Md Saiful Islam

WordPress Theme Development and Customization.

Metropolia University of Applied Sciences

Bachelor of Engineering

Media Engineering

Thesis

23 April 2018

Abstract

Author Title Number of Pages Date	Md Saiful Islam WordPress Theme development and Customization 33 pages + 0 appendices 23 April 2018
Degree	Bachelor of Engineering
Degree Programme	Media Engineering
Professional Major	Web Development
Instructors	Ilkka Kylmäniemi,Senior Lecturer.

The main goal of this final thesis was to study advanced level WordPress and implement that knowledge to develop a custom website from scratch. Currently there are few content management systems available for web development. WordPress is one of the most used platform now. It has been contributing tremendously last fewyears in web development industry.

It is being adopted widely by its users for further development. Being picked by massive number of users mean that it has proved as an efficient and fast-growing platform compare to another CMS.

It considered to be an excellent web developing platform because it has shown a great user experience towards users. During the learning and developing process, the main idea was to evaluate the entire system to prove that WordPress deserves to be a top developing environment and certainly it did in the end.

Therefore, the fundamental structure, functionalities, features, adoptability, popularity were discussed in this study.

Keywords	WordPress, Theme Customization API, Loop, Hook.

Table of Contents

List of Abbreviations

CMS	Content Management System
XAMPP	Cross-platform(X), Apache(A),MariaDB(M),Php(P) and perl(P) .

1 Introduction

Web development refers to activities related to website functionalities development. There are three fundamental components that conduct interaction between user and site. These are client (front-end), server (back-end) and database. Web development plays a significant role by ensuring that user can react with computer in a certain manner. Basically, web development is a process of creating advanced and complex website.

Generally, there are many open source platforms available for developing website. Content management system is one of the significant one. WordPress has got immense popularity among the users because of its huge developer community.

Empowering one third of the websites in the world, WordPress is not only secure but easy to use with its drag and drop functionalities The administration panel, commonly known as admin, is loaded with user-friendly components such as page, post , widgets etc. Moreover, users can add custom functionalities to WordPress by adding plugins which are like extensions. With over 50,000 plugins available to download, users can easily extend WordPress to their needs.

The main purpose of the thesis is to create a custom theme. The process involves using HTML, CSS, JavaScript for frontend and php for the backend.

The fundamental concepts of theme development, customization process and different functionalities related to it will be covered in this thesis. The aim is to provide solid understanding of how a theme can be developed and later how to customize a theme. This custom theme development process may help users to build their own website with loads of features, functionalities and so on.

2 Theoretical Background

WordPress

WordPress is one of the most popular content management system based on php and MySQL among individuals and companies in web developing world. WordPress became a commendable platform and grabbed attention in web developer community tremendously for its simplicity. The user interface of word press is very simple (1) and easy to work with. It has a built-in updater which allows users to update plugins and themes within users and WordPress admin dashboard. Users can keep safe contents of a website by using option called WordPress backups. In the beginning WordPress was a blogging platform to create simple blog but now it has become one of the finest content management. It is an open source platform (2) that means free of charge to download and install. It also gives users opportunity to modify, adding features and play within. WordPress is easy to use and flexible enough to create functionalities without facing difficulties.

Drupal

Drupal is another content management system which is quite familiar in the web world. Hence, Drupal is rather complex (3) compare to the other content management system. The reason of complexity is Drupal has more advanced functionality and features. The depth of Drupal is much deeper than WordPress. In addition to that Drupal is very complicated and confusing for the users if they do not know how the complete system is working. Initially it’s difficult to operate for the beginners who tend to work on Drupal. If users need multiple page templates in a system, Drupal is very reliable to handle users demand. Drupal contains things like site admins, content editors, private content etc. As a matter of fact, Drupal supports multiple stakeholders. Basically, Drupal was designed to make functionalities either strong or weak. If users are not comfortable enough to work on Drupal, the users might end up ignoring the advanced features that makes it so powerful and unique.

Generally, people are likely to work with WordPress rather than working with Drupal because WordPress plugins are rather easier to manage. But in Drupal it is necessary to hire developer, or they should be skilled enough for installing or updating modules. Drupal requires technical experience to operate it and it is being able to produce advance products for the clients or companies. Eventually, Drupal has such ability to control vast element of content. Usually Drupal is bit expensive though when it comes to the making website with high customization and it is very useful to create well organized website with loads of content.

Joomla

Joomla is open source content management system and it is based on model viewcontroller (4) (MVC) web application framework that allows to build powerful application and it helps to create dynamic website as well. It has a unique management interface that control all the functionalities and features. There are many extensions written (4) for Jooma that allows users to add functionalities, features and customize websites depending on their requirements.

Having said that Jooma is supporting only web developing communities but it also supports database and web server. It gives developer a strength to concentrate on development instead of having concerned about how website looks like in front -end.

As a web framework Jooma is significant in making simple php based application. The application may have those selected features in framework that are suitable for application without using the whole content management system. However, Jooma is used mainly to make super light web application which does not contain any difficult functionalities.

WordPress as a platform.

WordPress is such a massive platform in web developing industry because there are thousands of themes available for free and any of them can be used in a theme. So, it is possible to customize websites. A WordPress theme has many alternative templates to develop features for a site such as category pages and search result pages. There are different kind of WordPress theme available such as portfolio theme, e-commerce theme (1) etc. Furthermore, a theme can be developed from the scratch using WordPress. Themes in WordPress are easy to customize when default themes are chosen from WordPress itself. Because default theme has already got own option panel to allow change the colours, images, background colour, layouts etc. But if a theme is developed from the scratch, few things must be set up in theme developing system according to WordPress instructions. Each theme might be different, but it allows users to choice many options for their own website to switch in a different look instantly. A WordPress theme does not necessarily require much coding unless creating something unique and cool. A developing theme is being able to teach users more about html, css and php because in the front-end end loads of styling, looping, creating many features etc.

How theme works in WordPress

Understanding how theme works in WordPress (5) is quite important to a theme developer. There is a difference between template and theme especially when it comes to the WordPress. In addition, template is one file or could be more where codes are already written and that will call upon a theme. Basically, templates like header.Php, index.php, footer.php are called upon by theme. A theme is based on template files, images etc which tells the users browsers how a page will look like. However, a theme can be made of many template files, but two files are must be in the theme to create a theme in WordPress. They are index.php and style.css.

Themes

In WordPress a theme is considered when bunch of template files and stylesheets work together to define display of a WordPress developed website. A theme can be changed and added to the WordPress admin area under appearance (1).

There are several default themes available in wordpress.org own directory. Most of them are free but few of them are paid.

Each of the theme comes along with distinctive style, layout, design, functionalities, features etc. Generally, website making in WordPress is completely depending on users wish, taste and their requirements. WordPress has got many default themes and they are very specific such as there is a theme called photography theme which is designed only for creating photography websites.

Additionally, a theme can be modified by adding plugins or adding code in function.php file. Generally, a theme consists of three parts: style.css, template files and function.php file that allows user to change anything in custom theme.

Every theme must contain unique style.css file but it does not mean that other themes cannot have same style.css file. Usually the problem is that, as an example identifying information in the header must be different otherwise theme developer will have tough time during installation. Despite that, if it is not a child theme then a theme should have at least an index.php file to control the display of content. Child themes is very simple and includes only style.css file and images.

.

2.1 What are templates

To begin with, Templates files that work together to create a website and control how WordPress applications will look to the users. These files collect information from WordPress, MySQL (6)database and generate the html code and send it to the web browser. The templates can be used to customize a theme under many conditions. Template files are very powerful when it comes to develop a theme because users can add basically anything to the theme by adding solid codes in the template files to create theme functionalities better and easy to run.

2.1.1 Template Tags

Template tags are (6) the php functions that are built into the WordPress for generating and displaying information on the templates. WordPress ships with various template tags relating to post, author, comment, navigation etc. For instance:

```
<? php the_author(); ?>
```

This shows the post author .

2.2 Templates File hierarchy

To work with WordPress, developers must have notion to how hierarchy works in WordPress. Template files are modular, reusable (7). They are used to create website in (5)site pages while other template files used only in terms of certain conditions.

WordPress uses query string to decide either which template or set of templates going to be used to display the page. the string information will be contained in the link to every part of a website.

WordPress searches down in template hierarchy which template file should be suitable to work with depending on specific conditions of website. Usually every string query matches to query type to decide which page has been requested for loading. For example, a category page, search page. If WordPress cannot find the template file with matching name, it will skip to the next file, even if it doesn't find the matching name file at all then theme will end up using main index file.

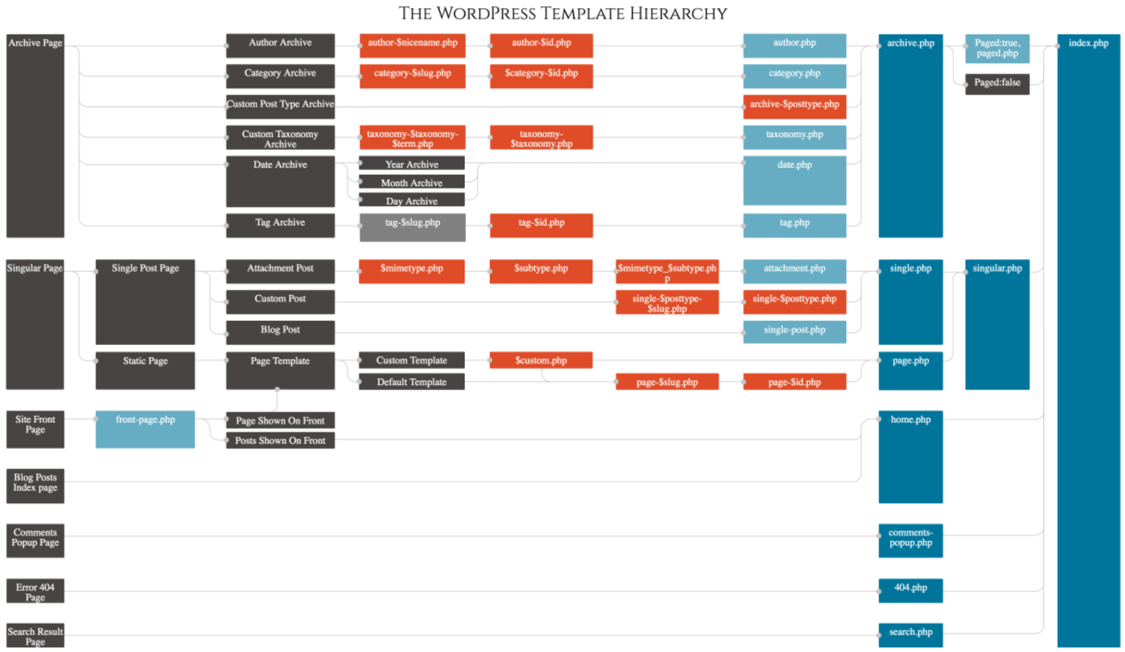


Figure: WordPress template hierarchy

2.3 Functions and loop

Functions

Functions. Php is another important file in WordPress. Theme developers will have to deal with functions consistently while they are developing a custom theme. For example, the_title or get_header file or wp_head function (5). Basically, function can work in WordPress in numerous ways such as they can return data to a variable or they can bring output straight to the template. As an example, two functions can act on similar content. Get_ the _title function return output to the string and the_title function will echo it straight out to HTML.

//get _ the_ title () returns title to a variable for use in php code.

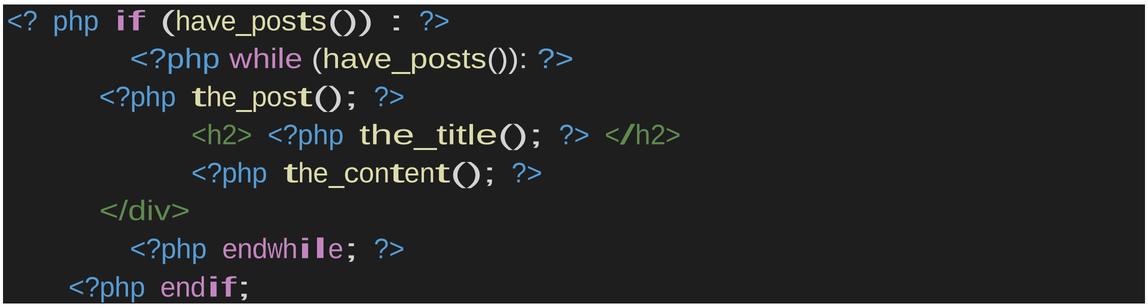
$title = get_ the_ title ();

// do things with $title variable

Echo" <h1>$title</h2>

The Loop

The most exciting thing in (8) WordPress is loop function and it has ability to insert content dynamically to the theme. This is one of the most important function in WordPress. For example, if users go to the dashboard and click on posts, "Hello world" post will be appeared by default.



```
<? php if (have_posts()) : ?>
        <?php while (have_posts()): ?>
      <?php the_post(); ?>
            <h2> <?php the_title(); ?> </h2>
            <?php the_content(); ?>
      </div>
        <?php endwhile; ?>
    <?php endif;
```

Figure: Loop in WordPress.

According to the code if there are posts, while there are posts, display the post. Apparently, anything inside the loop will be repeated if individual post continues but loop will end when individual post will end. For example, if there are five posts, loop will continue for five times then it will stop. Loop function should be in index file according to WordPress content management system. Furthermore, both post and page use loop function.

2.4 Hooks in wordpress

Hook plays a revolutionary role in WordPress especially when it comes to the customization part in WordPress. There are two hooks that are used for many developing aspects of a theme. They are Action hook and filter hook. Basically, (9) both are functions and they can be applied to change the WordPress default theme to custom new theme. Moreover, action hook and filter hook are not the same. WordPress action hook applies when certain event occurs in WordPress and filter hook able to modify certain functions. It's possible to change the interface of WordPress from theme or plugin by using hook without touching any core code of WordPress.

2.5 Wordpress API

WordPress provides different APIs for further customization. (10) Database, Options ,Theme Customizer ,HTTP are few examples of these APIs. The theme uses following APIs.

2.5.1 Theme Customization Api

Theme customization API is a framework that allows to live be previewing any changes for a developing website in WordPress. It helps users to add more powerful interactive option to the theme. It enables users to customize the several aspects of a theme or site from colour to menu, layouts etc. By using function file, themes, plugins, many options can be created additionally to the customization area. (11) In customization area users will be able to see few options by default because WordPress has already built in options that will be visible in customize area as soon as theme is created. Once many options are added to the customized area then other users might get to see those customize options if it is allowed for them to be checked. Different part of the customizer will be depending on relating to the statements of the theme that whether they are relevant to the front end or not which users is previewing.

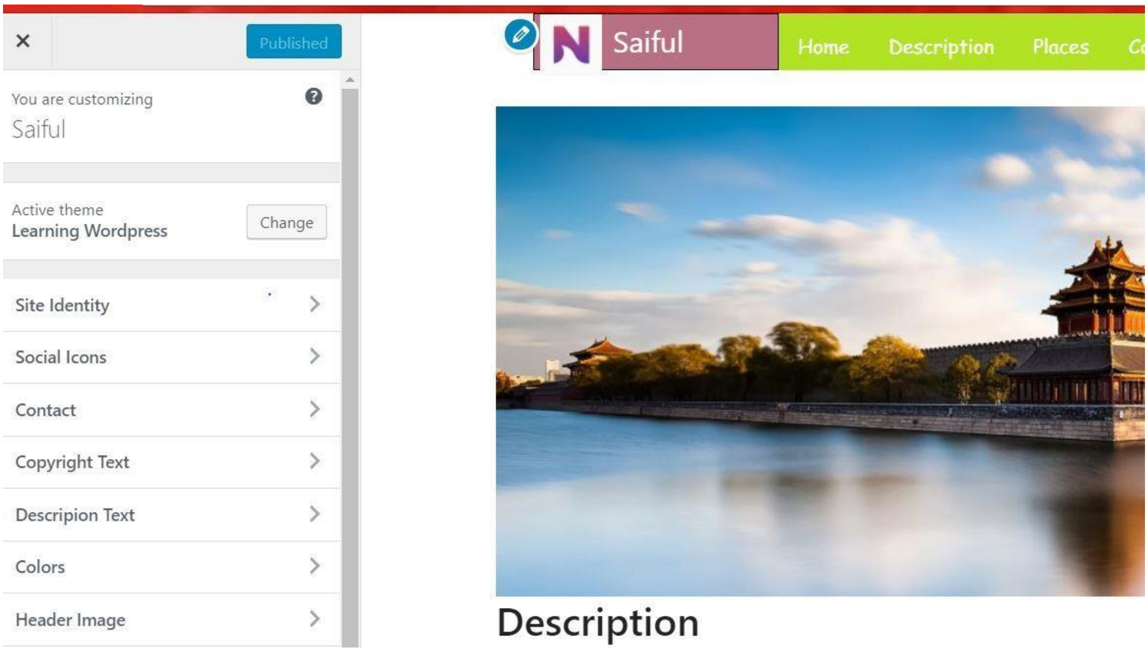
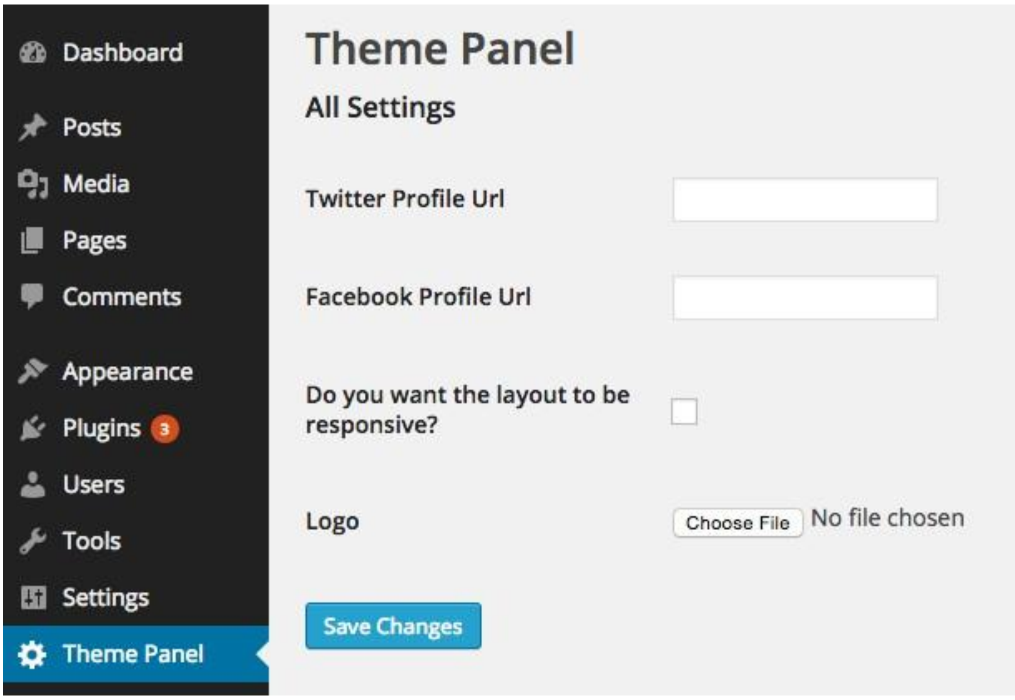


Figure: A customized API

2.5.2 Settings API

Using the Settings API, we can create custom setting to the theme. For example, Figure 1 shows custom setting called Theme Panel which has a form that let users input their social links and logo.



Theme Panel

All Settings

Twitter Profile Url

Facebook Profile Url

Do you want the layout to be responsive? □

Logo No file chosen

Figure 1: Setting API

3 Technologies Used

3.1 HTML & CSS

Html is a hypertext mark-up language. It is used to create pages that displayed on the web. There are series of connections available at each page to another that's called hyperlinks (12). Every single web page available in internet is used at least one version of html code or another. The html code provides format images, texts in a order that page based on which cascading style sheet is placed to change the appearance.

CSS

Cascading style sheet(CSS) is a stylesheet language used to format the layout of webpages. Generally, CSS used to define the text styles, colour of text, table sizes etc. It helps developer to create uniform look in web pages (13).

PHP

Php is a server-side scripting language (12). This scripting language mainly uses to build dynamic websites. PHP can create username, password, login page etc. Php code is inserted inside the html code. As an example, when clients visit a webpage that page contains the code and code will execute by server. That is why server should be installed in a local computer to test php locally. In addition, users don’t need any special plugins to see php in action at server. whereas, it is a scripting language it does not need to be compiled before it gets used. It works fine with MySQL database server and what happens is that when php code execute on the server, the html contents will be displayed in the browser.

3.2 MySQL

MySQL is one of the most widely used open source database management system in the world. is a database. The database system manages the databases and connect them with software. For instance, MySQL database can be used to run the website which built on database-based programming language. (12)MySQL is not that familiar for Larger system but mostly it will run for Microsoft SQLl server or oracle. MySQL is very fast, flexible, reliable and easy to operate. It helps to organize information’s and update databases whenever users wish to do. It is a highly demanding environment because its speed, connectivity, security allow to access databases faster and frequent on the interne

3.3 XAMPP

Xampp is a software package and it is an apache distribution for apache web server, MySQL database server, php server and many more. It is a local host or server that works on local computer (12). It is an open source development software. It works for all operating system such as google chrome, Mozilla Firefox etc.

Generally, xampp gives an opportunity for its user to test php, MySQL, Perl, apache project on local computer. If xampp downloaded already in a local pc, the user does not necessarily require internet connection to run the xampp projects. In addition, it gives an access to the users to test website before uploading it to the remote web server.

Moreover, two modules apache and MySQL needed to run the WordPress application in local computer.

3.4 Requirements

WordPress installation

There are few steps to follow to install WordPress. It is very easy to install; the process takes few minutes to complete. Nowadays many host offers tool called (**fantasico**) for automatic installation of WordPress (14).

Before starting installation, users need to have certain things such as

- Access to Web hosting server.
- Being able to create MySQL database.
- A text editor. There are many text editors out there.
- ftp clients, (File transfer protocol)

Browsers to run the programme. (Google chrome, Mozilla Firefox etc)

Now users should do few things to install WordPress

- make sure web host and users have ability to run the WordPress.
- Download the latest version of WordPress.
- Unzip the downloaded file in hard drive.

- Security key always should have secure password.

3.5 Installation

1. download xampp in local pc.

2. users must create a database for WordPress on web server. Normally Php myadmin administration tool is good for MySQL database. MySQL database has all the privileges to access and modify it.

3. **xampp** contains many files but there is a file name called "**Htdocs**" where downloaded WordPress unzip file should be located. Once users click on WordPress file, they will be able to see many WordPress files and users need to select one specific file called **wp-config** to define database name and password. Eventually users can start working on WordPress.

4 Theme Developement

The main objective was to create a dynamic website based on WordPress. the custom theme has been named "Learning WordPress". Learning WordPress" theme created for developing a website based on adventure. The Initial idea was to develop a website and customize it with the help of customization API framework (15).

Initially the development process was a bit confusing to structure the whole website because WordPress has got many functions, plugins, etc available by default.

In the beginning the header, footer, body, sidebar was created by conventional coding process. As development process goes by, the WordPress provided many default functions that helped through the critical aspects of developing process.

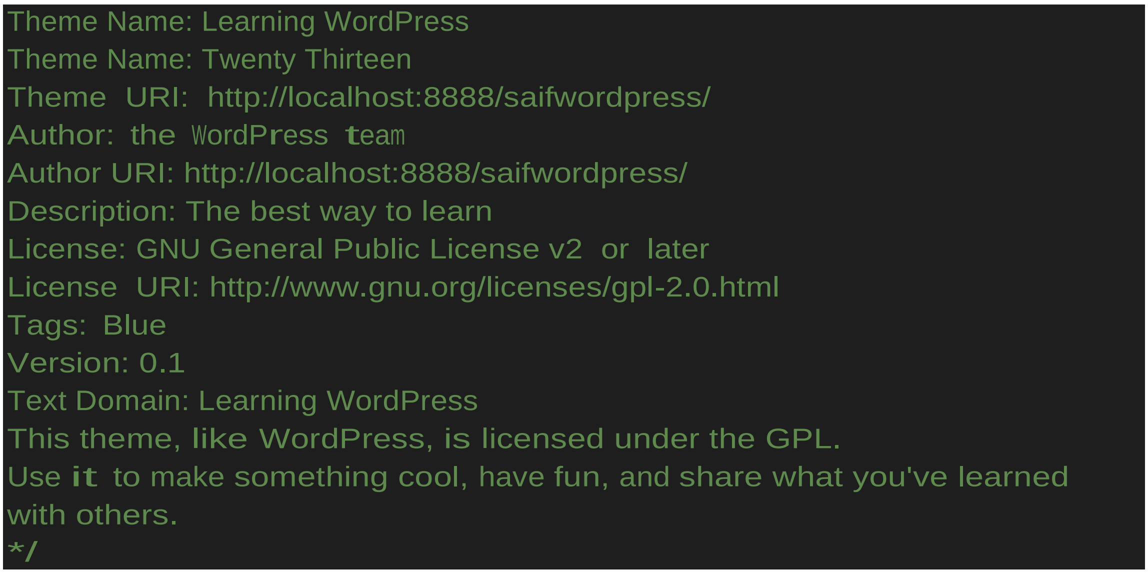
Creating a custom theme

Basically, WordPress has many folders and files but one of the folder is very significant which is **wp-content** folder where new theme file should be created separately. In the same folder there are few default themes available. In addition, a new theme can be switched to the default theme to see distinctive look of website. Perhaps a user might be able to get innovative ideas or adding new functionalities and features in terms of switching custom theme to default theme as a source of developing website.

Creating template files

A WordPress theme needs to have at least two files to exist. Index.php and style.css.

In custom theme, at first create a file called style.css. It contains a comment that quickly notifies the system that a theme exists here and then change the name, author, description etc (8) .

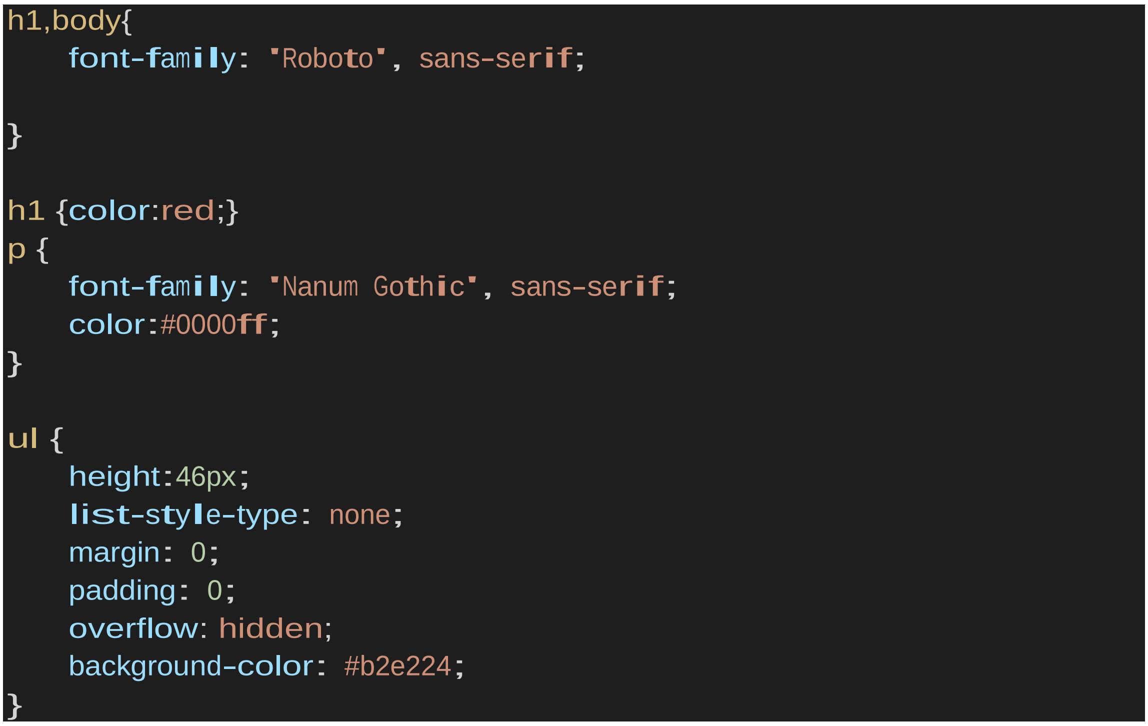


```
Theme Name: Learning WordPress
Theme Name: Twenty Thirteen
Theme  URI:  http://localhost:8888/saifwordpress/
Author:  the  WordPress  team
Author URI: http://localhost:8888/saifwordpress/
Description: The best way to learn
License: GNU General Public License v2  or  later
License  URI: http://www.gnu.org/licenses/gpl-2.0.html
Tags:  Blue
Version: 0.1
Text Domain: Learning WordPress
This theme, like WordPress, is licensed under the GPL.
Use it  to make something cool, have fun, and share what you've learned
with others.
*/
```

Figure: Existing theme

Style.css

Style.css template file is very crucial for a custom theme because styling the website completely relies on this file. For instance, if users want to change the colour of header or footer even layout of the page then they should work through the style.css file (16).



```
h1,body{
    font-family: "Roboto", sans-serif;

}

h1 {color:red;}
p {
    font-family: "Nanum Gothic", sans-serif;
    color:#0000ff;
}

ul {
    height:46px;
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #b2e224;
}
```

Figure: style.css file

Index.php

An index.php file should be created in the same folder. As soon as index file created, users can go to theWordPress dashboard and click on the appearance>> theme option to check whether a new custom theme does exist or not. Furthermore, users will be able to see default themes besides new custom theme.

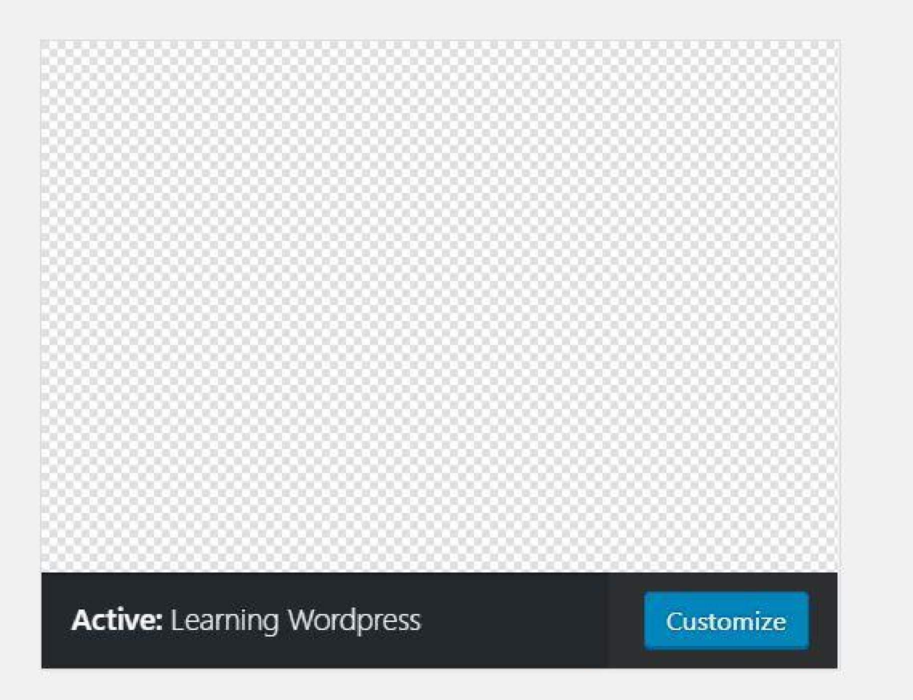
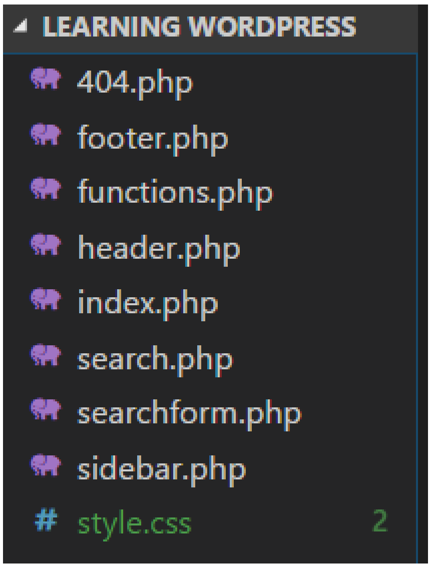


Figure: Activate the theme.

the new theme should be activated now and then need to go back to the main URL. The theme has been created already. Now it can be started working on developing process. This file contains code to maintain the main area of website and will give a direction to other files where to be included. Generally, index.php file in WordPress is used as a default file for rendering the website. If there are not any other existing file, WordPress will execute this file even if there is no file for type of content that is being asked for, WordPress will still fall back to the index file and execute it.

File Structure



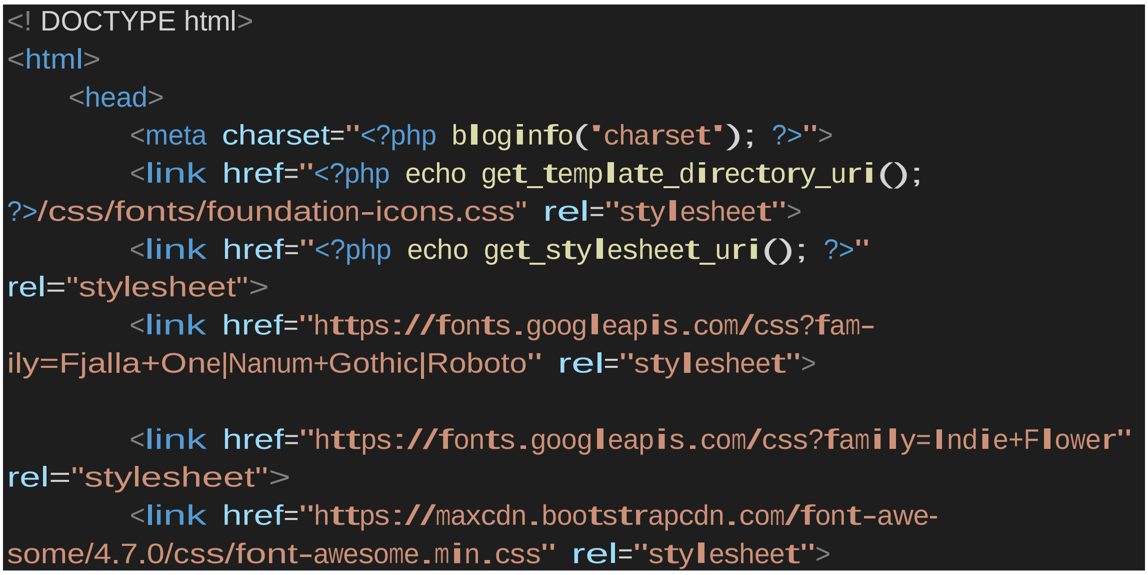
```
LEARNING WORDPRESS
  404.php
  footer.php
  functions.php
  header.php
  index.php
  search.php
  searchform.php
  sidebar.php
# style.css        2
```

Figure: File Structure

Our theme consists of template files , functions.php for adding functionality to wordpress and style.css for adding stylesheet to the theme .

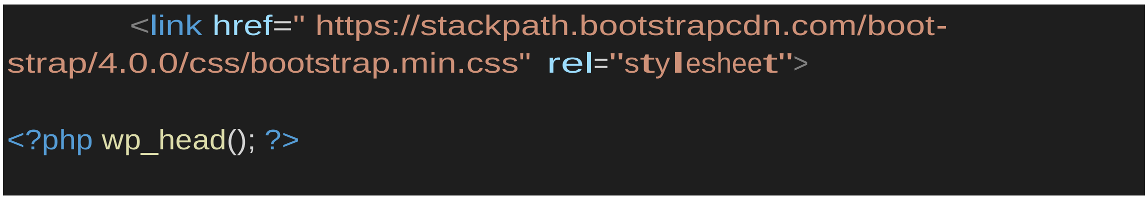
Header.php

Header file contains code for the header section of the theme. The header usually carries all the necessary styling for head section and top navigation to the website. The header file can include little content as much needed to be added to the theme. As an example



```
<! DOCTYPE html>
<html>
    <head>
        <meta charset="<?php bloginfo('charset'); ?>">
        <link href="<?php echo get_template_directory_uri();
?>/css/fonts/foundation-icons.css" rel="stylesheet">
        <link href="<?php echo get_stylesheet_uri(); ?>"
rel="stylesheet">
        <link href="https://fonts.googleapis.com/css?fam-
ily=Fjalla+One|Nanum+Gothic|Roboto" rel="stylesheet">

        <link href="https://fonts.googleapis.com/css?family=Indie+Flower"
rel="stylesheet">
        <link href="https://maxcdn.bootstrapcdn.com/font-awe-
some/4.7.0/css/font-awesome.min.css" rel="stylesheet">
```



```
        <link href=" https://stackpath.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"  rel="stylesheet">

<?php wp_head(); ?>
```

Figure: Header file in WordPress.

This is very straightforward example of header template file that will use throughout the developing process of a theme. It consists very simple <html> opening tag and a header block to display site logo (17).

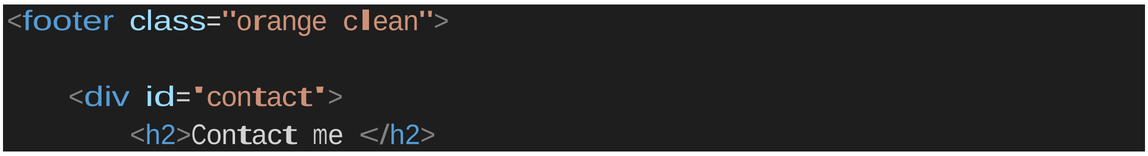
In the <head> section there is no style sheet is being declared because WordPress has a great way to deal with style sheet through wp_head() function.

wp_head() function

Initially wp_head() function is very useful and effective in the <head> section of all pages in a theme and it must be located before closing </header> tag. This function allows to include style sheet and script from a theme or plugins that might be used. If it is not added to the header file, developers may end up having difficulties to organize the whole website.

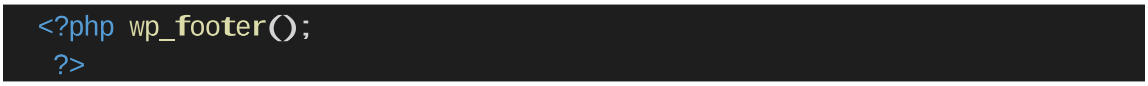
Footer.php

Footer.php file is very similar to the header.php file but it contains end part of the HTML content of a theme. Normally social icons, contact information, connecting links, additional text is in footer. The copyright information of a theme also located in the footer. It is such a place where a developer will likely find the </body> </html> pairing but it also could be used for common mark-up that located at the bottom section of every pages in a theme.



```
<footer class="orange clean">

    <div id='contact'>
        <h2>Contact me </h2>
```



```
 <?php wp_footer();
  ?>
```

Figure: wp_footer function.

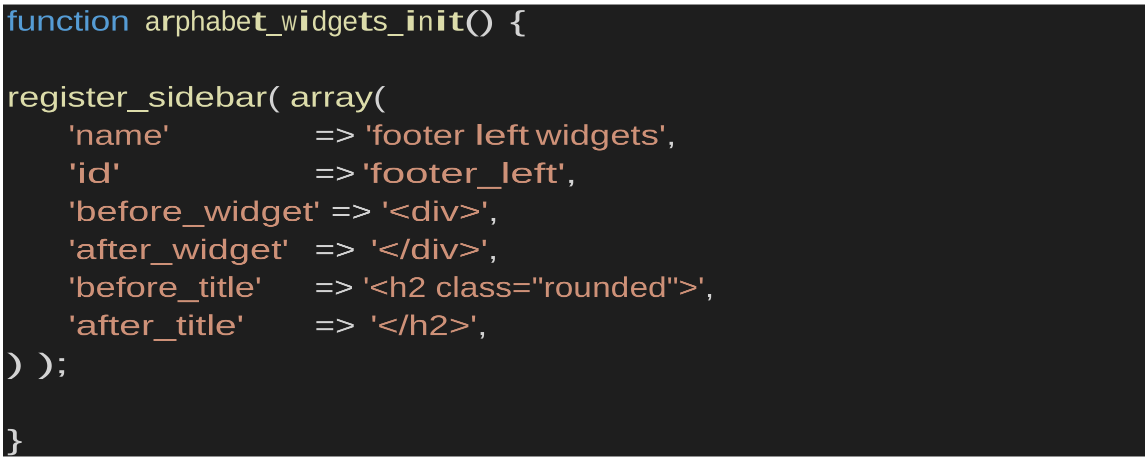
Wp_footer

It is a function of WordPress that applies when user want to change content of the footer area. The function does not work straightway to the theme, but it works through the users own code that intended to add for changing the content and then it flows the conventional way. It is called as a “hook” in WordPress. The function must be added before closing body tag otherwise hook will not work at all.

Sidebar.php

Sidebar is a basic template that can be used for content which appears on multiple pages in a website. It is not directly connected to the main page of the content, but it uses frequently to display widget which exist in WordPress by default or sidebar can be created by plugins or through the theme function file. Sidebar includes in a theme means user can add more content to the widget by customizer or widget panel area.

To use sidebars, users must register them in function.php file.



```
function arphabet_widgets_init() {

register_sidebar( array(
    'name'            => 'footer left widgets',
    'id'              => 'footer_left',
    'before_widget' => '<div>',
    'after_widget'  => '</div>',
    'before_title'    => '<h2 class="rounded">',
    'after_title'     => '</h2>',
) );

}
```

Including better and furnished files

Now WordPress developers can organise everything back in. Here is the new index.php file.

Developer can create better and well-furnished template files by WordPress default functions. Those functions help developer to reduce coding in the files that already being created. As we can see in the next line code

```
<?php get_header(); ?>
```

It will include template files header.php and its codes. As an example, if a website contains ten pages and header of the pages are like each other and then same code is being repeated each time which a lengthy process is quite and if there are some error it would be difficult to sort out. But WordPress has a unique way to handle it by using that get_header() function. This function pulls everything from the header.php file and make the main index file clean and light. Eventually we can create custom header as well by using this function (18) .

```
<?php get_footer();?>
```

It does the same as get_header function. Generally, a website contains same header, footer and sidebar for each page, if there are many pages existing. get_header (), get_footer(), get_ sidebar() these functions are there to do the trick for regular php files and they can be passed as a string in the parameter. In addition, by doing this, WordPress searches different files from the default to add to the theme.

```
<?php get_sidebar(); ?>
```

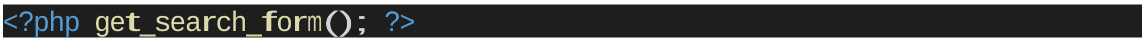
It can have multiple sidebars to use in various places in a theme to allow better functionality and structures.to create a new second sidebar, developer must create the template files called sidebar.secondary.php. to be able to include second sidebar in the template files, just call the get_sidebar () function wher e secondary will be string parameter.

Get_sidebar('secondary')

The same process can be applied in header and footer template files. It could be little odd to have multiple headers in a website. In addition, a client may ask developer, or a developer tend to make a website where two main section will exist. They can be linked together but display would be different with main navigation.

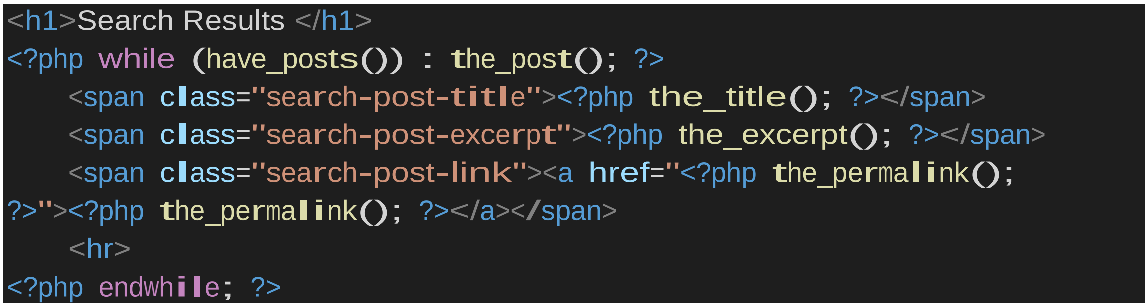
Search.php

Usually it is always better to create individual file for each section because it is easy to make changes or detecting error if error pops up. Generally, search option located in header area. To create search option in header, user can call the function in header.php



```
<?php get_search_form(); ?>
```

As an example if users want to see how many posts are available in the system, then WordPress uses loop function to find out posts through search option.

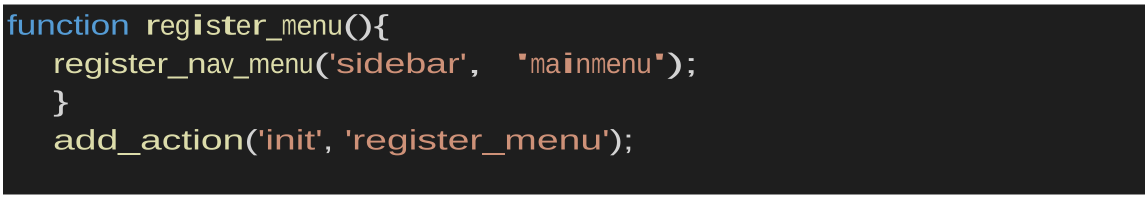


```
<h1>Search Results </h1>
<?php while (have_posts()) : the_post(); ?>
    <span class="search-post-title"><?php the_title(); ?></span>
    <span class="search-post-excerpt"><?php the_excerpt(); ?></span>
    <span class="search-post-link"><a href="<?php the_permalink();
?>"><?php the_permalink(); ?></a></span>
    <hr>
<?php endwhile; ?>
```

Figure: Searching post by looping.

Function.php

The function file is the root file for developing a theme. It acts like as a little plugin, adding features, functionalities, to a WordPress website through php code. It gets loaded automatically in front page and admin page of WordPress site. It contains most powerful codes of a WordPress theme. It is not possible to develop a theme without having function file in the WordPress. Basically, it works when current theme directory activated. This file contains many powerful function such as action hook functions and filter hook functions (19). Usually this file used to define classes, action, filter etc to be used by other templates of the theme. There are many functions available that bring data from WordPress to the theme by hooking. Such as



```
function register_menu(){
   register_nav_menu('sidebar',  'mainmenu');
   }
   add_action('init', 'register_menu');
```

Figure: Registering the menu.

The code registers the menu to the WordPress theme.

Technically the function files work as a hook all around.

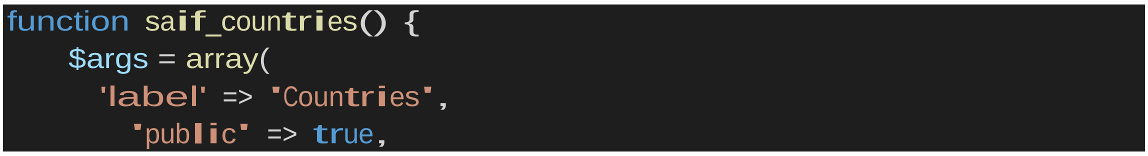
4.1 Custom Post types

Creating Custom post types manually

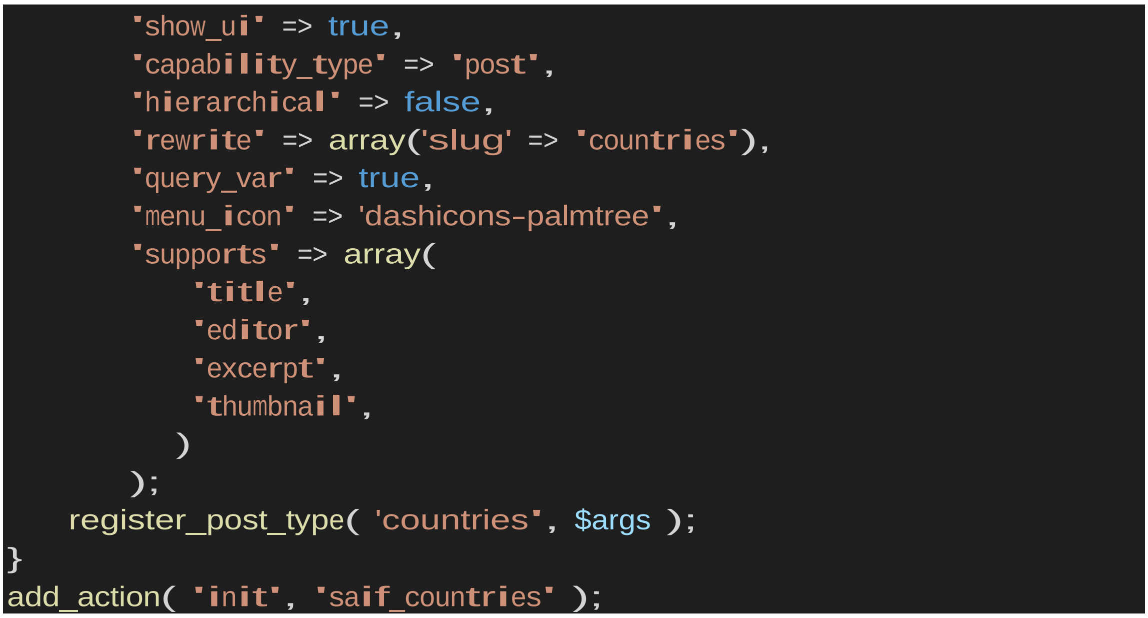
Custom post types could be any sort of content of a theme. WordPress has few custom post types by default. Those are post, page, attachment, revision and nav menu. Users can create their own custom post types and call them whatever they want to. For example, if user wants to visit listed countries probably user want to create a visit countries post type. The custom types can have different custom field and different custom category structure. Portfolio, blog, testimonial is post types (20).

There is a problem with custom post types when user creates custom post types using plugins. The problem is that when plugin is deactivated post types will be disappeared completely. Therefore, data will still be there of those custom post types, but they will be unregistered and inaccessible from the admin panel area.

First of all users need to register the register_post_type()function in the function.php file.



```
function saif_countries() {
    $args = array(
      'label' => 'Countries',
        'public' => true,
```

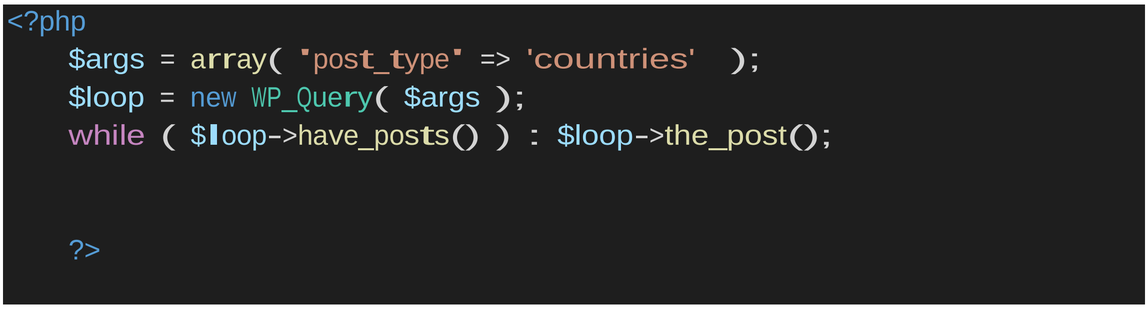


```
        'show_ui' => true,
        'capability_type' => 'post',
        'hierarchical' => false,
        'rewrite' => array('slug' => 'countries'),
        'query_var' => true,
        'menu_icon' => 'dashicons-palmtree',
        'supports' => array(
            'title',
            'editor',
            'excerpt',
            'thumbnail',
           )
        );
    register_post_type( 'countries', $args );
}
add_action( 'init', 'saif_countries' );
```

Figure: Custom post types.

The code does the trick that it registers the post types with an array of arguments. Those arguments are the main option of custom post types. this array has two parts. First one is label which is also an array. The second part carries other arguments like public_ show_ui,capability_types etc and those will be used in URL for this custom post type.

The advantage of creating custom post type is that it keeps away from the other regular post. Anyway, if developers want to display custom post type with other post in dashboard they should insert code in their main file.



```
<?php
    $args = array( 'post_type' => 'countries'  );
    $loop = new WP_Query( $args );
    while ( $loop->have_posts() ) : $loop->the_post();


    ?>
```

As soon as page loaded, dashboard will be like this. By clicking countries users will be able add countries in the custom field.

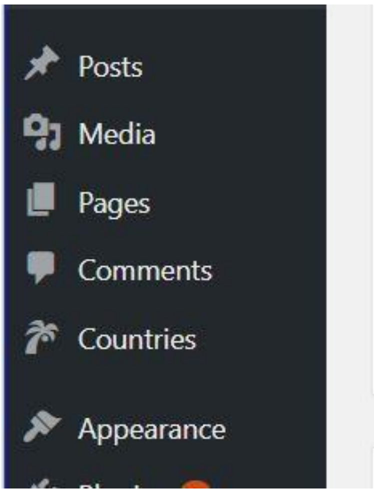


Figure: Custom post Countries.

4.2 Customization API

Developing customizer area

To begin with, developers should have at least two hooks to add option to the customizer.

1.customize_register

2.wp_head.

customize _register

Theme customize_register action hook used to define new sections, setting and control. Hence, this function brings dynamic option to the customization admin section where users can add dynamic contents to the website.

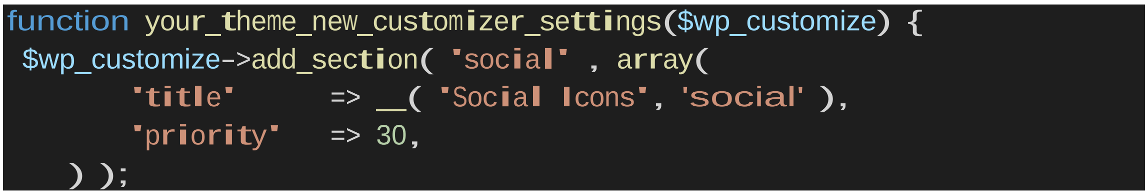
wp_head

This hook will allow to outside resources to insert additional code for adding extra functionalities in WordPress. For example, users can add additional css stylesheet to the theme. It allows to the developers' certain way to put their own code in specific place and make it work. The main objective of doing hook in WordPress is to provide custom post, contents, etc to the website. As an example, if developers want to use their own code to the WordPress blog or any piece of content, WordPress has certain function to hook with developers code that will go through the developers code and resume back to the normal flow.

Adding a new setting

To add a new setting in customizer, user need to call the $wp_customize->add_setting() method to the function template file. Users don't have to call the action hook Wpmu_customize_register() function in function template file because when the object $wp_customize will be added to the file, register action hook will be loaded automatically.

After saving the file user can go to the dashboard and click on customize option and user will not be able to see a section "social icon" in the customization area because if there is no controls inside the sections, WordPress will not display the section in customize panel (21).



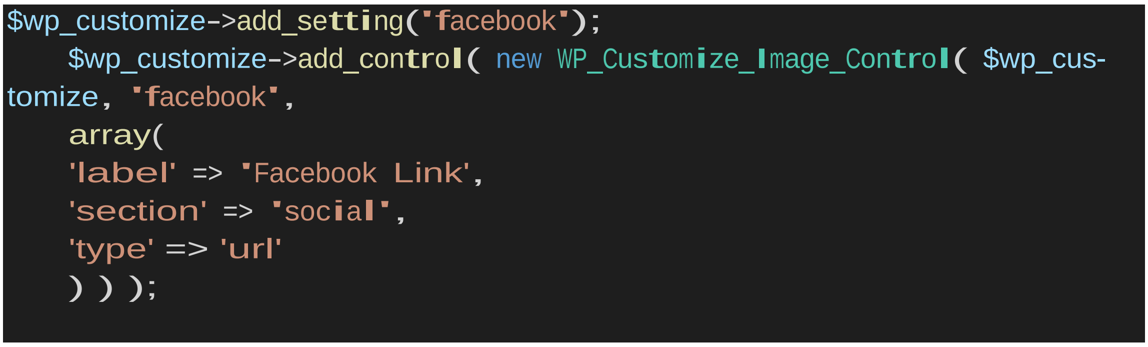
```
function your_theme_new_customizer_settings($wp_customize) {
 $wp_customize->add_section( 'social' , array(
       'title'      => __( 'Social Icons', 'social' ),
       'priority'   => 30,
    ) );
```

Figure: Social Icon section

In addition, users should specify the title of the section in the code to display what they want to be there in WordPress customize panel.

Adding a new control

Once control has been inserted to the section under same function, the user will be able see social icon section displayed in customize area of WordPress.



```
$wp_customize->add_setting('facebook');
    $wp_customize->add_control( new WP_Customize_Image_Control( $wp_cus-
tomize, 'facebook',
    array(
    'label' => 'Facebook Link',
    'section' => 'social',
    'type' => 'url'
    ) ) );
```

Figure:Add control in the section.

After executing the code the output will be like that.

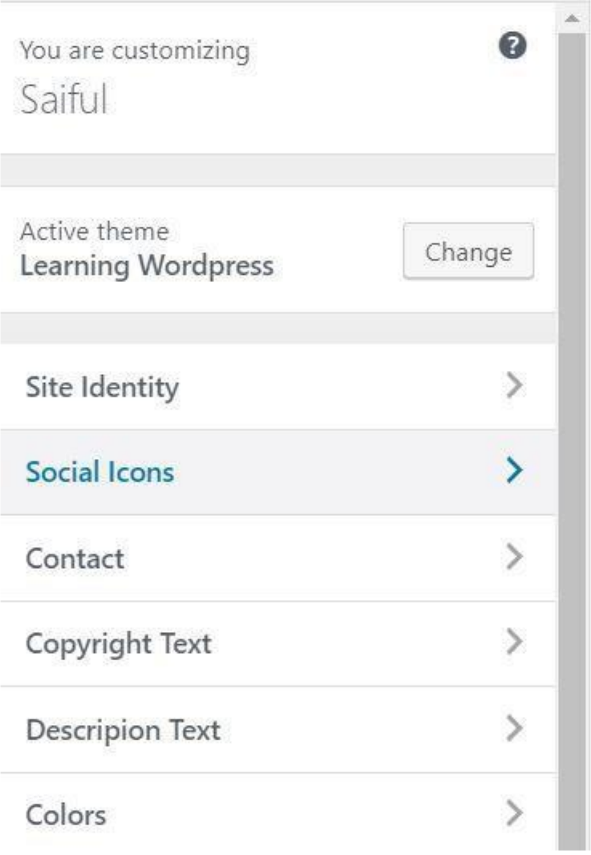


Figure: Social icons added to the customize.

As back-end set up now, developer can use get_theme_mode function to return numerical function value in customizer control. Basically, it does help to connect front-end with backend. As an example, when user call the function in front-end it will connect through the same parameter that has been called earlier in back-end.

For instance, something to be able to display and interact with it dynamically in the website, users must follow three steps in Funtion.php file.

Add new setting and section.

```
function your_theme_new_customizer_settings($wp_customize) {
    // add a setting for the site logo


    $wp_customize->add_section( 'social' , array(
        'title'      => __( 'Social Icons', 'social' ),
        'priority'   => 30,
    ) );
```

2. add new setting and section.

```
$wp_customize->add_setting('facebook');
    // Add  a  control to upload the  logo
    $wp_customize->add_control( new WP_Customize_Image_Control( $wp_customize, 'facebook',
    array(
    'label' => 'Facebook Link',
    'section' => 'social',
    'type' => 'url'
    ) ) );
```

3. specified the contents where to be displayed on the page in terms of functions

```
<div class="social">
<a href="<?php echo get_theme_mod( 'facebook' ); ?>" class="fa fa-facebook"></a>
<a href="<?php echo get_theme_mod( 'twitter' ); ?>" class="fa fa-twitter"></a>
<a href="<?php echo get_theme_mod( 'google' ); ?>" class="fa fa-google"></a>
```

As soon as page loaded, user will get to see the output on the page.

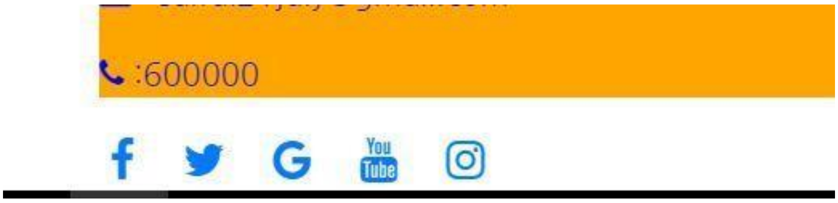


Figure: Social icons as an output.

exactly following the same way users can add any dynamic content to anywhere in the website.

Adding custom logo support to the theme

At first users must enable the use of custom logo support in the function.php file.

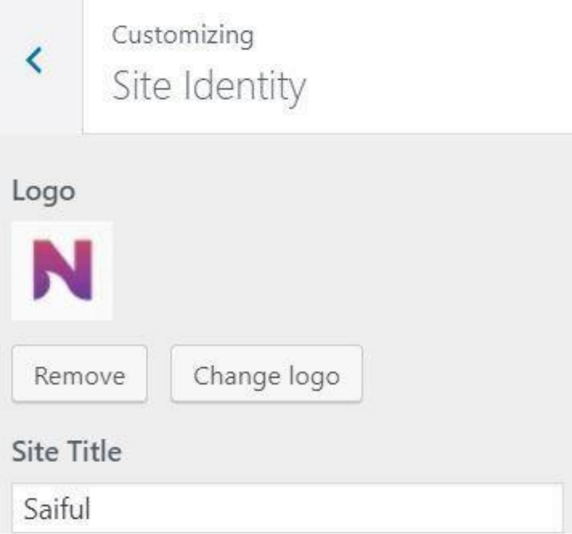
```
add_theme_support( 'custom-logo' );
```

after enabling custom logo support user can modify those five parameters by passing arguments to the Add_theme_support() using Array

```
function themename_custom_logo_setup() {
    $defaults = array(
        'height'      => 50,
        'width'       => 50,
        'flex-height' => true,
        'flex-width'   => true,
        'header-text' =>add_theme_support( 'custom-logo' ); array( 'site-
title', 'site-description' ),
    );
    add_theme_support( 'custom-logo', $defaults );
}
add_action( 'after_setup_theme', 'themename_custom_logo_setup' );
```

Figure: Custom logo setup

It’s already visible in the customize area. If users click on site identity they will see logo option in the customizer.



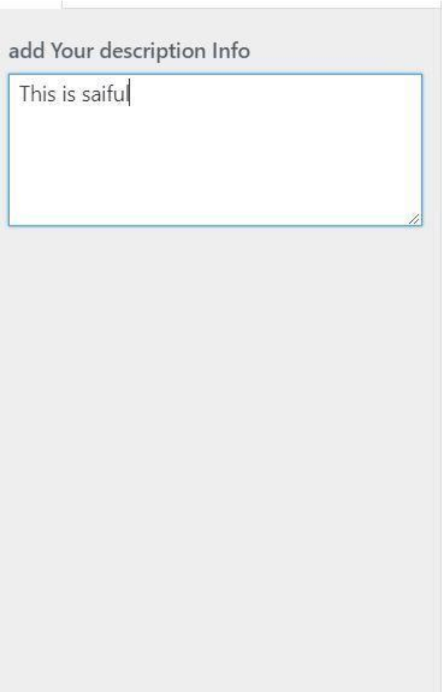
Site Title: Saiful

Figure:Custom logo in customize panel area.

As custom theme has been developed, the customization process followed according to theme customization API framework which is available in codex. Codex is a platform that provides anything related to theme development and customization.

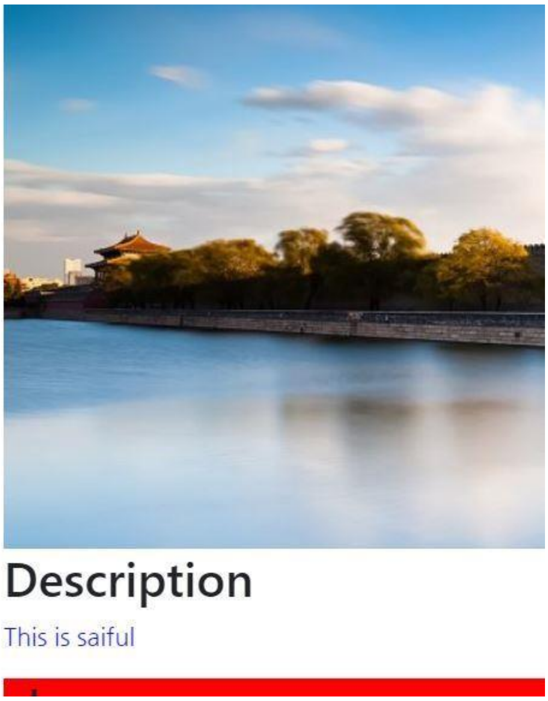
After customization, the website contains more dynamic option which means users can add or delete content by using customize panel. For example, in the header “logo” can be editable by using logo option.

In addition, the description of the content can be replaced by using customize panel.



add Your description Info

This is saiful

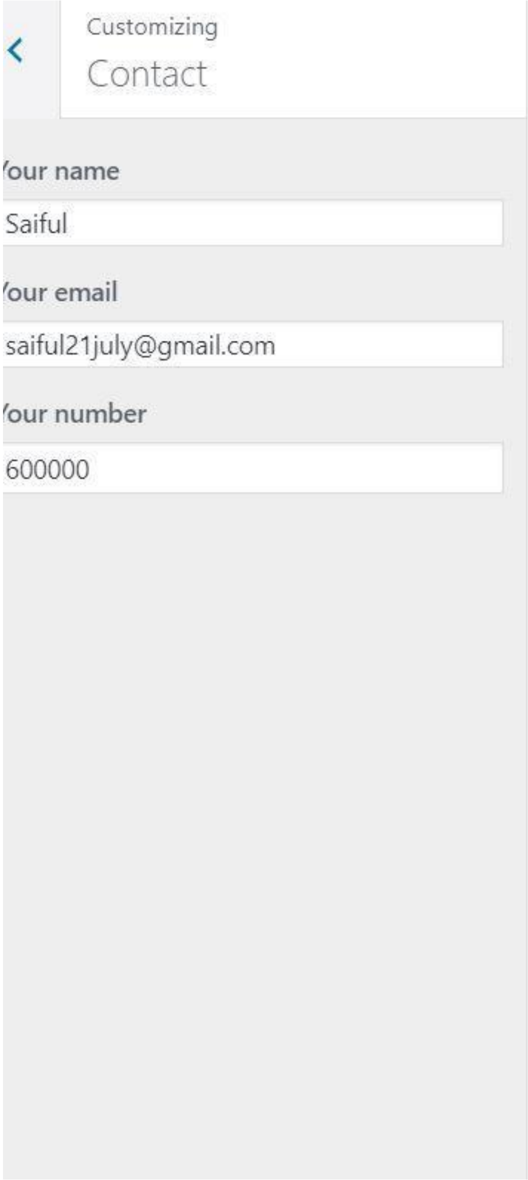


Description

This is saiful

Figure: Dynamic Description.

In the footer section the contact information can be modified by something else dynamically. basically, in the footer section social icons are interactive, copyright option is editable.

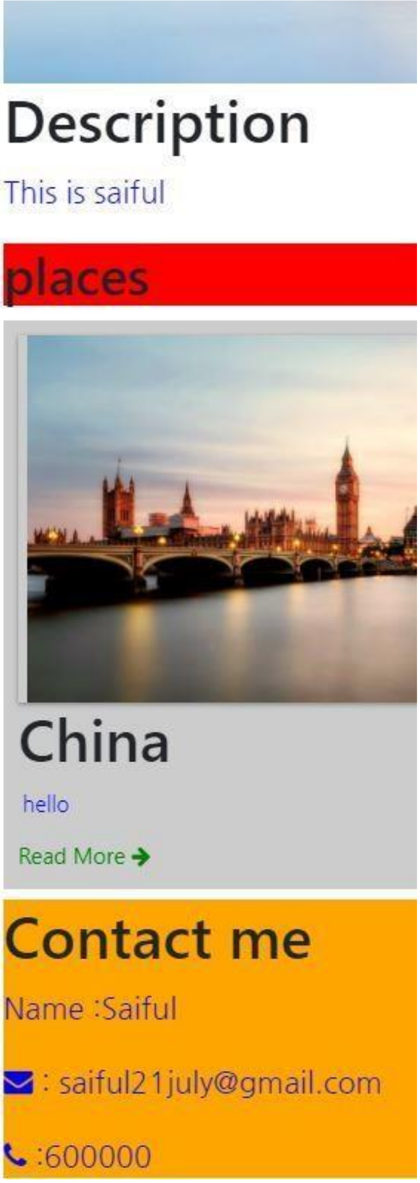


<
Customizing
Contact

/our name
Saiful

/our email
saiful21july@gmail.com

/our number
600000



Description

This is saiful

places

China

hello

Read More ➔

Contact me

Name :Saiful

✉ : saiful21july@gmail.com

☎ :600000

5 Discussion

In this chapter of learning, a brief evaluation will be discussed. the intention of doing practical work was to gain knowledge about core concept of WordPress architecture and how every single functions and features work together as a whole, how WordPress different from other platform to develop a website. The reason behind choosing this topic is to learn WordPress in advanced level, technical aspects of theme development process, how to create an interactive website etc.

However, during the learning process, it was found that WordPress is less time-consuming platform compared to others because in WordPress thousands of default functionalities, plugins, blogs, posts, pages are available to cooperate with theme making process. In addition, default function makes WordPress to be perfect when it comes to organise the contents into order. Making website in other platform may take a longer time to establish a structure of website.

The main idea behind custom theme development from thescratch is applying personal ideas to create something new using WordPress platform and at the same time implement it without much coding.

As a content management system WordPress has been proved successfully since big companies, users, clients, developers, students, adopted for its development. consequently, WordPress holds million users and developers to develop it further

6 Conclusion

The main purpose of the thesis work was to develop a theme and make it customizable based on WordPress content management system(CMS). The WordPress community has been developing consistently the WordPress last few years for its users, developers, companies and clients. Over the last few years the WordPress has been received significant amount of popularity for its simplicity, ability to create a website within brief period of time, in web development industry. Genuinely it has attracted to the passionate WordPress community to ensure it stays flexible, reliable, updated, secure etc. The WordPress community has already created thousands of themes, plugins to extend the platform.

To an extent, a clear instruction has been elaborated how to develop a theme from the scratch and make that theme customizable with customization API framework.

Nevertheless, developing a website and customize it on WordPress would be great initiative to enrich knowledge and practical skills. Therefore, a solid custom theme has been built already throughout the developing process.

Finally, it can be said that theme development customization is certainly worth for experiencing such a dynamic way of developing website.

7 References

1. wpbeginne. http://www.wpbeginner.com. [Online].; 2015 [cited 2018 3 15. Available from: http://www.wpbeginner.com/why-you-should-use-wordpress/.
2. wp101. https://www.wp101.com. [Online].; 2018 [cited 2018 4 12. Available from: https://www.wp101.com/tutorial/what-is-wordpress/.
3. Silverman M. https://thoughts.duoconsulting.com. [Online].; 2017 [cited 2018 4 16. Available from: https://thoughts.duoconsulting.com/blog/5-reasons-to-use-drupal-vs.-wordpress.
4. Joomla. https://www.joomla.org. [Online].; 2005 [cited 2018 4 16. Available from: https://www.joomla.org/about-joomla.html.
5. Onishi A. Pro Wordpress Theme development. 1st ed.: Apress; 2013.
6. codex.wordpress.org. https://codex.wordpress.org. [Online].; 2018 [cited 2018 4 19. Available from: https://codex.wordpress.org/Templates.
7. wordpress.org. https://developer.wordpress.org. [Online].; 2018 [cited 2018 4 18. Available from: https://developer.wordpress.org/themes/basics/template-hierarchy/.
8. Rascia T. https://www.taniarascia.com. [Online].; 2015 [cited 2018 4 10. Available from: https://www.taniarascia.com/developing-a-wordpress-theme-from-scratch/.
9. wpbeginne. http://www.wpbeginner.com. [Online].; 2018 [cited 2018 4 17. Available from: http://www.wpbeginner.com/glossary/hooks/.
10. codex.wordpress.org. https://codex.wordpress.org. [Online].; 2017 [cited 2018 4 17. Available from: https://codex.wordpress.org/WordPress_API%27s.
11. codex.wordpress.org. https://codex.wordpress.org. [Online].; 2016 [cited 2018 4 16. Available from: https://codex.wordpress.org/Theme_Customization_API.
12. wpblogx. https://www.wpblogx.com. [Online].; 2017 [cited 2018 3 25. Available from: https://www.wpblogx.com/what-is-xampp/.
13. techterms. https://techterms.com. [Online].; 2018 [cited 2018 3 5. Available from: https://techterms.com/definition/css.

14. wordpress.org. https://codex.wordpress.org. [Online].; 2009 [cited 2018 3 4. Available from: https://codex.wordpress.org/Installing_WordPress.

15. wordpress.org. https://codex.wordpress.org. [Online].; 2010 [cited 2018 4 5. Available from: https://codex.wordpress.org/Theme_Development.

16. Onishi A. Pro Wordpress theme development. 1st ed.: Apress; 2013.

17. ONISHI A. Pro wordpress theme development. 1st ed.: Apress; 2013.

18. Onishi A. Pro Wordpress theme development. 1st ed.: Apress; 2013.

19. wordpress.org. https://codex.wordpress.org. [Online].; 2011 [cited 2018 4 5. Available from: https://codex.wordpress.org/Installing_WordPress.

20. Onishi A. Pro Wordpress theme development. 1st ed.: Apress; 2013.

21. wordpress.org. https://codex.wordpress.org/. [Online].; 2012 [cited 2018 4 4. Available from: https://codex.wordpress.org/Theme_Customization_API.