Ugyen Lama

NEWS PORTAL FOR NEPALESE STUDENTS APPLYING TO STUDY IN FINLAND (WORDPRESS)

Providing students with information and resources

Bachelor’s thesis

Bachelor of Engineering

Information Technology

2024

South-Eastern Finland
University of Applied Sciences

Degree title	Bachelor of Engineering, Information Technology
Author(s)	Ugyen Chheda Lama
Thesis title	News portal for Nepalese students applying to study in Finland (WordPress)
Commissioned by	
Year	2024
Pages	45 pages
Supervisor(s)	Reijo Vuohelainen, Moliterno de Camargo Ulisses

ABSTRACT

This thesis focused on the creation of a WordPress-based online site that would give complete information and tools to Nepalese students looking for higher education possibilities in Finland. The planned portal would act as a one-stop shop for students, providing thorough information on Finnish universities, such as their locations, possible fields of study, and admission procedures. In addition, the site would provide current student life news, covering important themes such as housing, part-time work options, and student events.

The portal would also have an interactive event posting and booking mechanism, allowing users to plan and attend activities relevant to the students in Finland. By using WordPress as the content management system, the portal would provide simplicity of use, flexibility, and scalability, making it a useful tool for both prospective and present students. The development approach included WordPress, theme development, plugin integration, and user experience design to build an informative and user-friendly site.

The end product was an accessible, efficient, and community-focused platform that helps the Nepalese students bridge the information gap as they navigate their academic and personal adventures in Finland.

Keywords: WordPress, theme development, full-stack development, PHP, jQuery, MySQL, AJAX, web design, information and news portal, international student support

CONTENTS

1 INTRODUCTION

Finnish institutions continue to attract international students, thanks to their strong educational systems and welcoming environment. Among these applicants, Nepalese students represent a steadily growing proportion. However, their journey is not without hitches, and they face difficulties or rather lack of easy access to good information on studying or even living in Finland. More and more students find themselves relying on expensive consultancies for a basic service as guiding them through the admission process, as there is no single reliable source for much of the information students need, such as where to live or how to find work. To fill this gap, this thesis is intended to build a single digital interface with the unified content of the social and practical context for giving necessary tools and information to the students to make a successful transition to their new lives in Finland.

1.1 Background

Currently, Finland is proving to be a preferred study abroad destination by the international students across the globe including Nepalese students because of the quality of the education, diversified discipline and friendly environment towards students. However, students often encounter various difficulties in getting the right and full information on universities, admission, accommodation, jobs, and campus life. These problems are compounded by the fact that the material is presented in several platforms making it almost impossible for students to make sure they get all the information they need.

Additionally, most of the students depend very much on educational consultants in Nepal that offer them services in applying for universities. These consultants often charge fees of up to €1000, and since the services are primarily for the students' benefit, they are the ones who have to bear the cost themselves. These agencies may help students with their papers for admission, but they do not support the students after they get to Finland. Hence, students may reach a foreign destination with no plan on how to take the next step, for instance; where to live, how to get a job or how to manage to survive in Finland. Most students enter the country ill-equipped due to the lack of a single all-in-one, online platform providing pre-arrival and post-arrival information. This gap results into confusion and makes the adaptation to the student life harder in Finland.

This thesis services cater to the general needs of the foreign students, and no specific niches are targeted for example niches like students from Nepal in the universities. A few of the Finnish specific websites such as StudyinFinland.fi provide basic information to overseas students including information about the universities and general aspect of student life in Finland. However, these platforms are not developed to cater for the needs of students and often these lack information about accommodation, jobs and community events that are very important for new students arriving in Finland.

1.2 Objectives

The overall goal of this thesis is to create a WordPress-based web site which acts as the single online source of information for Nepalese students interested in studying in Finland. In particular, the platform will feature materials about Finnish universities, entrance criteria, programs, and extensive information and tips about life in the country. The platform's free, self-service tools allow students to skip expensive educational consultancies and apply on their own. Moreover, the site will have important information sections about housing, part-time jobs, student activities and news to help the students adapt to new surroundings. To enhance interconnectedness of the members in the platform, elements for postings and booking of events will be included. To meet the needs of students efficiently, focus will be subjected on the aspect such as scalability, accessibility, mobile responsiveness, and user-cantered design.

The technology aspect will utilize WordPress framework (Messenlehner & Coleman 2014) to ensure the ease of use and scalability, while customised them development will be included to meet specific needs of the target audience.

1.3 Importance of the study

This study is important as it addresses two major challenges faced by students: the high costs associated with relying on consultants for guidance and the lack of adequate support once they arrive in Finland. The web platform will provide students with free and easily available advisory services which will help students to arrange their university applications on their own, thus saving money by avoiding the need to pay for these services. Moreover, the students will get full information about living in Finland, employment, and activities through the portal.

This project will also help to reduce misunderstanding when the students arrive to Finland and will help them to fit easily in the Finnish society because there will be constant support offered to them throughout the process of getting to Finland. The student event posting and booking tool foster interaction and expand communalism; thus, enhancing the total student experience.

2 PROJECT FEATURES AND THEIR SIGNIFICANCE

In this section, one finds the specific goal of this thesis, which entails developing a unique WordPress (McCollin & Blakeley 2013) news site. Therefore, the main objective of the project is to deliver a solution that would meet the needs of the media industry professionals, journalists, and citizens. The project's major features will be discussed in the sub-chapters:

2.1 Custom theme development

The WordPress theme (Onishi 2014) developed should be fully responsive, so that the users can access the portal with their mobile phones, tablets as well as computers. The style and design of the theme should therefore allow visitors to easily access news items, categories and tags. Some of aspects that should be displayed in the user interface include featured article sections, category menu, and a quick search.

The overall visual appeal is essential for the attraction and retention of users. As a result, the design must be rather formal but not too technical; otherwise, it becomes boring, while it has to provide such sections as a calendar, a search system, etc. The platform's main visual appeal should be enhanced through improved typography, a refined colour palette, and thoughtful layout choices and device responsiveness, ensuring a consistent and enjoyable experience across all devices.

2.2 Content management system (CMS)

If one has to manage a huge quantity of news pieces, it must be efficient. It should allow the journalists to write, edit and sort-out posts with ease via the CMS (Barker 2016). Custom fields enable the portal to work with different types of materials at once, for instance, a video, a picture gallery, or an article.

2.3 Theme and plugin architecture

In this particular work, many crucial features which would normally be implemented by plugins (Lefebvre 2022) were integrated into the custom theme. This strategy ensures that the site is simplified and fine-tuned for performance by loading less reliance on outside elements. The following features were added as part of the custom theme development:

2.3.1 Post-types and tagging

The theme also has a classification system included with it, which helps to categorize post type articles such as articles on event, news, universities and videos. This ensures that post type items are well arranged and hence easily retrievable in the blog or site. The tagging tool helps to tag items that are similar, which enhances the ‘findability’ of material in the application.

2.3.2 Multimedia handling

Instead of using a plugin in the multimedia content, the theme was developed to address various media types including the use of videos, images as well as the use of audio files. Employing this technology makes it easier for the journalists and content developers to include higher quality multimedia into their news stories giving the readers much richer experience. The theme’s ability to handle multimedia is lightweight and well-suited to the platform’s needs. It functions efficiently without relying on third-party plugins.

2.3.3 Multi-categorization and tags

An effective organization is critical for news platforms with a large amount of material. The goal of this objective is to build a multi-categorization and tagging system to improve news article classification and site searchability. It will enable all the post-types to be categorized into numerous relevant sections and then they can be searched under different topics or areas. This technique will particularly help in organizing articles in ambiguous or multiple categories such as interdisciplinary articles or news that transverse various categories.

A good tagging system will ensure that any visitor is able to quickly jump to other similar articles just by clicking on the right tags. For instance, such articles as “Technology” and “Innovation” will be grouped together to ensure that those clicking on either of them can find also another.

2.3.4 User roles and permissions

Often, understanding the user roles and the permissions gives a good understanding of the management as well as the functioning of the digital platforms most especially when it comes to the creation, posting and interaction with users. It originates from role-based access control (RBAC) model, which is widely known as the mechanism that allows users access only those facets and data which they require for specific job roles. In this platform, user roles are categorized into two broad groups: the content managers and the end users.

Content Managers Contributors:
It is the duty of contributors to develop and post the articles. It is mostly limited to the ability to write content, and thus they are not allowed to post or edit content. This restriction measures up the control by the editorial board and thus ensures that only quality material is published.

Editors/Contributors: Editing, revising and approval of articles which are submitted to the editor will also be done. They perform the important function of enhancing the legitimacy and quality of the material provided within the platform. Editors also investigate the issue of ensuring that the content to be published meets the intended goal of the platform, as well as its policies.

Administrators: Administrators hold the highest level of access, overseeing both the front-end and back-end of the platform. They manage user accounts, configure security settings, update plugins, and maintain the overall functionality and integrity of the system. They have full content and technical control of the platform as they edit its content and handle technical issues. They are responsible for managing the user accounts as well as the security of the system, and plugins. Moderators keep themes running, fix bugs, and update features to make sure the overall platform works well.

End Users

Readers: Readers are general users who visit the platform to consume published news and updates. They have no content management privileges but can navigate through categories, search for articles, and explore multimedia content.

Interactors: Interactors include readers who actively engage with the platform by commenting on articles, participating in discussions, and rating content. This role enhances user participation and creates a sense of community.

Every role has its own set of privileges consistent with the principle of least privilege and everyone else has been given limited access or no access at all.

2.3.5 Advanced searching and sorting

Complex searching options will be among the features of the news platform. It should be possible to search news by a number of criteria that can help to deliver only the content the user is interested in. Users will be shown items in the chronological order of the published date. Users will also have the opportunity to only show the items that interest them by certain keywords or words of their choice

2.3.6 Interactive user features

This goal is to enhance greater user participation by having features that will prompt the readers to respond. These features include the following:

- **Comments and discussion**

Readers will be able to express their opinions on articles posted making comments and contributing to discussions with other users or even responding to information in real time.

- **Article Rating**

The article rating feature allows readers to rate news and events, making it easier for others to discover popular or trending content. It’s a great way for users to share their opinions and for the site to showcase what people are loving the most. Ratings can be used to create sections like "Top-Rated" or "Most Popular," making it easier for visitors to find great reads and enhancing their overall experience.

.

- **Social Media Sharing**

Sharing buttons will be placed into each news story, to enable readers to share articles with friends on social networks including Facebook, Twitter and LinkedIn to propagate information.

2.4 Advertisement integration

Advertising has to be a key basis on which the long-term sustainability of the platform will depend upon. This purpose looks into ways to incorporate adverts without detracting from the user experience: Ads will be placed correctly so that they are not a problem when reading a text on a website. This is important to ensure that a great user experience is offered while generating income.

The platform will try out diverse ad approaches and models like banners and native ads to diversify incomes potential without compromising the content of the site.

2.5 Performance optimization

One of the objectives is the enhancement of the functioning of the news portal. To increase the number of users using such a site, quick loading of the page and easy to use features are important considerations. This will be done through:

- **Lazy lading**

Lazy loading in WordPress improves website performance by delaying the loading of images and iframes until they are needed. Introduced in version 5.5, this feature reduces initial load times, saves bandwidth, and enhances user experience, especially for mobile users. By using the loading="lazy" attribute, WordPress ensures only visible content is loaded first, improving page speed and SEO (Enge, Spencer, and Stricchiola 2015). Overall, lazy loading is a simple yet effective way to boost performance and create a seamless browsing experience.

- **Caching**

Through caching technologies, static copies of pages are stored and delivered more quickly to returning users, improving load times and overall performance. Caching in WordPress helps website run faster by temporarily storing frequently used data, like pages or database results, so it doesn’t have to be regenerated every time someone visits. This means visitors get the content quicker, and your server works less, making your site faster and more efficient.

- **Minimizing resource sizes**

Images, scripts, and other resources will be optimized to make the website load faster and perform better. By reducing file sizes and cleaning up unnecessary elements, pages will load more quickly, even for users with slower internet connections. This not only improves the experience for visitors but also makes the site more mobile-friendly and easier to use. Plus, faster load times can help boost search engine rankings and keep the website running smoothly, even when lots of people are visiting at the same time.

2.6 Scalability and future development

The platform will be engineered to accommodate more content and more users. With the growth in traffic, performance becomes another area of concern with regards to how both existing and new traffic will be accommodated. In addition, the architecture will also provide for the modularity or retrofitting thus ensuring that the platform can be updated to accommodate other advancement of technology and new users’ specific requirements as they emerge in future.

3 THEORETICAL BACKGROUND

WordPress is a free, content management system (CMS), used by millions, to create, manage and publish websites. Originally created as a blogging tool in 2003, it has expanded to become one of the most used CMS systems across the world helping to support about 40 % of all websites. The WordPress has a very simple interface, which allows people without much technical knowledge to create effective sites. The level of flexibility is here in various themes and numerous plugins which enable user to tailor the website in accordance with the needs, whether private bloggers, small, or big enterprises. It is free for use, can be modified, copied and distributed and is enhanced by a dedicated community of developers. WordPress has flexibility, with high compatibility that can accommodate new project, large projects, and can extend with new themes, plugins, and applications for purposes like business, portfolio, and content-based website. Available in two forms, WordPress.org, a self-hosted version with full customization capabilities, and WordPress.com, a hosted service offering a managed experience.

This project involves building a web platform using WordPress, leveraging its robust content management capabilities, customizable themes, and extensible plugin architecture to create a user-friendly, responsive and scalable solution tailored to the needs of the target audience.

The theoretical framework of this project outlines the necessary background information and technology for the creation of solid WordPress. This section covers areas including web-based news portals, content management systems (CMS), WordPress as a development platform, databases

(Silberschatz, Korth, and Sudarshan 2020), UX design (Gothelf & Seiden 2016) and web security (Zalewski 2011). Knowledge of these components enables appreciation of best practices that infuse the platform, as well as directing the creation of a safe, enable user-oriented space that disseminates useful information.

3.1 Web-based news portal

Namely, a web-based news portal is an internet-based technique that aims to deliver news, information and articles for a given target group. Such portals are usually designed with a range of materials, such as information on current events, authors' opinions, official statements, and any types of materials shared in multimedia forms that are potentially relevant for readers. Functions such as content sorting, user engagement, and customisation are highlighted in order to allow the readers to make use of the most timely and appropriate content. Web, HTML (Duckett 2011), CSS (Meyer 2017), jQuery (Duckett 2014), News portals use web technologies to build application and rely on PHP (Duckett 2022), Python (Lutz 2013) or Node.js (Casciaro & Mammino, 2020) to process data and manage content.

3.2 Web application development with full stack

Full stack (Northwood 2018) means that a developer works on both sides of an application, starting with the server and database layer as well as user interfaces. Regarding the concept of web applications, the front end shall work under the responsibility of UI and UX while the back end oversees business, data storage and data retrieval aspects. editable front-end frameworks like React (Stefanov, 2016), Vue (Passaglia, 2017) and Angular (Seshadri, 2018) enhance interactivity and performance. On the back-end side, one can find Node.js, Django (Melé 2024) and PHP to enable the dynamic content as well as data processing. In full-stack applications, there are API calls between the front end and the back end, and databases act as repositories for data in MySQL (Beighley & Morrison 2009), MongoDB (Bradshaw, Brazil & Chodorow 2019), PostgreSQL (Korry 2014), etc.

3.3 WordPress as full-stack web development platform

WordPress acts as a strong web application development tool since it combines front-end and back-end functions. At the front end- WordPress enables website developers to design simple and attractive websites through themes, these are the files that dictate the overall layout, design, and usability of a website. HTML, CSS, and JavaScript allow for customization of the themes, and responsive web design for devices compatibility along with adding the latest JS frameworks like React or Vue for interactivity features.

WordPress on the back end has a plethora of features, which is based on PHP and MySQL or MariaDB (Bartholomew, 2015). It offers assist to the content management extension with custom post types and taxonomies to help developers to sort and arrange the data as they wished. Plugin development enables one to incorporate server-sides functionalities such as APIs, advanced user authentication and business requirements. WordPress REST API helps build headless applications, where WordPress core is at the back end, with the front end using new JS tools.

WordPress connects front-end and back-end development based on the possibilities such as the template structure, where the content is generated with PHP, and AJAX (Holdener III, 2008) to implement interactivity with non-refreshed pages. It stands out from its competitors in full-stack deployments including fully customized web applications, WooCommerce-based e-commerce platforms, and content-focused websites, blogs, or news sites. As we have seen, WordPress has a lot of benefits that makes it suitable for web development including simplicity, flexibility, modularity, and scalability among others, but it has also a trend of disadvantages for instance performance, and more the use of plugins. Nevertheless, WordPress continues to be a flexible and open-powered environment for full-stack Web application development for various applications and users.

3.4 WordPress as content management system

Structured differently from CMS Definition, a Content Management System (CMS) is a system that allows users create, modify and foster content delivery. This is so because WordPress comes highly recommended as CMS that is both flexible, easy to work with and built on a very rich plugin architecture. WordPress which is created in PHP and MySQL allows the customization of the themes and plugins making it possible to develop the news portal site with many features. There are other CMSs which are generally of similar functionality to WordPress but may vary in flexibility, complexity and support; some of the best known include Joomla and Drupal.

WordPress is also an efficient development platform, especially when the site is built from a custom design and with custom plugins. WordPress themes are responsible for the designs and the format of the website, while plugins add on features. Themes and plugins work with WordPress APIs, Hooks and Filters to communicate with the core of this CMS and modify the content types, fields and interactions. Thus, WordPress provides a lot of tutorials, guidelines, and libraries, as well as several frameworks, such as REST API, which can turn WordPress into a powerful application for developing highly tuneable, scalable applications.

WordPress theming is an only mechanism that defines the layout of any WordPress based site. It employs PHP files known as template files to dictate how various sections of website including headers, footers, the sides, and content zones are to appear. Templating system at its basic level has a two-dimensional structure or hierarchy and WordPress uses this way to select a file that is most relevant or willing to render on that particular page or on that particular post. Figure 1 shows the visual representation of the WordPress template hierarchy.

Figure 1. WordPress Template Hierarchy (Source: WordPress Developer Handbook, https://developer.wordpress.org).

This modularity allows developers to design these themes and templates to cater for the specific page, category or post type.

3.5 Database and data management

The database is used for both storing and the handling of data. The two examples of Relational DBMS are MySQL, which operates with blogging platform WordPress, and uses SQL to efficiently retrieve, modify, and organize data, and another system that excels in managing structured data using SQL. While tagged news articles can be stored as values in keys, arranged sequentially, tables retain referential integrity and support joining multiple tables, which are options suitable for intricate data relationships such as in the case of categories. MongoDB is a good example of non-relational databases well suitable for applications that must be highly scalable or do not need a strict adherence to a defined schema; conversely, relational databases are well-used in CMS-based portals because of their stability and reliability.

3.6 Front end technologies and user experience

WordPress incorporates a stack of front-end technologies that make the websites developed with its help to be dynamic, interactive, and easy for use. In essence, HTML offers the content and structural framework, while CSS is applied for layout, colour, font and animation, to make the website sensitive to the device it is being viewed on. JavaScript when used with tools like jQuery, provides interactivity and dynamic functionality to the features like sliders, form validations and live content updates. The Gutenberg Block Editor is built using features that have adopted modern technologies including React to make content editing a block-based process. Furthermore, AJAX is used for update-any operations in the absence of page refresh, which improves performance and encourages users.

3.7 Web security

Web security is an important factor in WordPress since it is used by nearly every site in the World Wide Web today. This contains the basic however crucial layers of protection like updates, authorized use, role base access control, and authentication against threats. However, more steps are required for improving the security system. Such threats as brute force (Shema 2012), XSS (XSS Attacks 2007), and SQL (Clarke 2012) Injection can be resolved by security plugins that include Wordfence and Sucuri, SSL, robust passwords, and data backup. Measures include changing of themes and plugins, the disabling of certain features, XML-RPC among them; and use of real-time monitoring to achieve a tight security. The WordPress itself contains a set of relatively protective features when users adopt some standard procedures, the risks will be minimal, and users will have a credible website.

3.8 Performance optimization and scalability

The responsibilities and capacity are two of the most significant factors when it comes to the web portals with a vast number of users on the site. Lowering load times are achieved through options such as content caching, image compression and lazy loading among others in addition to the use of Content Delivery Networks (CDNs) to ensure content is delivered to different geolocations to suit the users from different parts of the world. Basically, on the server-side feature such as load balancing and database optimization is employed to cater for increased traffic.

Some of the optimizations regarded as specific to WordPress are decreasing the numbers of used plugins, optimizing the data base and using cache plugins to minimize the load on the server. Developing this theoretical base promotes a proper and detailed approach to defining the general concept of a web-based news portal, as well as its key technologies and methods. Such concepts as CMS, full stack, security, and scalability act as the technological foundation.

3.9 Search engine optimization

Implementing SEO feature is crucial in the news portals for improving the position of the news site on search engines and moreover an efficient site-search feature helps the visitors to find the relevant content easily. In SEO, factors like structured data, meta tags, keywords, and performance optimization that enhance the ranking. WordPress has SEO plugins that perform these tasks, and custom search enhances user features by enabling results customization through category or keyword.

3.10 Analytics and user tracking

Statistics is useful for the news portals to control the activity of the visitors and popularity of certain articles and posts. Google Analytics or other WordPress-specific analytics tools let you monitor the number of visitors per page, their length of stay and their activity. These findings assist the administrators of the sites to customize the sections of the content they offer, determine the hours of the day that most visitors log on and enhance customer satisfaction.

3.11 Advertisement integration

News portals depend on the advertisement and the management system such as Google AdSense or WordPress ad plugin helps in the advertisement placement, targeting, and tracking. Typically, ad management systems have items like targeting of ad areas, adaptable ad formats and reviewing of performance. The ethical issues like do not use pop-ups are also important for user experience in the current scenario as well. The advertisement for the companies with direct ad placement can be implemented by establishing agreement with the businesses.

4 PRACTICAL IMPLEMENTATION

The key features of WordPress including custom theme development, content management, performance optimization, and robust security, are designed to ensure a user-friendly and secure experience.

All the users’ interactions with microservices are classified according to the roles of the interacting entities. Administrators are responsible for the entire platform from the user administration to the installation and configuration of plugins to the security options and the underlying background. Contributors apply, assign editors to review and approve submissions, and various materials are controlled, checked, and maintained by editors for quality purposes. Many contributors provide articles or posts for consideration that the editorial board directly approves before getting published. Audiences consume published content, post comments on articles, like posts, engage in conversations, and share articles on social media. Some of the ordinary microservices include article management, user identity, multimedia support, event booking, search functionalities. These services are also layered with well-integrated, generalized plugins and APIs that support modularity.

The user interaction flow begins with contributors submitting content, which editors then review and approve before making it accessible to readers. Administrators manage users and configurations throughout this process. For instance, a user submitting an article involves content validation and storage in the database. Editors review the article using the content management service, and once approved, the post becomes accessible to readers. A database schema illustrates the relationships between users, posts, categories, tags, and metadata, showing entities like users (roles), posts (content), comments (interactions), and tags (organization).

The next sub-chapters discuss the methods implemented toward achieving these goals and making the platform scalable, secure, and efficient as it advances.

4.1 Technology behind WordPress

The WordPress is built on a foundation of PHP server-side scripting language, and it depends on MySQL or MariaDB options for database to store and manage content along user information and settings. It is created to perform on web servers such as Apache web server (Laurie & Laurie 2003), or NGINX (Fisher 2016), compatible with contemporary PHP types and SQL databases, to operate efficiently. For the frontend, WordPress uses HTML, CSS, and JavaScript to provide appealing and responsive design aspects. It also supports modern JavaScript frameworks such as React for Interactive UI. The presence of a RESTful API means that WordPress can be extended and, thus, integrated with other applications external to WordPress. This combination of technologies also makes WordPress not only easy to use, adaptable, and interoperable, but also readily extensible, secure, and appropriate for a variety of applications, ranging from simple personal sites through to corporate intranets.

In the context of service design, WordPress meets the basic needs of the targeted user personas (students who look for information on studying and living in Finland) through the system's structure for content search and validation as well as the definition of features. It affords easy integration of other components of the user interface and user experience like user friendly navigation, enhanced search and support for media, guaranteeing a bespoke solution to the user challenges.

From the system architecture perspective WordPress offers the coherent technological platform. The PHP-built core, MySQL DB, and one of the largest theme and plugin collections provide developers with best-of-breed tools and options to build a useful, robust, and enjoyable interface for their users. The features that include proper responsiveness and multilingualism make WordPress very reliable and available to its user across the different users' bunch and gadgets.

From the view of system engineering, WordPress allows iterative development since its structure is modularity based. Custom post types, tagging and event booking and even an advanced search can be developed in a sprint fashion as engineers develop this platform repeatedly, more elements that can be added over the platform can be developed in incremental sprints including this scalability enables smooth deployment and future enhancement as informed by user experience.

4.2 WordPress file structure

The WordPress file structure is designed in a way that categorizes the basic parts of the website, themes, plugins, and users' content.

- **Core Files**

Core files are stored in the root directory, as well as in the wp-includes, and wp-admin subdirectories. Its central architecture allows it to focus on system critical aspects of user authentication, route handling, and admin dashboard.

- **Themes and plugins**

Theme files are stored in wp-content/themes directory. A theme defines the design and layout of a website which includes files such as index.php, style.css, functions.php and other part of the website like header, footer, JavaScript files.

Plugins are located in wp-content/plugins directory. They provide additional functionalities like SEO optimization, building contact forms easily and faster or adding e-commerce features.

All the media files like images, videos and documents uploaded by user are stored in wp-content/uploads directory. They are stored in the chronological order of the date in the different directories created when uploaded.

Wp-config.php contains the database connection credentials and important site-specific configuration (WordPress Developer 2024, Editing wp-config.php). Figure 2 shows the contents in wp-config.php.

```
define( 'DB_NAME', 'projectThesis' );
define( 'DB_USER', 'root' );
define( 'DB_PASSWORD', '' );
```

```
define( 'DB_HOST', 'localhost' );
define( 'DB_CHARSET', 'utf8mb4' );
define( 'DB_COLLATE', '' );

define( 'AUTH_KEY','m_(&ryS*v{z5Deh?L]0^%Chew(Iug' );
define( 'SECURE_AUTH_KEY','Ug]gk.?{>|CF[5.{);
define( 'LOGGED_IN_KEY', 'H(wiWWtcO!)N' );
define( 'NONCE_KEY',' -Ktxa/cMZBY;.u0nv]lLil 3n,{E$' );
define( 'AUTH_SALT','D6ho/`2|+O^7PX*.tF5{U%/F5l) yS' );
define( 'SECURE_AUTH_SALT', `e9VqIZdjFW:$| Pn&' );
define( 'LOGGED_IN_SALT', 'cJ^K>v`+:|c/b[nM|a:A>O6`GZ');
define( 'NONCE_SALT','3W#O=*QzS}8fb+0pSfZJ-].CDfI0' );
define('PW_GOOGLE_API_KEY', 'AIzaSyC_gyEnPoVZq-j3bMU');

$table_prefix = 'pt_';
define( 'WP_DEBUG', false );
```

Figure 2. Wp-config file content

WordPress also contains the file .htaccess which is used for URLrewriting and additional server configuration.

WordPress installations follow a structured hierarchy to organize files efficiently. The wp-admin directory manages backend functionality, while wp-content houses customizations like themes, plugins, and media uploads. The wp-includes directory contains core files, classes, and functions. (Learn WordPress 2024, The WordPress file structure) Figure 3 shows the file structure of WordPress platform.

```
wordpress/
├── wp-admin/
│   ├── css/
│   ├── js/
│   ├── images/
│   ├── includes/
│   ├── admin.php
│   ├── menu.php
│   └── ...
├── wp-content/
│   ├── themes/
│   │   ├── your-theme-name/
│   │   │   ├── style.css
│   │   │   ├── functions.php
│   │   │   ├── index.php
│   │   │   ├── header.php
│   │   │   ├── footer.php
│   │   │   ├── page.php
│   │   │   ├── single.php
│   │   │   └── ...
│   │   └── ...
│   ├── plugins/
│   │   ├── plugin-name/
│   │   │   ├── plugin-file.php
```

```
│   │   │   └── ...
│   │   └── ...
│   ├── uploads/
│   │   ├── year/
│   │   │   ├── month/
│   │   │   │   ├── image.jpg
│   │   │   │   └── ...
│   │   │   └── ...
│   │   └── ...
│   └── index.php
├── wp-includes/
│   ├── css/
│   ├── js/
│   ├── images/
│   ├── class-*.php
│   ├── functions.php
│   └── ...
├── index.php
├── license.txt
├── readme.html
├── wp-activate.php
├── wp-blog-header.php
├── wp-comments-post.php
├── wp-config.php
├── wp-config-sample.php
├── wp-cron.php
├── wp-links-opml.php
├── wp-load.php
├── wp-login.php
├── wp-mail.php
├── wp-settings.php
├── wp-signup.php
├── wp-trackback.php
└── xmlrpc.php
```

Figure 3. Folder structure of a WordPress installation

WordPress also supports the creation of a network of websites within a single installation which is ideal for managing multiple blogs or website known as multisite.

4.3 Database

WordPress uses a structured relational database managed by MySQL or MariaDB. The WordPress database is divided into seven core tables. Plugins and advance customization may add additional tables to extend functionality such as for storing e-commerce transactions or event bookings.

WordPress uses a database to store and organize all the information needed to run a website. This includes user data, posts, pages, custom post types, and settings, ensuring that dynamic content can be fetched and displayed efficiently. The database consists of multiple tables, such as wp_posts for storing content, wp_users for user details, and wp_options for site settings. Each table has a specific purpose, contributing to the seamless functioning of a WordPress site (Learn WordPress 2024, The WordPress Database). Figure 4 shows the phpMyAdmin interface, which is used to manage the database for WordPress website.

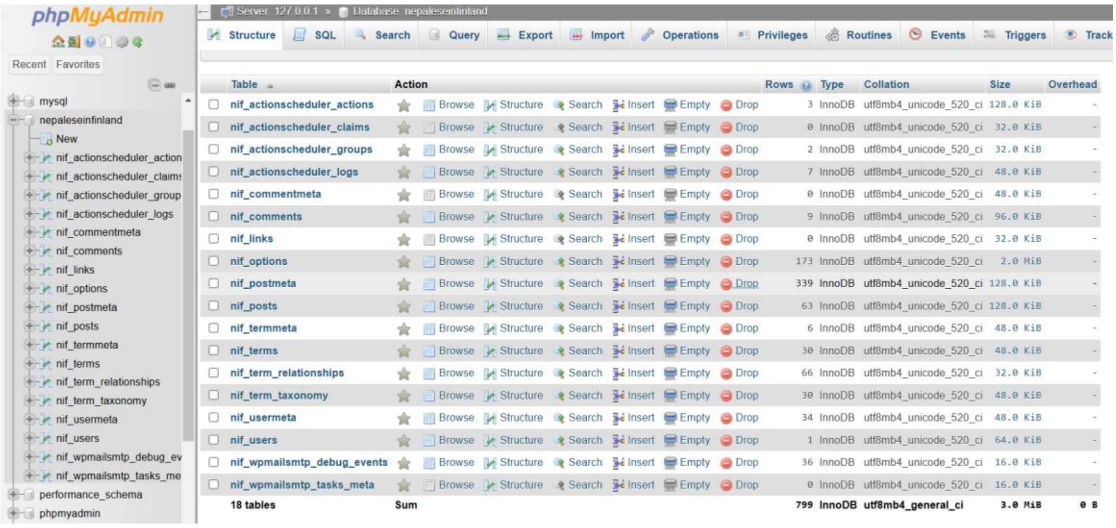

Figure 4. Overview of typical WordPress database tables in phpMyAdmin.

These tables are designed and complement each other in a way to hold all the necessary data that is required for running a WordPress site effectively. This structure enables the content, user interaction and settings to be viewable and editable in both WordPress dashboard and back-end database instruments like this one.

4.4 Custom WordPress theme development

WordPress theme was developed with the aim to provide a unique, easy to navigate and appealing interface that will cover all the necessities of users. The theme is also designed to be responsive for both desktop, mobile phones and tablets to increase screen coverage. It is important to address the diversity of hardware that users interact with through their devices, ensuring consistency in user experience.

A WordPress theme is composed of multiple files that work together to create the front-end design and functionality of a site. Key components include the style.css file, which serves as the primary stylesheet containing theme metadata and styling rules, and the functions.php file, a PHP script used to add custom functionality and enable theme features. The structure is further defined by template files like index.php, header.php, footer.php, page.php, and single.php, which delineate various parts of the site’s layout and structure. Additionally, optional templates such as archive.php, search.php, and 404.php handle specific use cases to enhance the site's versatility.

Themes integrate HTML, CSS, and PHP to deliver both static and dynamic content seamlessly. The WordPress Template Hierarchy plays a crucial role in determining which template files are used to render specific content types; for example, a single blog post is typically displayed using the single.php template. Developers also leverage the WordPress Customizer API to provide user-friendly customization options, enabling users to adjust elements such as colors, logos, and layouts effortlessly. Furthermore, themes can include widget areas and support custom menus, offering site owners flexibility in configuring the front-end to suit their needs. Figure 5 shows the typical file structure of a WordPress them, but the theme structure can be modified and expanded based on the specific requirements of a project (WordPress Developer, 2024, Theme Structure).

```
Theme Directory/
├── style.css
├── functions.php
├── index.php
├── header.php
├── footer.php
├── sidebar.php
├── page.php
├── single.php
├── archive.php
(categories, tags, etc.)
├── search.php
├── 404.php
├── screenshot.png
├── assets/
│   ├── css/
│   ├── js/
│   ├── images/
│   └── fonts/
├── template-parts/
│   ├── content.php
│   ├── content-single.php
content
│   └── content-page.php
```

```
├── inc/
│   ├── customizer.php
│   └── hooks.php
├── languages/
│   └── your-theme-name.pot
├── woocommerce/
│   ├── archive-product.php
│   ├── single-product.php
│   └── ...
└── README.md
```

Figure 5. Typical file structure of a WordPress theme.

Themes can be organized into logical directories to improve maintainability. Common directories include assets/ for CSS, JavaScript, fonts, and images, template-parts/ for reusable components like headers, footers, and content blocks, and inc/ for PHP files that handle custom functions, hooks, and other logic. Additional templates can be created based on the WordPress Template Hierarchy. For instance, files like category-news.php can be added for a custom category page or single-product.php for a custom single post type. Custom page layouts can also use files such as template-full-width.php. Dynamic naming and file inclusion allow WordPress to automatically look for specific files based on content or user needs. For example, author-{name}.php can be used for individual author pages. This capability enables highly dynamic and personalized designs by including custom template files as needed.

Figure 6 shows the snippet from the functions.php file, where I add the necessary styles and scripts (WordPress Developer 2024, Including CSS & JavaScript (Archived)).

```
function enqueue_theme_scripts() {
wp_enqueue_style('bootstrap.min',get_template_directory_uri().'
/assets/css/bootstrap.min.css');
wp_enqueue_style('style',get_template_directory_uri().'/assets/
css/style.css');

wp_enqueue_script('main',get_template_directory_uri().'/assets/
js/main.js',array(),_S_VERSION, true);
wp_enqueue_script('custom',get_template_directory_uri().'/js/cu
stom.js',array('jquery'),_S_VERSION,true);
}
add_action('wp_enqueue_scripts','enqueue_theme_scripts' );
```

Figure 6. Adding CSS and JavaScript to the theme.

The layout of the theme was designed with consideration of its readability and other usability features. I created a neat home page design providing areas for featured articles and posts, other recent post areas with clear demarcated categories to enhance experience. The theme also provides client-side templates for headers and footers to maintain a consistent layout across the site, while flow elements have been optimized to improve user navigation throughout the site.

Figure 7 represents the WordPress Data Architecture, which shows how different data objects, metadata, and relationships are structured and interlinked with WordPress.

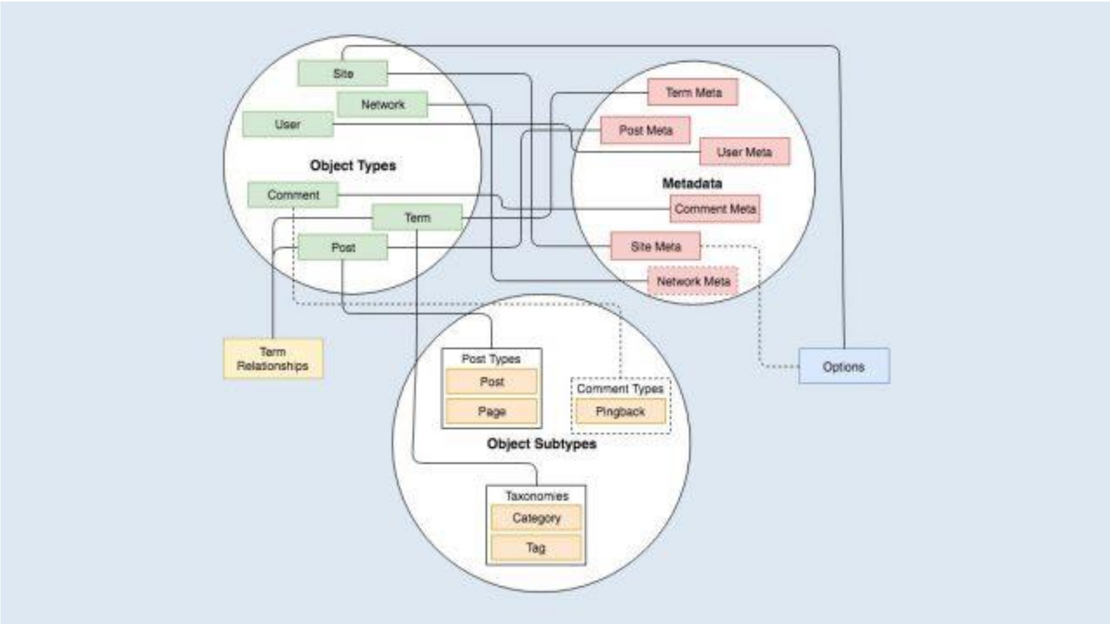

Figure 7. WordPress data architecture showing an overview of object types, metadata, and relationships in the WordPress ecosystem (Source: Felix Arntz, https://felix-arntz.me).

From the perspective of design standards, I chose a sensible colour palette with a touch of friendly appeal for the content, as well as typefaces that would improve legibility. Priority was given to the elements of site structure, position of category headings and posts which involve more attention of visitors. The design was kept formal to match the informational focus of the site but warm enough to encourage community participation (UX Design Institute 2024).

Design considerations and decisions also included specific post format styles for university profiles, event postings, and news stories and other formats to present content in an appropriate manner for each format. Figure 8 shows how the header and footer sections were defined in header.php and footer.php, respectively:

```
<?php get_header(); ?>
   <div class="content">
      <h1>Welcome to My WordPress Site</h1>
   </div>
<?php get_footer(); >?
```

Figure 8. Adding header and footer section to the template.

The structure of the code uses two WordPress functions: get_header which displays the site’s header information and get_footer which displays the site’s footer information. In between these functions, a section with a class attribute (error-area) is used most probably for displaying any error message like 404 error or serve as content area for users. Thus, it can maintain a high level of similarity in the general layout of the whole series at the same time as offering the capability to adjust the header and footer according to necessity.

4.5 Content Management Systems for posts and post types

To address the issue of the different types of content, WordPress was decided to be used as CMS since it is flexible and can easily be customized. Custom post types (CPTs) were made for news articles, events, universities, and video blogs as main categories. The CPTs themselves also had specific fields applicable to the contents of that CPT like dates, locations, prices, faculties, features depending upon its type to improve the management contents process (WordPress Developer Resource 2024, Custom Post Type).

Furthermore, Custom Fields and Meta Boxes were combined to enable more data, such as event, universities, and news details to be associated with any respective CPTs. It established a simple backend for administrators and contributors to work with, a backend that allowed for structured entry of data as well as better content categorization. Figure 9 shows how the new custom post type can be created scripts (WordPress Developer Resource 2024, Register Custom Post Type):

```
function uas_post_type() {
```

```
 $labels = array('name' => _x( 'UAS', 'domain' ),
 'singular_name'=>_x( 'UAS', 'domain'),
 'menu_name'=>__( 'UAS', 'domain'),
 'name_admin_bar'=>_( 'UAS', 'domain'),
 'archives'=>__( 'UAS Archives', 'domain'),
 'attributes' =>__( 'UAS Attributes', 'domain'),
 'parent_item_colon'     => __( 'Parent UAS :', 'domain'),
 'all_items'             => __( 'All UAS', 'domain'),
 'add_new_item'          => __( 'Add New UAS', 'domain'),
 'add_new'               => __( 'Add UAS', 'domain'),
 'new_item'              => __( 'New UAS', 'domain'),
 'edit_item'             => __( 'Edit UAS', 'domain'),
 'update_item'           => __( 'Update UAS', 'domain'),
 'view_item'             => __( 'View UAS', 'domain'),
 'view_items'            => __( 'View UAS', 'domain'),
 'search_items'          => __( 'Search UAS', 'domain'),
 'not_found'         =>__( 'Not found','domain'),
 'not_found_in_trash'=>__( 'Not found in Trash','domain'),
 'featured_image'    =>__( 'Featured Image','domain'),
 'set_featured_image'=>__( 'Set featured image','domain'),
 'remove_featured_image'=>__('Remove featured image','domain'),
 'use_featured_image'=> __('Use as featured image','domain'),
 'insert_into_item'  => __( 'Insert into UAS', 'domain'),
 'uploaded_to_this_item' =>__('Uploaded to this UAS','domain'),
 'items_list'        =>__( 'UAS list', 'domain'),
 'items_list_navigation' =>__('UAS list navigation','domain'),
 'filter_items_list' => __( 'Filter UAS list', 'domain'),
    );
    $args = array(
        'label'      => __('UAS', 'domain'),
        'description'=> __('Add all universities','domain'),
        'labels'     => $labels,
        'supports'=>
          array('title','editor','thumbnail','comments'),
        'hierarchical'          => false,
        'public'                => true,
        'show_ui'               => true,
        'show_in_menu'          => true,
        'menu_position'         => 6,
        'menu_icon'          =>'dashicons-welcome-learn-more',
        'show_in_admin_bar'     => true,
        'show_in_nav_menus'     => true,
        'can_export'            => true,
        'has_archive'           => true,
        'exclude_from_search'   => false,
        'publicly_queryable'    => true,
        'capability_type'       => 'page',
    );
    register_post_type( 'uas', $args );

}
add_action( 'init', 'uas_post_type', 0 );
```

Figure 9. Creation of custom post type named events

This CPT is designed for events in the project where functionalities include adding descriptions, categorizing events by tags and featuring them on the website. It makes sure that content is properly stored, easily retrievable and manged and is in compliance with the requisite of the site. We can add or remove the fields of custom post type as we wish.

4.6 Multi- Categorization and Tags in News Platform

WordPress is perfect for news portal content organization since it has built-in support for taxonomy. In this project, news stories, universities, events, video blogs were grouped using both hierarchical taxonomies (categories) and non-hierarchical taxonomies (tags). This method enables multi-categorization, which means that an item may be connected with certain tags (like "Admission" or "Locations ") and belong to more than one category (like Degree or IT). The theme was improved to get and display articles based on numerous categories and tags by modifying the loop and query methods. This enhanced the user experience by making it simpler to locate similar information. Figure 10 show how custom taxonomy can be created for categorization of specific post type (WordPress Developer Resource 2024, Register Taxonomy):

```
if ( ! function_exists( 'uas_taxonomy' ) ) {
    function uas_taxonomy() {
      $labels = array(
        'name' =>_x('Categories','General Name','domain' ),
        'singular_name'=>_x('Category', 'domain'),
        'menu_name' =>__( 'UAS Categories', 'domain' ),
        'all_items' =>__( 'All Categories', 'domain' ),
        'parent_item'=>__( 'Parent Categories', 'domain' ),
        'parent_item_colon'=>__('Parent Categories:','domain'),
        'new_item_name'=>__('New Item Category', 'domain'),
        'add_new_item'=>__('Add New Category', 'domain' ),
        'edit_item'=>__( 'Edit Category', 'domain' ),
        'update_item'=>__( 'Update Category', 'domain' ),
        'view_item' => __( 'View Category', 'domain' ),
        'add_or_remove_items'=> __('Add or remove Categories',
'domain'),
        'popular_items'=>__('Popular Categories','domain'),
        'search_items' => __('Search Categories','domain'),
        'not_found'    => __( 'Not Found', 'domain' ),
        'no_terms'    => __( 'No Categories', 'domain' ),
        'items_list' => __( 'Categories list', 'domain' ),
        'items_list_navigation'=>__('Items
navigation','domain'),
        );
        $args = array(
            'labels'                    => $labels,
            'hierarchical'              => true,
            'public'                    => true,
```

```
            'show_ui'                   => true,
            'show_admin_column'         => true,
            'show_in_nav_menus'         => true,
            'show_tagcloud'             => true,
        );
        register_taxonomy('uas_category',array('uas'),$args);
    }
    add_action( 'init', 'uas_taxonomy', 0 );
}
```

Figure 10. Creation of custom taxonomy name UAS categories for post type UAS.

The code shows how to create a custom taxonomy known as the UAS Categories in WordPress. It is also possible to create taxonomies that are different from categories, tags or other conventional types of divisions.

4.7 Creating Metadata for Enhanced Content Management

Custom Fields were created to store metadata specific to each post type. Using WordPress’s add_meta_box function, metadata fields were added to the post editor screen, allowing administrators to input additional details when creating or editing content (WordPress Developer Resources 2024, Add Post Meta). Figure 11 shows a code snippet from functions.php that registers some of the meta data fields for the "UAS" post type (CMB2 2024).:

```
add_action('cmb2_admin_init', 'uas_post_meta');
function uas_post_meta() {
      $cmb = new_cmb2_box(array(
          'id' => 'uas',
          'title' => __('More Info:', 'textdomain'),
          'object_types' => array('uas'),
          'context' => 'normal',
          'priority' => 'high',
          'show_names' => true,
          'closed' => false,
      ));
      $cmb->add_field(array(
      'name' => 'Hot uas',
      'desc' => 'Check if this is hot uas.',
      'id'   => 'uas_hot',
      'type' => 'checkbox',
      ));

      $cmb->add_field(array(
      'name' => 'Fees',
      'desc' => 'Add ticket price if uas is not free uas.',
      'id'   => 'uas_price',
      'type' => 'text',
      ));

   $API_KEY = get_theme_mod('google_map_api');
```

```
    $cmb->add_field( array(
      'name' => 'UAS Location',
      'desc' => 'Drag the marker to set the exact location',
      'id' => 'uas_location',
      'type' => 'pw_map',
      'split_values' => true,
      'api_key' => $API_KEY,
    ) )
}
```

Figure 11. Creating meta field for post type UAS.

The code defines meta boxes and uses the CMB2 library for WordPress platform to do so. Meta boxes can now be employed to incorporate more fields and filters to refine a post type called UAS and customization option for its meta information like admission dates, university location and conducting courses.

4.8 Event booking system

The functionality of the booking has been built to fit into the WordPress framework where users can make booking through a booking form. This form is purely meant to capture some broad information about the attendee including the attendee’s name, email, phone number, number of visitors, date of booking among others in case there is any message. For each submission, connections to event post types are created with the help of links done with WordPress custom meta fields. This integration means that all the data related to bookings are captured and localized within the WordPress admin dashboard to help event organizers to monitor and manage all bookings in real-time. The system also increases the actual level of convenience for users while providing administrators with convenient access to booking data. Figure 12 code snippet integrates a booking management system into a WordPress environment by creating and handling custom meta fields for event post types.

```
add_action('wp_ajax_submit_event_booking','handle_event_booking
');
add_action('wp_ajax_nopriv_submit_event_booking','handle_event_
booking');

function handle_event_booking() {
    if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
        wp_send_json_error('Invalid request method.');
    }
    if (isset($_POST['formData'])) {
        parse_str($_POST['formData'], $booking_details);
```

```
                $name =
        isset($booking_details['booking_details']['name']) ?
        sanitize_text_field($booking_details['booking_details
        ']['name']) : '';
        Similar fields for email, phone, message,date and
        details are created.
$post_id = isset($booking_details['event_id']) ?
absint($booking_details['event_id']) : 0;
$booking_info = "Name: $name\nEmail: $email\nPhone: $phone\nNo.
of People: $nopep\nBooking Date: $booking_date";
$existing_booking_info = get_post_meta($post_id,
'booking_details', true);
$updated_booking_info = '';
$existing_booking_info = get_post_meta($post_id,
'booking_details', true);
if ($existing_booking_info) {
        $updated_booking_info .=
wp_kses_post($existing_booking_info) . "\n";
         }

$updated_booking_info.="Name:".esc_html($name)."|";
$updated_booking_info.="Email:".esc_html($email)."|";
$updated_booking_info.="Phone:".esc_html($phone)."|";
$updated_booking_info.="No.of People:".esc_html($nopep)."";
$updated_booking_info.="Date:".esc_html($booking_date)."|";
$updated_booking_info.="message:".esc_html($booking_message)."|
";
update_post_meta($post_id,'booking_details',$updated_booking_in
fo);
$email = 'ugyenchheda@gmail.com';
$subject = 'Booking Reservation Details';
$message = $booking_info;
$headers = 'From: Your Website <ugyenchheda@gmail.com>';
$email_sent = wp_mail($email, $subject, $message, $headers);
if ($email) {
   $response = array('success' => true, 'message' =>
'<p class="mywight">Booking Success!</p>
 We will contact you with further information.
<br> Name :'.$name.'<br>Email:'.$email.'<br>Phone:
'.$phone.'<br> No. of People:'.$nopep.' <br> Booking Date:
'.$booking_date.'<br> Message:'.$booking_message.'',);
 } else {
       $response = array( 'success' => false,
        'message' => 'Booking successful, but failed to send
        the email.',);
      }
      wp_send_json($response);
   }
   wp_send_json_error('Booking details not provided.');
}
```

Figure 12. Dynamic event booking handler with secure post meta storage and admin notification.

In general, this code helps to specifically perform management and settings of the booking details of event post type in WordPress enabling the administrators to comfortably deal with bookings at the same time as promoting interactivity.

Respectively it retrieves data from a booking form, checks the validity of a request, and cleans user inputs so that personal data such as name, email, phone number, number of attendees, booking date, and any additional messages be safe. The details of the bookings are then stored a meta information linked with the particular event post to help event organizers manage bookings. Also, the code has an email delivery functionality which sends confirmation email to specific email address on success of a booking containing details of such booking in case of communication or updates. It also responds by returning a JSON array to show whether the booking was done or not and thus gives a user-friendly interface. This functionality seeks to help in managing bookings for events within WordPress in the most convenient way for all users including site administrators.

This code makes the addition of event booking system in WordPress whereby the booking information is safely collected then stored, event organizers are informed over e-mail and users are promptly replied in regard to their respective booking standings.

4.9 Search, filtering and sorting

A search engine is necessary to manage massive amounts of material. WP_Query, a class that supports more intricate search parameters, was used to expand WordPress' built-in search capabilities. An effective search system is necessary to handle high amounts of news items and assist visitors in finding pertinent material fast.

In this project, WordPress's default search capability was updated to give more accurate and user-friendly results. Because the search is based on keyword terms, readers may find news items by entering particular words or phrases. To guarantee that the most current and pertinent material shows up first, search results are arranged according to the date of publication. This method helps customers receive current news updates without requiring more intricate filtering methods by keeping the search feature straightforward but efficient. Figure 13 show code snippet which customizes the WordPress search functionality to only search for specific post types and exclude others.

```
function search_only_title_except_pages( $search, $query ) {
    if ( ! is_admin() && $query->is_main_query() && $query-
>is_search() ) {
        global $wpdb;
        $search = '';
        $search_terms = $query->get( 's' );
        if ( ! empty( $search_terms ) ) {
            $search = $wpdb->prepare(
                " AND $wpdb->posts.post_type IN ( 'post',
'news', 'events', 'uas' )
                AND $wpdb->posts.post_title LIKE %s",
                '%' . $wpdb->esc_like( $search_terms ) . '%'
            );
        }
        $exclude_page = get_post_type_object( 'page' );
        if ( $exclude_page && isset( $exclude_page-
>rewrite['slug'] ) ) {
            $exclude_page_slug = $exclude_page-
>rewrite['slug'];
            $search .= $wpdb->prepare( " AND $wpdb-
>posts.post_type NOT LIKE %s", $exclude_page_slug );
        }
    }
    return $search;
}
add_filter( 'posts_search', 'search_only_title_except_pages',
10, 2 );
```

Figure 13. Custom search query, restricting to titles of specific post types and excluding pages

This code customizes the WordPress search functionality to limit the search results to specific post types (post, news, events, uas) and only matches results based on post titles. Additionally, it excludes pages from the search results. This functionality ensures that the user is directed to more relevant website’s content, reduces unnecessary database queries, improving search performance and reducing server load. By controlling the type of content displayed in search results, we can highlight high-value or frequently updated content, improving user engagement and SEO rankings.

4.10 Search engine 0ptimization (SEO) for improved visibility

More specifically, underline of different SEO factors was made aiming to improve web presence and attract more users to the created portal. Thus, the platform enhances content and technical aspects aspiring to deliver relevant information to the students, who seek information concerning studying and living in Finland.

SEO-friendly URLs were achieved for all post types and taxonomies meaning the URLs are readable and relevant in subject matter. Moreover, title, description meta tags were introduced for every page to provide search engines with right keywords with necessary information about the contents' relevance (Semrush 2024). Figure 14 show how to add meta tags dynamically in the header.php file:

```
<meta charset="<?php bloginfo( 'charset' ); ?>">
<title><?php wp_title('|', true, 'right'); ?> <?php
bloginfo('name'); ?></title>
<meta name="description" content="<?php echo get_the_excerpt();
?>">
```

Figure 14. Addition of meta tags dynamically into header part from page title and excerpt.

This code is for the improvement of SEO and usability of the website. Although these are not all the categories that are obligatory for an SEO plan, they are critical to the technical approach to the website improvement for increased ranking and better customers' experience.

4.11 Analytics and User Tracking

Google Analytics was chosen as the primary tool for tracking user activity across the platform. With this integration, data is collected on user metrics such as page views and session, bounce rate and user demographics and location.
Figure 15 shows how Google Analytics was added to header.php for seamless tracking, using a sample Tracking ID:

```
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());
  gtag('config', 'UA-12345678-1');
</script>
```

Figure 15. Addition of Google Analytics to the theme.

This script is the Google Analytics which uses Global Site Tag (gtag.js) in order to track and monitor the website's traffic.

4.12 Interactive User Feature

Also to improve people's interest and make the platform concept more similar to the community, interactive elements were included. Such features provide an ability of users to interact with the content and share useful information, which will make the platform even more appealing and useful for students studying in Finland.

A sharing feature that empowered the users to share the posts to social sites like the Facebook, the Twitter and LinkedIn among others was included. This option makes it easier to disseminate information quickly and boosts the traffic in the given platform. Figure 16 shows the snippet which dynamically generates social media sharing links.

```
echo '<a href="https://www.facebook.com/sharer/sharer.php?u='.
urlencode(esc_url(get_permalink())).'</a>';

echo '<a href="https://twitter.com/intent/tweet?text='.
esc_attr(wp_get_document_title()).'. '.esc_url(get_permalink())
.'">a>';

echo '<a
href="https://www.linkedin.com/shareArticle?mini=true&url='.
urlencode(esc_url(get_permalink())).'&title='.esc_attr(wp_get_d
ocument_title()).'</a>';

echo '<a href="https://pinterest.com/pin/find/?url='.
urlencode(esc_url(get_permalink())).'</a>';

echo '<a href="mailto:?subject='.
esc_attr(wp_get_document_title()).'.'. esc_url(get_permalink())
.'"></a>';
```

Figure 16. Snippet to general social media sharing links.

For the generation of dialogues and maintaining a sense of people togetherness, a commenting feature was incorporated. Different persons can discuss the materials provided, share their thoughts, and questions with other readers. The native comment section in WordPress was enhanced for improved experience, regarding moderation options for nicely mannered communications.

A post rating feature was then incorporated in the development design to enable users to rate articles thus providing feedback on the quality and degree of popularity. This is quite useful to other readers in determining which post has been read most often or has received many likes.

4.13 Advertisement Integration

Ad placements were performed within a specified website section, for example sidebars, header and at the footer of an article so that they would not interfere with the consumption of content. Ads format like banners were used to select the more effective monetization models that could generate sustainable revenues without compromising user loyalty.

Customization option is created for adding banner in header, sidebar and footer part of the website using the WordPress Customizer API. It defines settings and controls to allow administrator to add advertisement banner with the link. Figure 17 show the snippet how to create functionality to add customization options for a footer advertisement banner using WordPress Customizer API.

```
$wp_customize->add_setting( 'adv_banner_footer', array(
'default'  => get_template_directory_uri() .
'/assets/images/ad/ad-1.png',));

$wp_customize->add_control( new WP_Customize_Image_Control(
$wp_customize,'adv_banner_footer',
array(
'label'    => __( 'Footer Adv Banner', 'domain' ),
'section'  => 'section_adv_banner',
'settings' => 'adv_banner_footer',) ));
$wp_customize>add_setting( 'footer_adv_link');
$wp_customize->add_control(  'header_text', array(
            'label'     => __('Link to Footer Adv.', 'domain'),
            'section'   => 'section_adv_banner',
            'type'  => 'text',
            'settings'  => 'footer_adv_link', ) );
$wp_customize->add_setting( 'adv_banner', array(
     'default'          => get_template_directory_uri() .
'/images/ad/ad-2.jpg',));
```

Figure 17. Customizing footer advertisements, adding dynamic banner and link options in WordPress theme customizer

With the WordPress Customizer API the adverts were then compiled and incorporated into the layout so that the administrators only needed to change the Ad link and Ad images by going to the theme source code. Figure 18 shows the code snippet to render dynamic footer advertisement using theme modification.

```
 <?php
  $footer_adv_link = get_theme_mod('footer_adv_link');
  $adv_banner_footer = get_theme_mod('adv_banner_footer'
  if(!empty($adv_banner_footer)) {?>
    <a href="<?php echo $footer_adv_link; ?>"><img src="<?php
      echo $adv_banner_footer ; ?>"class="img responsive"></a>
<?php }; ?>
```

Figure 18. Configuration of advertisement banner in the theme

This code lets the administrator pin and display a selectable customizable advertisement banner at the footer zone of the site. The features include changing the advertisement link and banner through the WordPress Theme Customizer to enable the easy editing of the administrators without codes.

4.14 Integration of Google Maps and YouTube Videos

For localization purposes, the platform uses Google Maps (Google Maps Platform 2024) for locations of events and universities and for multimedia content, the platform utilises YouTube videos. All these integrations provide the ability to view geographical context for events along with the access to video content concerned with studying and life in Finland.

This is as expected from the integration with Google Maps, location of the events and the universities will have a direct representation into the front end. (Google Maps Platform 2024). The map functionality includes search and autocomplete capabilities, allowing creator to type an address and receive auto-suggestions for matching locations, making it quick and intuitive to find specific places. The integration also automatically retrieves and saves latitude and longitude coordinates, ensuring accurate geolocation data without manual input. Figure 19 shows how google map is integration looks like in UAS post type for selecting its location.

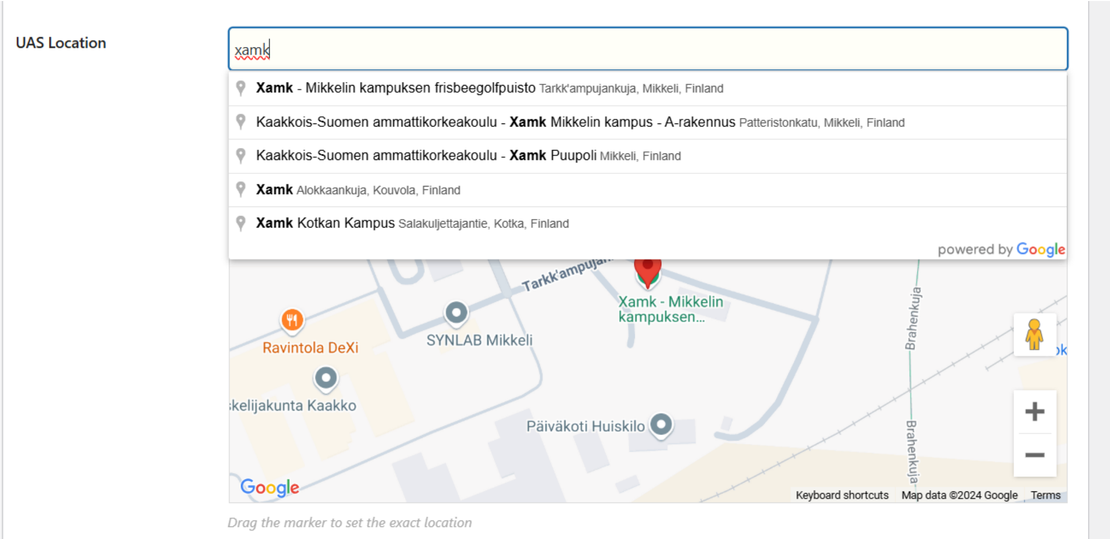

Figure 19. Backend Google Maps integration for selecting and associating university location.

A visual map is integrated directly into the meta box, enabling users to preview and confirm the selected location with ease. Additionally, the design is responsive, adapting seamlessly to various. Figure 20 shows the snipped used to retrieve the formatted address of a location based on its latitude and longitude using the Google Maps Geocoding API.

```
function event_location($latitude,$longitude) {
    $API_KEY = get_theme_mod('google_map_api');
$url =
"https://maps.google.com/maps/api/geocode/json?latlng=".$latitu
de.",".$longitude."&key=".$API_KEY."";
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($ch, CURLOPT_PROXYPORT, 3128);
    curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
    $response = curl_exec($ch);
    curl_close($ch); $returnBody = json_decode($response);

    $status = $returnBody->status;
    if($longitude||$latitude){
        if($status == "REQUEST_DENIED"){
            $result = $returnBody->error_message;
        } else {
            $result = $returnBody->results[0]
            >formatted_address;
        }
        return $result;
    }
}
```

Figure 20. Function to retrieve formatted address using Google Maps API

The platform is built to support the embedding of YouTube videos, which are essential for delivering multimedia content. The function is created which extracts the video ID from a YouTube URL, enabling the platform to embed videos using only the video url into the meta data of custom post types. Figure 21 show the function created to extract video ID from a YouTube URL.

```
function get_youtube_video_id($url) {
    $query_string = parse_url($url, PHP_URL_QUERY);
    parse_str($query_string, $query_params);
    if (isset($query_params['v'])) {
        return $query_params['v'];
    } else { return false; }
}
```

Figure 21. This function for embedding YouTube videos programmatically.

If the URL is https://www.youtube.com/watch?v=abc123xyz, the function will return abc123xyz, which is the YouTube video ID. If the URL is invalid or doesn't contain a video ID, it will return false.

4.15 Performance Optimization

To improve the speed of accessing the page other optimizations such as caching, reduction of resource usage, lazy loading and usage of AJAX based content loading were used to ensure that as the traffic increases the news portal can easily cope with the increased flow and still be able to respond faster to the users. AJAX improves the user experience. Instead of presenting a boring, static page, AJAX allows you to present a dynamic, responsive, user-friendly experience. Users can get immediate feedback that some action they took was right or wrong. No need to submit an entire form before finding out there is a mistake in one field (WordPress Developer Resources 2024, AJAX).

AJAX functionality also involves storing a copy of each page so that when the visitor returns it is easy to access because the file is still open. Both page and browser caching were then done using a plugin known as WP Super Cache. Page caching stores a copy of a modified page as a static page, while browser caching store resources such as CSS and JavaScript on the user's browser the next time the specific webpage is accessed. Figure 22 show how AJAX has been integrated for load more option in the homepage with advantage of seamless navigation and visual feedback enhancing the user experience.

```
jQuery(document).ready(function() {
    var currentPage = 1;
    var maxPages = 1;
    var loading = false;
    var loadedPostIds = [];
    jQuery('#fully-loaded').hide();
    jQuery(document).on('click', '#load-more-news', function()
{
  if (!loading && currentPage <= maxPages) {
    loading = true;
    var nextPage = currentPage + 1;
    var ajaxurl = my_ajax_object.ajax_url;
    var $appendContainer = jQuery('#append-here');
    jQuery.ajax({
      url: ajaxurl,
      type: 'POST',
      data: {action:'loadingNews',:nextPage,_news_category:
my_ajax_object.homepage_news_category,_of_news_hp:
my_ajax_object.no_of_news_hp,loaded_post_ids: loadedPostIds,},
beforeSend: function() {
        jQuery('#load-more-news').html('Loading’);
      },
 success: function(response) {
        jQuery('#load-more-news').text('Load More');
if (response.content) {
  $appendContainer.append(response.content);
  currentPage = nextPage;
  loading = false;
  loadedPostIds = response.loaded_post_ids || loadedPostIds;
  if (currentPage >= response.max_pages) {
     jQuery('#load-more-news').hide();
     jQuery('#fully-loaded').show();
} else {
   jQuery('#load-more-news').show();
   jQuery('#fully-loaded').hide();
   maxPages = response.max_pages || maxPages;
     } else {
         loading = false;
         console.error('No content in response.');
      }},
   error: function(xhr, status, error) {
 console.error(xhr.responseText);
 jQuery('#load-more-news').text('Load More');
 loading = false;
 alert('An error. Please try again.');
       }
     });
    }
  });
```

```
})
```

Figure 22. AJAX-based load more button with real-time error handling and updates

The AJAX “Load More” button allows users to load more posts into the page without the need to refresh the page which in turns improves on the page load time by only loading the contents that have been requested. This method enhances performance and cuts across the bandwidth thus makes browsing easier.

Resource Optimization entails minimizing image, script, and stylesheet size with a view of enhancing loading. Jpeg and png images were optimized for high quality in small sizes, whereas additional comic characters were removed using Autoptimize plugins for minifying CSS and JS languages. Code splitting means that only part of JavaScript and CSS loads at the beginning of the page, and the rest loads in the background. Figure 23 shows snippet for defining custom image sizes.

```
add_image_size('item_image', 300, 300, true);
add_image_size('post_image_xs', 100, 90, true);
add_image_size('post_image_s', 240, 186, true);
add_image_size('widget_right_thumbnail', 232, 123, true);
add_image_size('post_image_m', 263, 175, true);
add_image_size('post_image_l', 387, 242, true);
add_image_size('post_image_xl', 774, 484, true);
add_image_size('feature_galleries', 1090, 521, true);
```

Figure 23. Defining custom image sizes.

Customizing the image sizes in WordPress has its advantages such as not loading unnecessary width and height, unification of the design, and the ability to create a responsive design by generating images for different resolutions, as well as an improvement in loading speed, eventually improving the performance.

4.16 Web Security

For the platform security, several WordPress-specific steps are taken in order to preserve users’ data, to prevent unauthorized access, and to increase security level in general.

- Strong passwords and role-based access controls were implemented.
- Automatic updates are implemented for the newer versions of WordPress and plugins to fix security issues.
- Daily backups using UpdraftPlus are stored offsite, with regular testing of the restoration process to ensure data recovery capability in case of issues.
- WP Activity Log monitors the activity of users, and watches for suspicious actions; Error logging is turned on to capture server and PHP errors and to give information about some security concerns.
- Plugins such as Wordfence serve for Malware scanning, Firewall and Login protection. Scans are made on a frequent basis to identify spam files in the system and ensure platform purity.

5 RESULTS AND DISCUSSION

The aim of this thesis was to address the challenges Nepalese students face when studying and living in Finland. To achieve this, a WordPress-based platform was developed to provide essential information about universities, housing, jobs, life in Finland, and community activities. The platform was specifically designed to help students adapt to Finnish life while reducing reliance on expensive consultancy services.

This project was both exciting and rewarding, combining creative aspects like custom theme development, interactive features, and user-friendly design with practical elements such as event booking, multimedia integration, and advanced search. The challenges of using customized plugins provided a great learning opportunity, helping transition from a basic WordPress user to a more skilled professional capable of tackling advanced WordPress technologies. It offered valuable insights into custom themes, plugins, and performance optimization, while emphasizing the importance of user experience in creating intuitive digital platforms.

However, there were some challenges along the way. Fixing plugin compatibility issues was frustrating because some plugins were poorly documented, leading to a lot of trial and error. Balancing platform performance while adding complex features, like advertisement management, google map integration and multimedia, was also tricky and time-consuming. On top of that, advanced WordPress features like REST API integration and custom taxonomies had a steep learning curve, which sometimes slowed me down.

Regarding further development of the site, the next step is the integration of a translating function so that the site is available in different languages. This feature will make platform accessible to users from diverse linguistic backgrounds. Finnish universities' APIs are further intended to be integrated to offer real-time updates on admission, courses and academic information right within the platform. Further, an introduction of live chat service will be stimulating to ensure provision of real time support for users to ensure enhancement of the overall positive and more of a community type of language is encouraged.

Importantly, there is already strong interest and support for this platform from a group of Nepalese individuals who have been living in Finland for over decades and has expressed their willingness to assist in making this platform successful. They are willing to participate by assisting in the promotion and writing of the news and contents of the site, and by making certain that the site is useful to its users. Such involvement will play a critical role in extending the platform's visibility and by extension, its reliability. Additionally, collaboration with the Embassy of Nepal in Denmark (as there is currently no Nepalese embassy in Finland) could be pursued as a strategic step to gain institutional support for the platform. This partnership could enhance credibility and provide resources to expand the platform's reach and effectiveness, making it a valuable tool for the Nepalese community in Finland.

REFERENCES

Barker, D. 2016. Web Content Management: Systems, Features, and Best Practices. 1st ed. Sebastopol: O'Reilly Media [Accessed 9 September 2024].

Bartholomew, D. 2015. Getting Started with MariaDB - Second Edition. 2nd ed. Packt Publishing. [Accessed 7 September 2024].

Beighley, L. & Morrison, M. 2009. Learning MySQL: Get a Handle on Your Data. 1st ed. Sebastopol: O'Reilly Media [Accessed 6 September 2024].

Bradshaw, S., Brazil, E. & Chodorow, K. 2019. MongoDB: The Definitive Guide: Powerful and Scalable Data Storage. 3rd ed. Sebastopol: O'Reilly Media [Accessed 6 September 2024].

Casciaro, M. & Mammino, L. 2020. Node.js Design Patterns. 3rd ed. Birmingham: Packt Publishing [Accessed 6 September 2024].

Clarke, J. 2012. SQL Injection Attacks and Defense. 2nd ed. Burlington: Syngress [Accessed 6 November 2024].

Duckett, J. 2014. JavaScript & jQuery: Interactive Front-End Web Development. 1st ed. Indianapolis: Wiley [Accessed 3 September 2024].

Duckett, J. 2022. PHP & MySQL: Server-side Web Development. 1st ed. Indianapolis: Wiley [Accessed 5 September 2024].

Enge, E., Spencer, S., and Stricchiola, J. 2015. The Art of SEO: Mastering Search Engine Optimization. 3rd ed. Sebastopol: O'Reilly Media. [Accessed 4 November 2024].

Fisher, C. 2016. NGINX: A Practical Guide to High Performance. 1st ed. Birmingham: Packt Publishing [Accessed 6 September 2024].

Gothelf, J. & Seiden, J. 2016. Lean UX: Applying Lean Principles to Improve User Experience. 2nd ed. Sebastopol: O'Reilly Media [Accessed 3 September 2024].

Google Maps Platform. 2024. Maps JavaScript API. Web page. Available: https://developers.google.com/maps/documentation/javascript/add-google-map#web-components/ [Accessed 13 October 2024].

Holdener III, A.T. 2008. Ajax: The Definitive Guide. 1st ed. Sebastopol: O'Reilly Media [Accessed 13 October 2024].

Korry, D. 2014. PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database. 2nd ed. Sebastopol: O'Reilly Media [Accessed 13 October 2024].

Laurie, B. & Laurie, P. 2003. Apache: The Definitive Guide. 3rd ed. Sebastopol: O'Reilly Media [Accessed 13 October 2024].

Learn WordPress. 2024. The WordPress Database. Web page. Available: https://learn.wordpress.org/lesson/the-wordpress-database/ [Accessed 15 October 2024].

Learn WordPress. 2024. The WordPress File Structure. Web page. Available: https://learn.wordpress.org/lesson/the-wordpress-file-structure/ [Accessed 5 October 2024].

Lefebvre, Y. 2022. WordPress Plugin Development Cookbook. 3rd ed. Birmingham: Packt Publishing [Accessed 08 September 2024].

Lutz, M. 2013. Learning Python. 5th ed. Sebastopol: O'Reilly Media [Accessed 08 September 2024].

McCollin, R. & Blakeley, T. 2013. WordPress Theme Development – Beginner's Guide. 1st ed. Birmingham: Packt Publishing [Accessed 1 October 2024].

Messenlehner, B. & Coleman, J. 2014. Building Web Apps with WordPress: WordPress as an Application Framework. 1st ed. Sebastopol: O'Reilly Media [Accessed 6 September 2024].

Meyer, E. 2017. *CSS:* The Definitive Guide. 4th ed. Sebastopol: O'Reilly Media [Accessed 10 October 2024].

Northwood, C. 2018. The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer. 1st ed. London: Apress [Accessed 1 September 2024]

Onishi, A. 2014. Pro WordPress Theme Development. 1st ed. New York: Apress [Accessed 1 October 2024].

SEMrush 2024. Seo Best Practices. Web page. Available: https://www.semrush.com/blog/seo-best-practices/ [Accessed 26 November 2024].

Silberschatz, A., Korth, H.F. & Sudarshan, S. 2020. Database System Concepts. 7th ed. New York: McGraw-Hill [Accessed 12 September 2024].

Shema, M. 2012. Hacking Web Apps: Detecting and Preventing Web Application Security Problems. 1st ed. Waltham: Syngress [Accessed 5 November 2024].

Stefanov, S. 2016. *React: Up & Running: Building Web Applications*. 1st ed. Sebastopol: O'Reilly Media [Accessed 5 September 2024].

Study in Finland.2024. Finnish National Agency for Education. Web page. https://www.studyinfinland.fi [Accessed 5 September 2024].

UX Design Institute.2024. UX Design Principles. Web page. https://www.uxdesigninstitute.com/blog/ux-design-principles/ [Accessed 5 September 2024].

Vincent, W.S. 2023. Django for Beginners: Build websites with Python and Django. 5th ed. Still River: Still River Press. [Accessed 06 September 2024].

Williams, B., Damstra, D. & Stern, H. 2020. Professional WordPress: Design and Development. 3rd ed. Indianapolis: Wrox Press [Accessed 06 September 2024].

Williams, B., Tadlock, J. & Jacoby, J.J. 2020. Professional WordPress Plugin Development. 2nd ed. Hoboken: Wiley [Accessed 08 September 2024].

WordPress Developer Resource. 2024. AJAX. Web page. Available: https://developer.wordpress.org/plugins/javascript/ajax/. [Accessed 17 October 2024].

WordPress Developer Resource. 2024. Add Post Meta. Web page. Available: https://developer.wordpress.org/reference/functions/add_post_meta/ [Accessed 2 October 2024].

WordPress Developer Resource. 2024. Custom Post Types. Web page. Available: https://developer.wordpress.org/plugins/post-types/ [Accessed 26 September 2024].

WordPress Developer Resource. 2024. Including CSS & JavaScript (Archived). Web page. Available: https://developer.wordpress.org/themes/ [Accessed 4 September 2024].

WordPress Developer Resource. 2024. Register Taxonomy. Web page. Available: https://developer.wordpress.org/themes/ [Accessed 2 September 2024].

WordPress Developer Resource. 2024. Registering Custom Post Types. Web page. Available: https://developer.wordpress.org/plugins/post-types/registering-custom-post-types/ [Accessed 5 October 2024].

WordPress Developer Resource. 2024. Theme Structure. Web page. Available: https://developer.wordpress.org/themes/ [Accessed 5 October 2024].

WordPress Developer Resource. 2024. WP-Config Advanced Administration. Web page. Available: https://developer.wordpress.org/advanced-administration/wordpress/wp-config/ [Accessed 5 October 2024].

XSS Attacks. 2007. XSS Attacks: Cross Site Scripting Exploits and Defense. 1st ed. Burlington: Syngress [Accessed 10 November 2024].

Zalewski, M. 2011. The Tangled Web: A Guide to Securing Modern Web Applications. 1st ed. San Francisco: No Starch Press [Accessed 4 November 2024].