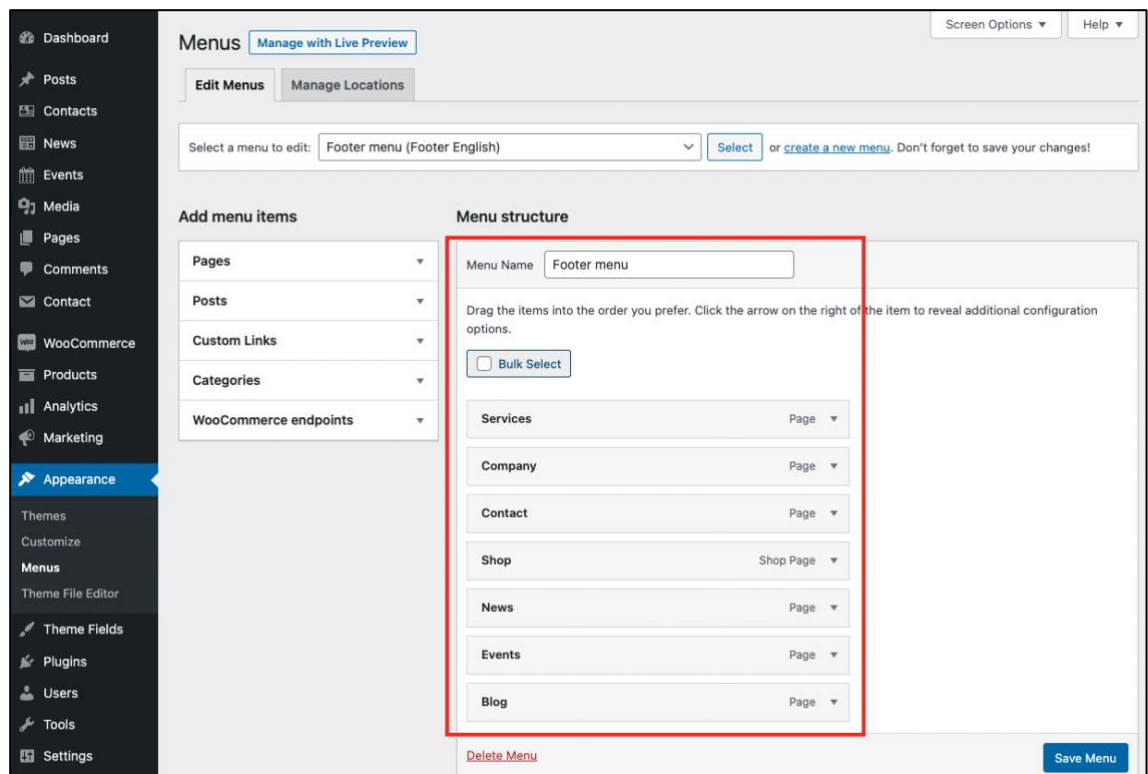


Figure 21. The footer menu in the WordPress admin area.

Meanwhile, the front-end view of the footer menu is displayed on the top-right side of Figure 22 below.

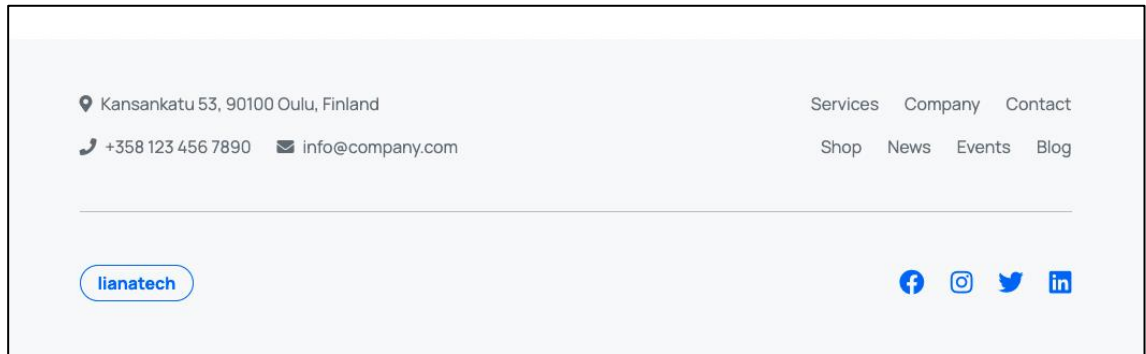


Figure 22. The Lianatech theme footer.

As seen in the figure above, the theme’s footer also includes a set of “global elements” such as contact information and social media links. To add global recurring site elements, a set of custom options pages have been registered inside the theme with ACF field groups attached to them. Figure 23 below displays an options page named “Global Elements”, accessible through the WordPress admin area.

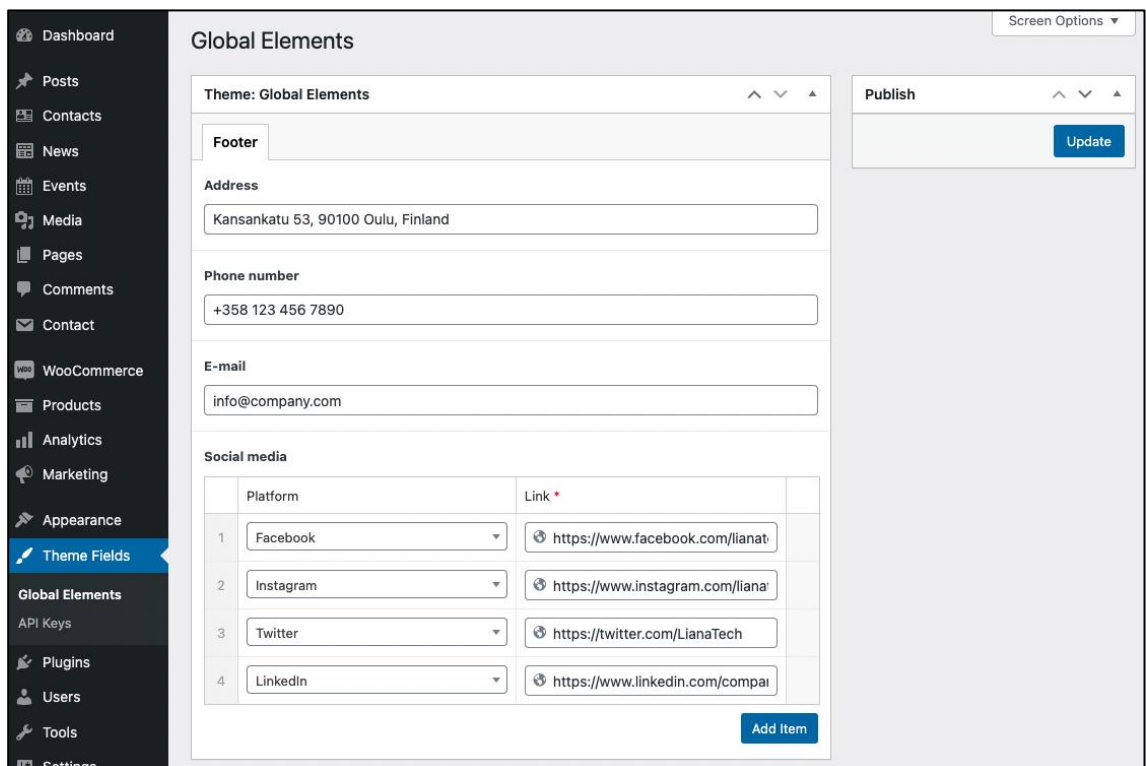


Figure 23. The "Global Elements" option page.

While it currently only serves to modify the theme's footer, the Global Elements page could be expanded to include customization settings for other site areas as well.

6.5 Shop, Contact Forms, and Language Translations

Lianatech is compatible with a basic installation of WooCommerce, form creation with Contact Form 7, and language translation with Polylang. The theme provides basic style overrides for these plugins, so in cases where the plugins generate content on the site's front-end, its style will be consistent with the rest of the theme. These three plugins are active on the theme's demo site, showcasing the functionality they add to WordPress. For WooCommerce, a shop page with a selection of sample products has been set up, which is displayed in Figure 24.

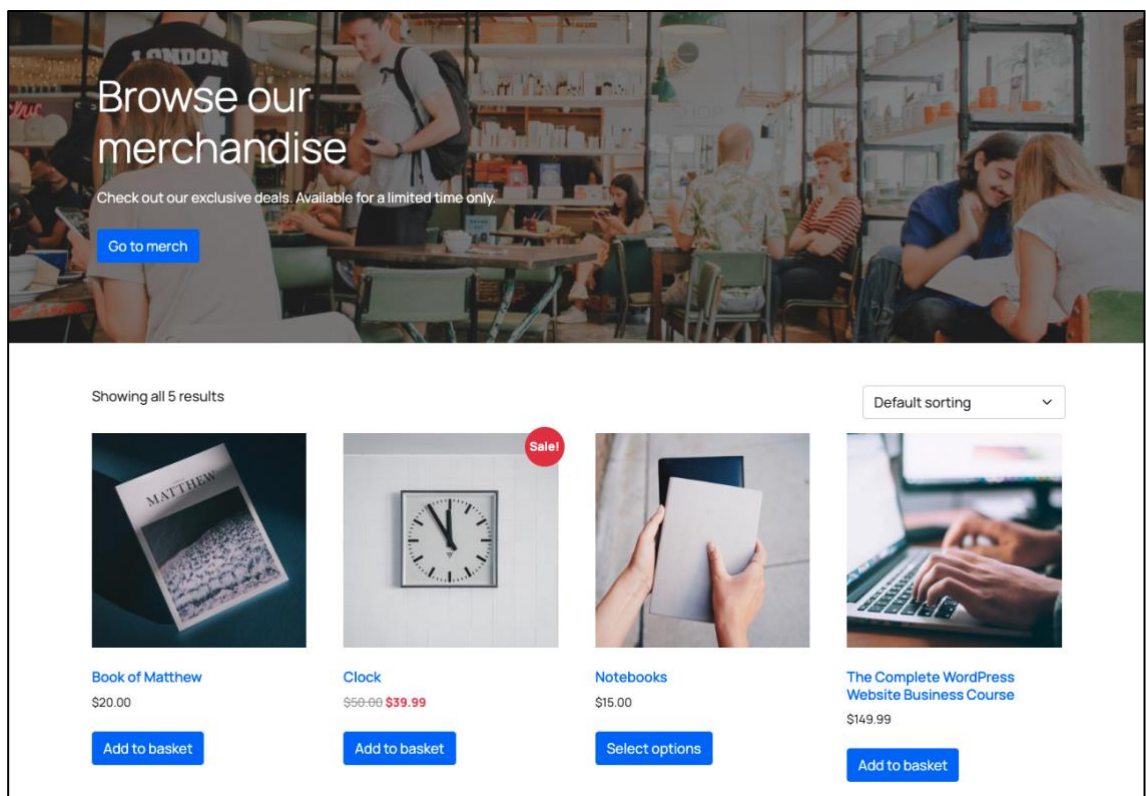
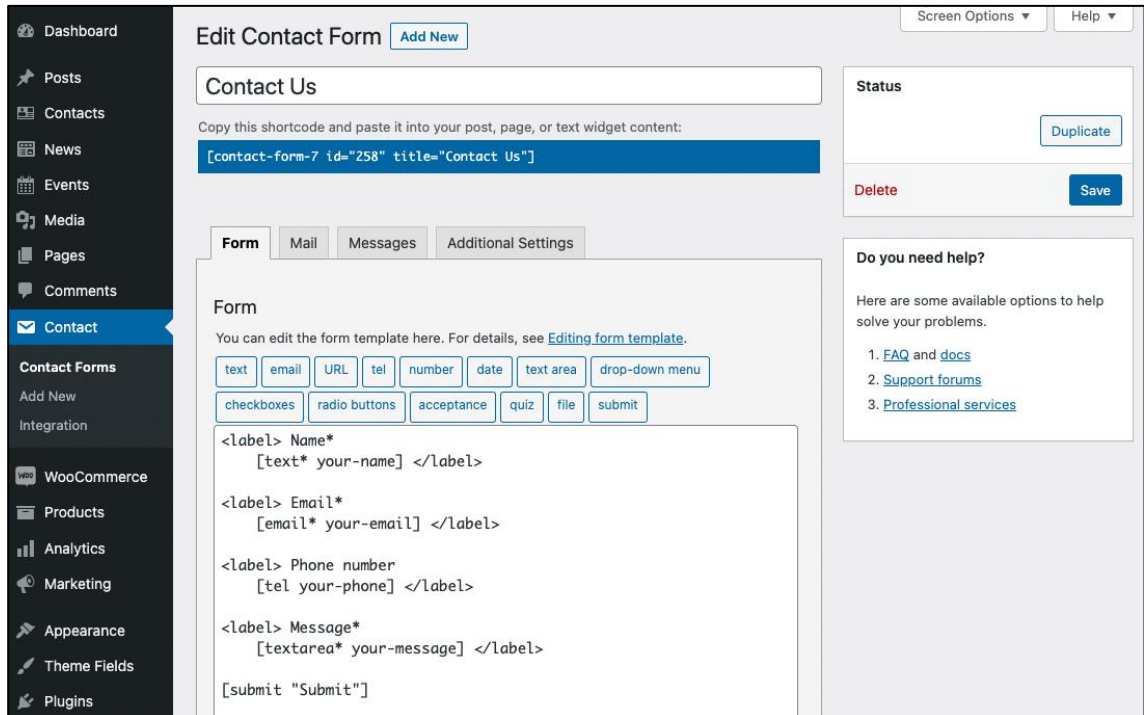


Figure 24. A basic shop page on the demo site.

For contact forms, two forms have been included in the demo site: a newsletter subscription form on the homepage, and a “contact us” form on the site’s contact page. These were created in the admin area and added to pages with shortcodes, making use of the theme’s own “Custom Shortcode” block. Figure 25 illustrates how the contact page’s form looks in the back-end.



The screenshot displays the WordPress admin interface for editing a contact form. On the left is a dark sidebar menu with options like Dashboard, Posts, Contacts, News, Events, Media, Pages, Comments, Contact, Contact Forms, Add New, Integration, WooCommerce, Products, Analytics, Marketing, Appearance, Theme Fields, and Plugins. The main content area is titled 'Edit Contact Form' with an 'Add New' button. Below the title is a text input field containing 'Contact Us'. A message says 'Copy this shortcode and paste it into your post, page, or text widget content:' followed by a blue box containing the shortcode `[contact-form-7 id="258" title="Contact Us"]`. There are tabs for 'Form', 'Mail', 'Messages', and 'Additional Settings', with 'Form' selected. The 'Form' tab shows a list of field types: text, email, URL, tel, number, date, text area, drop-down menu, checkboxes, radio buttons, acceptance, quiz, file, and submit. Below this is a code editor showing the form template:

```
<label> Name*  
[text* your-name] </label>  
  
<label> Email*  
[email* your-email] </label>  
  
<label> Phone number  
[tel your-phone] </label>  
  
<label> Message*  
[textarea* your-message] </label>  
  
[submit "Submit"]
```

 On the right side, there is a 'Status' section with a 'Duplicate' button, a 'Delete' button, and a 'Save' button. Below that is a 'Do you need help?' section with a paragraph and three links: '1. FAQ and docs', '2. Support forums', and '3. Professional services'.

Figure 25. A contact form built with Contact Form 7.

The resulting form on the demo site’s contact page is visible in Figure 26 below.

Oulu HQ
Kansankatu 53
90100 Oulu
Finland
[+358 123 456 7890](tel:+3581234567890)
info@company.com

[Facebook](#) [Instagram](#) [Twitter](#) [LinkedIn](#)

Contact Us

Name*

Email*

Phone number

Message*

Figure 26. A Contact Form 7 form on the site's contact page.

Lastly, a Finnish language version of the site was set up with Polylang to demonstrate the plugin's multilingual features. All the site's pages and post types were duplicated inside WordPress with Polylang and translated into Finnish. Figure 27 below showcases the Finnish-language version of the demo site's homepage.

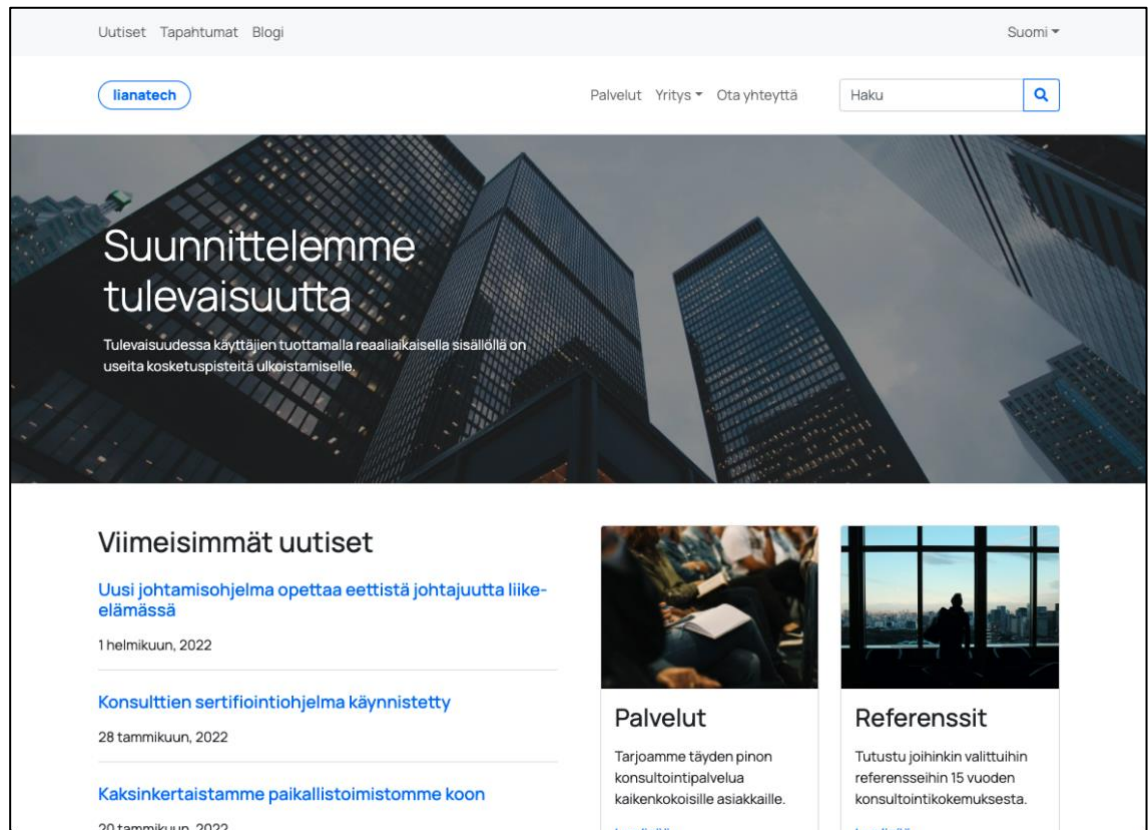


Figure 27. The Finnish-language homepage of the demo site.

To facilitate switching between language versions in the front-end, Polylang's own language switching element was added to the top right corner of the page. Custom styles were added to the switcher to maintain consistency with the theme.

7 Conclusion

This thesis has presented a solution for Liana Technologies to undertake enterprise-level website builds using WordPress. Based on an examination of the current state of WordPress development processes and practices, the work has resulted in a custom theme developed for the case company, as well as a proposed set of plugins that may be incorporated to achieve the functionality available in LianaCMS. It is recommended that the case company review these findings and begin testing implementations of the proposed solutions, ensuring that the most common use cases of enterprise-level sites are addressed.

As there are several approaches to building custom sites with WordPress, this thesis has aimed to provide a structure for builders in the Websites team to learn about the CMS and familiarize themselves with up-to-date practices. Given WordPress's constantly evolving nature, it is important to continuously evaluate the project's results in order to accommodate changes and improvements made to the platform. In the future, it may be worthwhile to consider building Gutenberg blocks natively without third-party frameworks and developing custom plugins alongside custom themes, in order to separate site function from site appearance.

References

- 1 Liana Technologies. About Us – Facts and Figures [Internet]. 2021 [cited 2021 Oct 25]. Available from: <https://www.lianatech.com/resources/about-us/facts-and-figures.html>
- 2 Liana Technologies. About Us – History [Internet]. 2021 [cited 2021 Oct 25]. Available from: <https://www.lianatech.com/resources/about-us/history.html>
- 3 Liana Technologies. Websites – Overview [Internet]. 2021 [cited 2021 Oct 25]. Available from: <https://www.lianatech.com/solutions/websites/overview.html>
- 4 WordPress.org. Meet WordPress [Internet]. 2021 [cited 2021 Nov 15]. Available from: <https://wordpress.org/>
- 5 Fitzgerald A. What Is a CMS and Why Should You Care? [Internet]. HubSpot Blog. 2021 [cited 2022 Jan 11]. Available from: <https://blog.hubspot.com/blog/tabid/6307/bid/7969/what-is-a-cms-and-why-should-you-care.aspx>
- 6 Wibowo I. CMS Market Share Trends – What’s Changed in 2021 [Internet]. ToolTester. 2022 [cited 2022 Jan 11]. Available from: <https://www.tooltester.com/en/blog/cms-market-share/>
- 7 Opensource.com. What is open source? [Internet]. 2022 [cited 2022 Jan 12]. Available from: <https://opensource.com/resources/what-open-source>
- 8 WordPress.org. Support – History [Internet]. [cited 2022 Jan 11]. Available from: <https://wordpress.org/support/article/history/>
- 9 MacDonald M. WordPress: The Missing Manual. 3rd ed. O’Reilly Media, Inc; 2020.
- 10 WordPress.org. Make WordPress [Internet]. 2021 [cited 2022 Jan 11]. Available from: <https://make.wordpress.org/>
- 11 Leary C. Why is WordPress so Popular? [Internet]. Sitecare. [cited 2022 Jan 11]. Available from: <https://sitecare.com/blog/why-is-wordpress-so-popular/>
- 12 WordPress.com. Support – Post vs. Page [Internet]. [cited 2022 Feb 19]. Available from: <https://wordpress.com/support/post-vs-page/>
- 13 WordPress.org. Support – Taxonomies [Internet]. [cited 2022 Feb 19]. Available from: <https://wordpress.org/support/article/taxonomies/>

- 14 WordPress.org. Support – Post Types [Internet]. [cited 2022 Feb 19]. Available from: <https://wordpress.org/support/article/post-types/>
- 15 WordPress.org. Theme Handbook – Categories, Tags, & Custom Taxonomies [Internet]. [cited 2022 Feb 19]. Available from: <https://developer.wordpress.org/themes/basics/categories-tags-custom-taxonomies/>
- 16 WordPress.com. Support – Plugins [Internet]. [cited 2021 March 11]. Available from: <https://wordpress.com/support/plugins/>
- 17 Tuca A. Here Are the 10 Most Popular WordPress Plugins of All Time (Based on Data) [Internet]. Themeisle. 2022 [cited 2022 March 11]. Available from: <https://themeisle.com/blog/most-popular-wordpress-plugins/>
- 18 WordPress.org. Support – Using Themes [Internet]. [cited 2022 March 13]. Available from: <https://wordpress.org/support/article/using-themes/>
- 19 WordPress.org. Theme Handbook – What is a Theme? [Internet]. [cited 2022 March 13]. Available from: <https://developer.wordpress.org/themes/getting-started/what-is-a-theme/>
- 20 Scepter Marketing. How Many WordPress Themes Are There? [Internet]. 2022 [cited 2022 March 27]. Available from: <https://sceptermarketing.com/how-many-wordpress-themes/>
- 21 WordPress.com. Themes – Blank Canvas [Internet]. [cited 2022 March 27]. Available from: <https://wordpress.com/theme/blank-canvas-blocks>
- 22 Hughes J. What Is Gutenberg? An Intro to the New WordPress Block Editor [Internet]. Themeisle. 2021 [cited 2022 March 15]. Available from: <https://themeisle.com/blog/what-is-gutenberg/>
- 23 WordPress.org. About – Roadmap [Internet]. [cited 2022 March 15]. Available from: <https://wordpress.org/about/roadmap/>
- 24 Ventura M. Editor Technical Overview [Internet]. Make WordPress Core. 2017 [cited 2022 March 15]. Available from: <https://make.wordpress.org/core/2017/01/17/editor-technical-overview/>
- 25 Sowmedia. Tips for Keeping Your WordPress Website Up-To-Date [Internet]. WP Upgrader. 2018 [cited 2022 Apr 3]. Available from: <https://www.wpupgrader.com/wordpress/tips-keeping-wordpress-website-date/>
- 26 Elementor [Internet]. [cited 2022 March 13]. Available from: <https://elementor.com/>
- 27 Elementor. Hello Theme [Internet]. [cited 2022 March 13]. Available from: <https://elementor.com/hello-theme/>

- 28 IQComputing. Introduction to WooCommerce [Internet]. 2022 [cited 2022 Apr 3]. Available from: <https://www.wpupgrader.com/wordpress/tips-keeping-wordpress-website-date/>
- 29 WordPress.org. Plugins – Polylang [Internet]. [cited 2022 Feb 26]. Available from: <https://wordpress.org/plugins/polylang/>
- 30 Duò M. Advanced Custom Fields Tutorial: Your Ultimate Guide [Internet]. Kinsta. 2022 [cited 2022 Apr 25]. Available from: <https://kinsta.com/blog/advanced-custom-fields/>
- 31 The WordPress.com Team. The 12 Best WordPress Form Builder Plugins Reviewed [Internet]. WordPress.com. 2021 [cited 2022 Apr 27]. Available from: <https://wordpress.com/go/website-building/the-12-best-wordpress-form-plugins-reviewed/>
- 32 Miyoshi M. Getting Started with Contact Form 7 [Internet]. Contact Form 7. [cited 2022 Apr 27]. Available from: <https://contactform7.com/getting-started-with-contact-form-7/>
- 33 Advanced Custom Fields. Resources – Flexible Content [Internet]. [cited 2022 Feb 26]. Available from: <https://www.advancedcustomfields.com/resources/flexible-content/>
- 34 Advanced Custom Fields. Resources – Options Page [Internet]. [cited 2022 Feb 26]. Available from: <https://www.advancedcustomfields.com/resources/options-page/>
- 35 Max Mega Menu [Internet]. 2022 [cited 2022 Apr 25]. Available from: <https://www.megamenu.com/>
- 36 Assist Start Website Management & Development. Disadvantages of Using WordPress Plugins [Internet]. 2022 [cited 2022 Apr 25]. Available from: <https://assiststart.com/disadvantages-of-using-wordpress-plugins/>
- 37 WordPress.org. Plugins – Advanced Custom Fields [Internet]. [cited 2022 Apr 26]. Available from: <https://wordpress.org/plugins/advanced-custom-fields/>
- 38 Advanced Custom Fields. ACF 5.8 – Introducing ACF Blocks for Gutenberg [Internet]. [cited 2022 Feb 26]. Available from: <https://www.advancedcustomfields.com/blog/acf-5-8-introducing-acf-blocks-for-gutenberg/>
- 39 Delicious Brains Inc. How to Create Custom WordPress Editor Blocks in 2021 [Internet]. 2021 [cited 2022 Mar 27]. Available from: <https://deliciousbrains.com/custom-gutenberg-block/>
- 40 Local. Features [Internet]. 2022 [cited 2022 Apr 27]. Available from: <https://localwp.com/features/>

- 41 Bootstrap [Internet]. [cited 2022 Apr 27]. Available from: <https://getbootstrap.com/>
- 42 WP Pusher [Internet]. 2021 [cited 2022 Apr 28]. Available from: <https://wppusher.com/>
- 43 Advanced Custom Fields. Resources – Local JSON [Internet]. [cited 2022 Apr 28]. Available from: <https://www.advancedcustomfields.com/resources/local-json/>
- 44 Fonticons, Inc. Font Awesome [Internet]. [cited 2022 Apr 28]. Available from: <https://fontawesome.com/>

WordPress Template Hierarchy

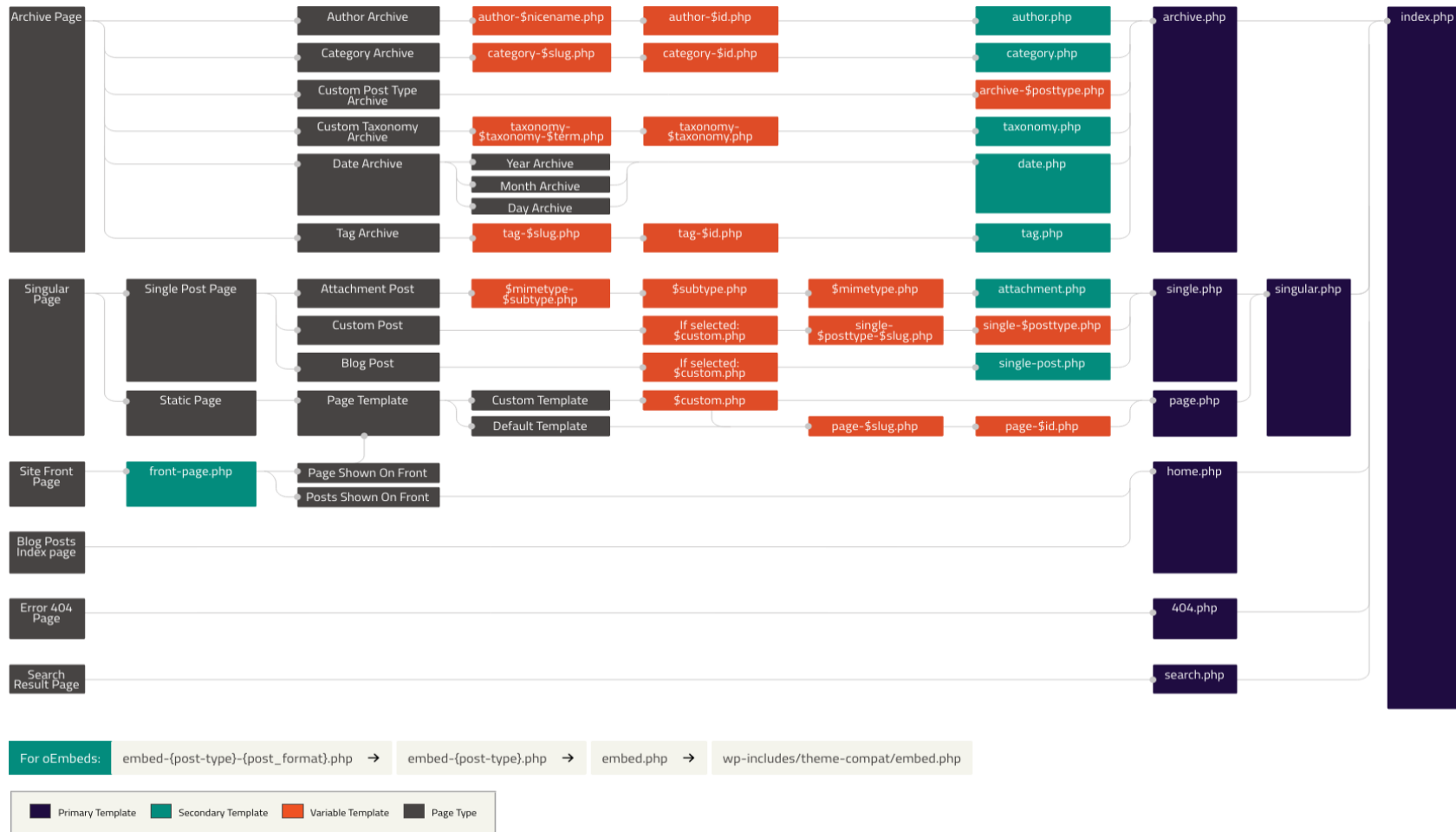


Figure 1. The WordPress template hierarchy. [18]