Wordpress Plugins for Symbiota2

Venkatesh Kadali
Utah State University

WORDPRESS PLUGINS FOR SYMBIOTA2

by

Venkatesh Kadali

A report submitted in partial fulfillment of the requirements for the degree

of

Master of Computer Science

Approved:

Dr. Curtis Dyreson
Major Professor

Dr. Amanda Lee Hughes
Committee Member

Dr. Douglas Galarus
Committee Member

UTAH STATE UNIVERSITY
Logan, Utah

2018

ABSTRACT

WordPress plugins for Symbiota2

by

Venkatesh Kadali, Master of Computer Science
Utah State University, 2018

Major Professor: Dr. Curtis Dyreson
Department: Computer Science

Symbiota is an open source project which helps biologists concentrate their efforts on curation of quality datasets rather than creating the tools. Symbiota's graphical user interface is hardcoded into many PHP files resulting in poor maintainability and extensibility. The main goal of Symbiota is to separate the user interface of Symbiota with its core structure as the user interface is hard-coded with PHP. This helps in reducing the source lines of code. The secondary goal of Symbiota is to help Symbiota developers to add the Symbiota features to their website and also should identify whether the separation of user interface with core structure is feasible. This project is completely related to second goal. The content management in the second goal can be done by using a free and open source content management system called WordPress.

WordPress plugins help WordPress developers to add features to their website without knowing a single line of code. In this project, two WordPress plugins were developed to help Symbiota2 developers to easily add Symbiota features to their website without knowing anything about coding. Symbiota Search is the first WordPress plugin developed in this project. It helps WordPress developers to add Symbiota search feature. Symbiota Validation is second plugin developed in this project. It helps WordPress developers to add the sign in and sign up features of Symbiota to their website.

(54 Pages)

ACKNOWLEDGEMENTS

To my parents Satyanarayana Kadali, Naga Suneetha Kadali and my sister, Pavani Keerthi Kadali, who have been my great strength and support to me all through. I would like to express my sincere gratitude and deep appreciation to my Esteemed Major Professor, Dr. Curtis Dyreson, for providing me constant support, active guidance, comments and valuable suggestions throughout the course of this project to reach a successful completion.

I extend my gratitude to Utah State University for giving me this opportunity.

Thanks to Dr. Douglas Galarus and Dr. Amanda Lee Hughes for being on my Committee.

Venkatesh Kadali

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Symbiota has been greatly expanded, with many projects funded by US National Digitization Program using Symbiota since 2012 [11]. The central premise of this open source software project is that better quality of biodiversity portals can be built through a relationship between the software engineers and the scientific community.

The Symbiota project [2] is programmed using many PHP files. A file has a functionality to query the data, filter the data, manipulate the data and perform other functions. The user interface of Symbiota is hardcoded in PHP. The code can be reduced in size and complexity by cleanly separating the graphical user interface from the core functionality. This separation is beneficial for maintainability and extensibility. The main goal of this project is to check whether the separation of Symbiota's user interface with its core structure is feasible and help Symbiota developers add the Symbiota features to their website.

This content management can be done by adding a Model, View, Controller (MVC) framework to Symbiota. A free and open source content management system (CMS), such as WordPress, makes it comparatively easy for users to create and manage their content with minimal technical know-how.

The integration of CMS with Symbiota will be beneficial by:

1) Making the process easy for the user to change the site design. The user can change the design while keeping the site functional because the content and design are in separate virtual boxes.

2) Supporting Internationalization so that projects can easily be adapted to a specific local languages and cultures.

3) Improving site maintenance by updating the CMS software automatically and adding additional functionality without breaking the site.

4) Improving the user experience and flexibility by providing responsive layouts.

5) Facilitating WYSWIG editing – allowing a developer to see what the result will look like while the interface is being created.

6) Enabling social media syndication – helping the website to get more exposure by distributing the content on high authority social media sites.

This project demonstrates two WordPress Plugins - Symbiota Search Engine WordPress plugin and Symbiota Validation WordPress plugin. These two plugins help WordPress developers add several Symbiota features to their website. Symbiota Search Engine WordPress plugin is a Read only plugin which helps in getting the search results from the Symbiota repositories. Symbiota Validation WordPress plugin is a read and write plugin which helps with sign in and sign up validations.

This paper presets - Symbiota search engine overview in Chapter 2, WordPress and WordPress plugins overview in Chapter 3, Symbiota search WordPress plugin architecture in Chapter 4, Symbiota validation WordPress plugin architecture in Chapter 5 and Summary in Chapter 6. The installations required for this project are explained in the Appendices.

1.1 Background

More than 1 billion U.S. specimens are widely used in scientific research, land management decision making and education. Information associated with those specimens and collated into large data products can allow wider use of this massive body knowledge to answer new research questions. However, digitizing this information is a massive undertaking that needs large-scale financial and technical support [14] [15]. The Symbiota Software Project was developed for networking biodiversity data with an emphasis on the software design principles and functionality. With an estimated total of more than 350 participating collections, 2500 active data providers and 12+ million digitized specimen records, Symbiota is a leading open source platform in North America for mobilizing, integrating, and using specimen based and observation based occurrence records and derivative products [4].

The Symbiota website [13] describes the Symbiota project as follows:

"The Symbiota Software project is functioning on building a library of webtools to help biologists in demonstrating specimen based virtual flora and fauna. The central premise of this open source software project is that, through a partnership between software engines and the scientific community, higher quality and more publicly useful biodiversity portals can be built. An open source software framework allows the technicians to create the tools, thus freeing the biologist to concentrate their efforts on the curation of quality datasets."

The main goal of Symbiota is to separate the User Interface of Symbiota with its core structure as the User Interface is hard-coded with PHP. This helps in reducing the huge source lines of code. The second goal of Symbiota is to help Symbiota developers add the Symbiota features to their website. The second goal should also answer a question whether the separation of User Interface with core structure is feasible. This project is completely related to the second goal.

The Symbiota project is programmed into myriad PHP files [5] which queries the data, filters the data, manipulates the data and perform several other functions [3]. PHP is a very popular programing language that has become very popular since its development in 1994. With the way the PHP programming language was created, it is difficult to use it to program huge applications. Since the programming language is not highly modular, huge applications created out of the programming language will be difficult to maintain. As the Symbiota project is becoming huge day by day, it is very difficult to maintain. Reducing the size of code would be beneficial for maintainability and extensibility. The code can be reduced in size and complexity by cleanly separating the graphical user interface from the core functionality. Content management can be done by adding an MVC framework to Symbiota.

WordPress is the most popular self-hosted website software in use today. It is available as an open source project, and built largely on the top of the MySQL database and PHP programming language. WordPress is the publishing mechanism behind thousands of individual blogs and the engine that powers high-volume, high-profile sites such as CNN's website and blogs [6] [7]. It was designed for anyone comfortable navigating a browser, but is accessible to web designers and developers as well. One of the main reasons WordPress is such a popular platform is the ease with which it can be extended. Plugins are the primary reason for this and allow endless possibilities in extending WordPress [17].

This project is completely built in WordPress as WordPress plugins. This project consists of two WordPress plugins. Symbiota Search WordPress Plugin is the first plugin designed to help WordPress developers to add a Symbiota specimen search engine feature to their own WordPress website. One of the central features of Symbiota is its ability to integrate multiple collections within a single interface. The Symbiota specimen search engine enables collections to be queried by taxonomic, geographic, and collector details. The second plugin is Symbiota validation WordPress plugin which helps the WordPress developers to add the Symbiota validation forms which are connected to the Symbiota database into their WordPress websites.

These two WordPress plugins have different Symbiota functionalities which can be used in any WordPress website. The User Interface of Symbiota is separated with the core structure in this project so that the WordPress developer can easily add the Symbiota's features into their website. This becomes a proof-of-concept for the first goal of Symbiota to separate the user interface with the core structure. This clearly demonstrates that the separation of Symbiota's user interface with Symbiota's core structure is feasible.

CHAPTER 2

SYMBIOTA SEARCH ENGINE AND SYMBIOTA VALIDATION OVERVIEW

2.1 Specimen Search Engine Architecture

The Symbiota specimen search engine [1] shown in Figure 2.1 is comprised of four parts

1) Taxonomic Criteria: A user can search for records by family name, common name, scientific name, family or scientific name and, higher taxonomy in a taxonomical criterion. Different collection managers have different taxonomic opinions of how to manage collections so it is a necessary to establish a method of searching for synonyms. To resolve this problem an option called taxonomic thesaurus search is available.
2) Geographic Criteria: A user can search in two combinations geographically. The first combination is a collection of country, state, county and locality which returns all the records containing words matching the request. The second combination is a collection of latitude and longitude coordinates that searches using latitude and longitude coordinates which returns specimens that have matching geo reference points.
3) Collector Criteria: The Collector criteria is a combination of Collector's last name, number and date. The queries including these text searches will return all the records containing words matching the request.
4) Specimen Criteria: A user can search according to the catalog number by limiting to type specimens or by limiting to specimens with images or by limiting to specimens with genetic data.

2.2 Specimen Search Engine Features

- Results are displayed in a concise list by default. Complete collection details including images can be accessed through a link.

- Clicking on the scientific name will display the profile page of species, which will display details such as synonyms, common names, taxon authors, distribution maps, and all images relevant to the scientific name.

- One can view the distribution of specimens that have been georeferenced by clicking on the map tab.

- Distribution maps can be viewed using Google Maps or Google Earth.

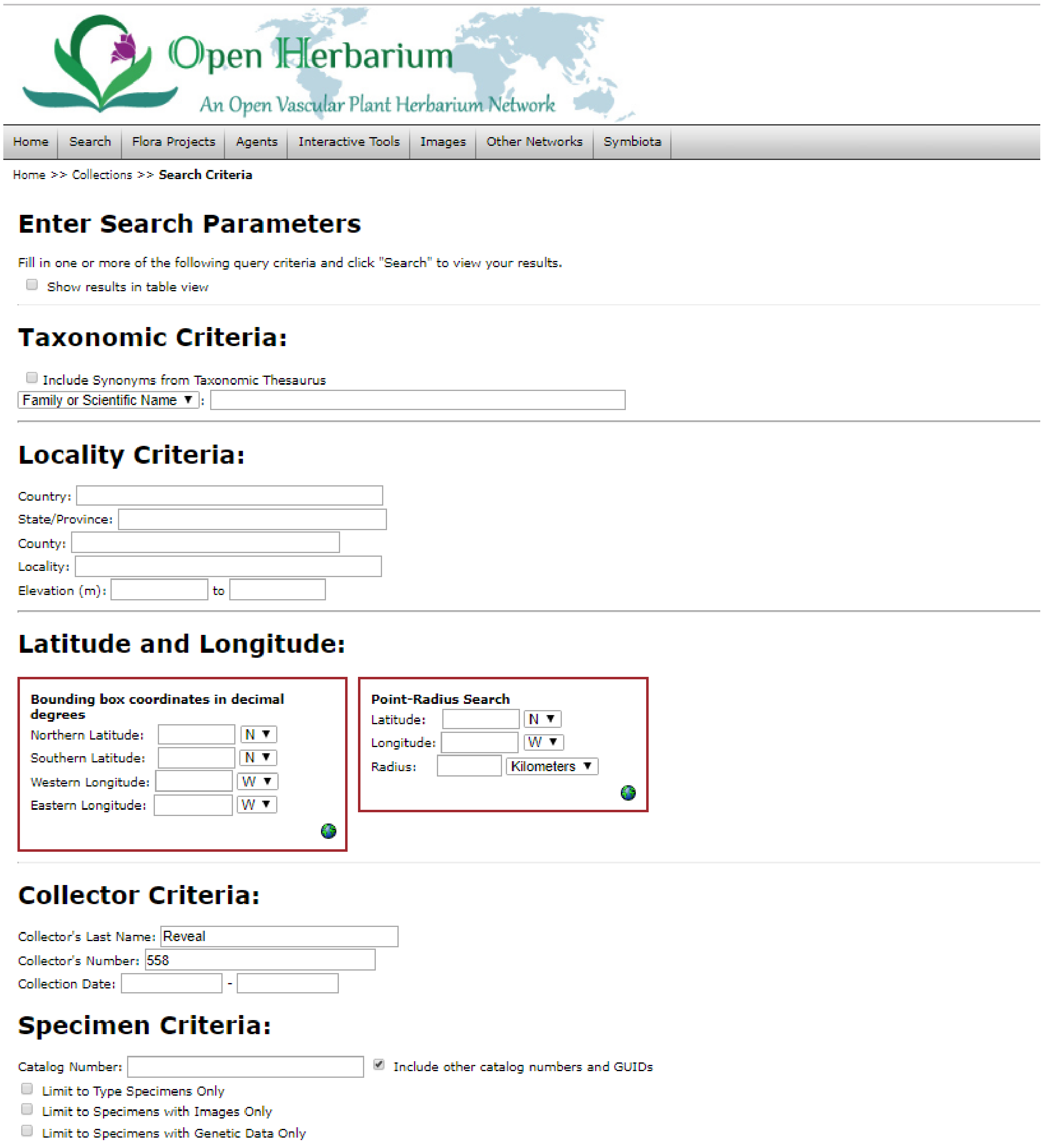
- Clicking on a specimen marker will display full collection details for that specimen.
- By selecting the checklist tab, results can also be compiled as a unique species list.
- The species list is displayed based on the taxonomy of how the specimens are treated within the collections searched by default.
- The Taxonomic Filter is used to run the search results through the taxonomic thesaurus and produce a list consisting of a single taxonomic option.

The Specimen Search Engine is one of the Symbiota modules that needs to be refactored into WordPress plugins to help developers to add the search functionality into their websites. The main goal of this project is to refactor Symbiota into plugin modules that have cohesive functionality. Refactoring Symbiota's functionality into plugins will create an extensible framework for development.

2.3 Symbiota Validation Architecture

The Symbiota validation page consists of required fields and optional fields. The required fields are necessary to create a new user login and the optional fields are to collect more information about the user which are nonmandatory. The required fields consist of user name, password confirmation, complete name and email address details. The optional fields consist of information regarding the address and Institution of the user. All these details will be stored in a Symbiota's database whenever the user click "Create Login" button.

Symbiota Validation is another module which needs to be refactored into WordPress plugins to help developers to add the validation feature into their WordPress websites. The main feature of this plugin is that it will write the data into Symbiota database.


Open Herbarium
An Open Vascular Plant Herbarium Network
Home | Search | Flora Projects | Agents | Interactive Tools | Images | Other Networks | Symbiota
Home >> Collections >> Search Criteria

Enter Search Parameters

Fill in one or more of the following query criteria and click "Search" to view your results.

☐ Show results in table view

Taxonomic Criteria:

☐ Include Synonyms from Taxonomic Thesaurus

Family or Scientific Name ▼ :

Locality Criteria:

Country:
State/Province:
County:
Locality:
Elevation (m): to

Latitude and Longitude:

Bounding box coordinates in decimal degrees

Northern Latitude: N ▼
Southern Latitude: N ▼
Western Longitude: W ▼
Eastern Longitude: W ▼

Point-Radius Search

Latitude: N ▼
Longitude: W ▼
Radius: Kilometers ▼

Collector Criteria:

Collector's Last Name: Reveal
Collector's Number: 558
Collection Date: -

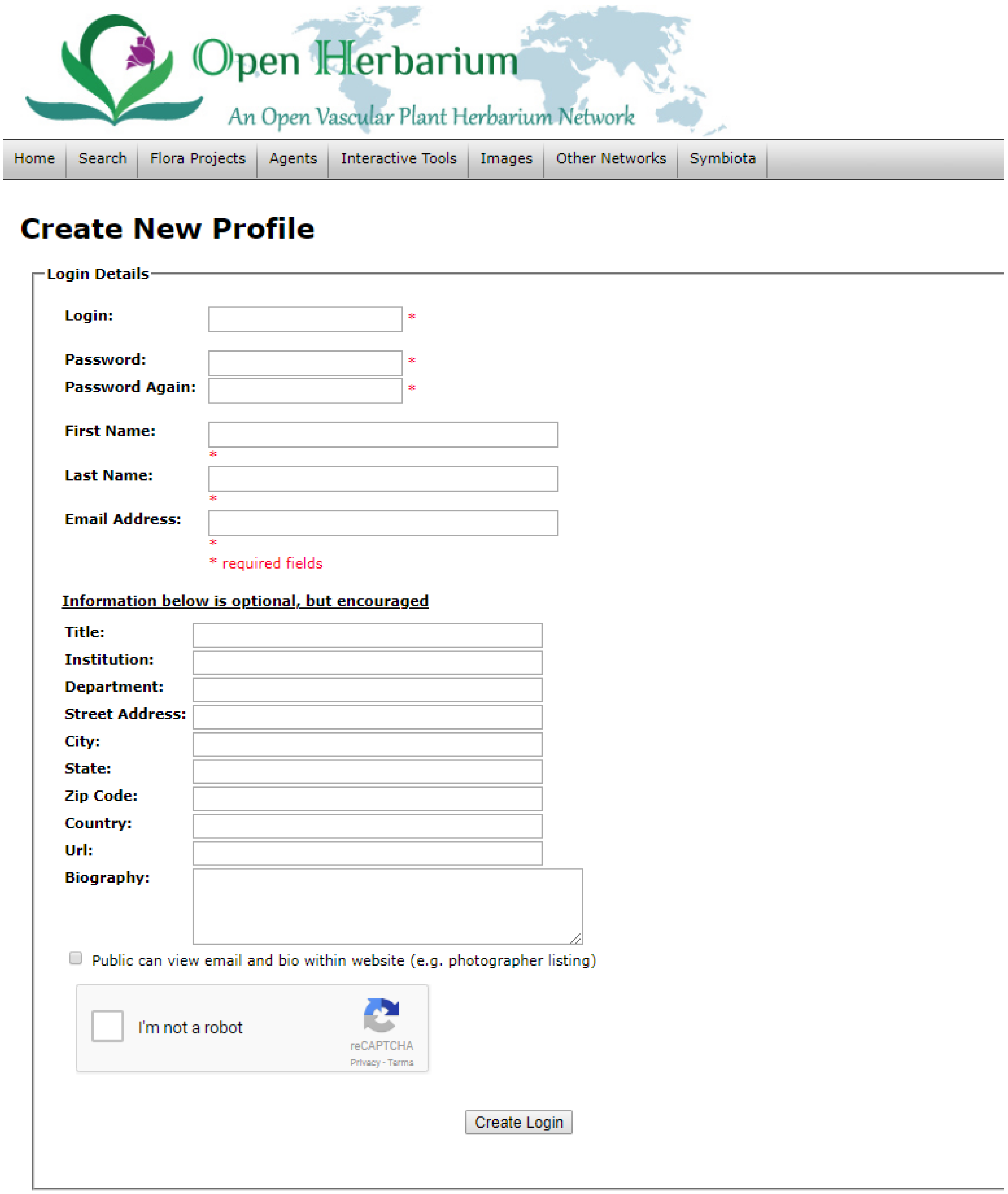
Specimen Criteria:

Catalog Number: ☑ Include other catalog numbers and GUIDs

☐ Limit to Type Specimens Only
☐ Limit to Specimens with Images Only
☐ Limit to Specimens with Genetic Data Only

Figure 2.1: Symbiota specimen search engine comprising four divisions

url: http://openherbarium.org/portal/collections/harvestparams.php



Create New Profile

Login Details

Login:	*
Password:	*
Password Again:	*
First Name:	*
Last Name:	*
Email Address:	*

* required fields

Information below is optional, but encouraged

Title:
Institution:
Department:
Street Address:
City:
State:
Zip Code:
Country:
Url:
Biography:

Public can view email and bio within website (e.g. photographer listing)

Figure 2.2: Symbiota form validation page to register a new user

url: http://openherbarium.org/portal/profile/newprofile.php

CHAPTER 3

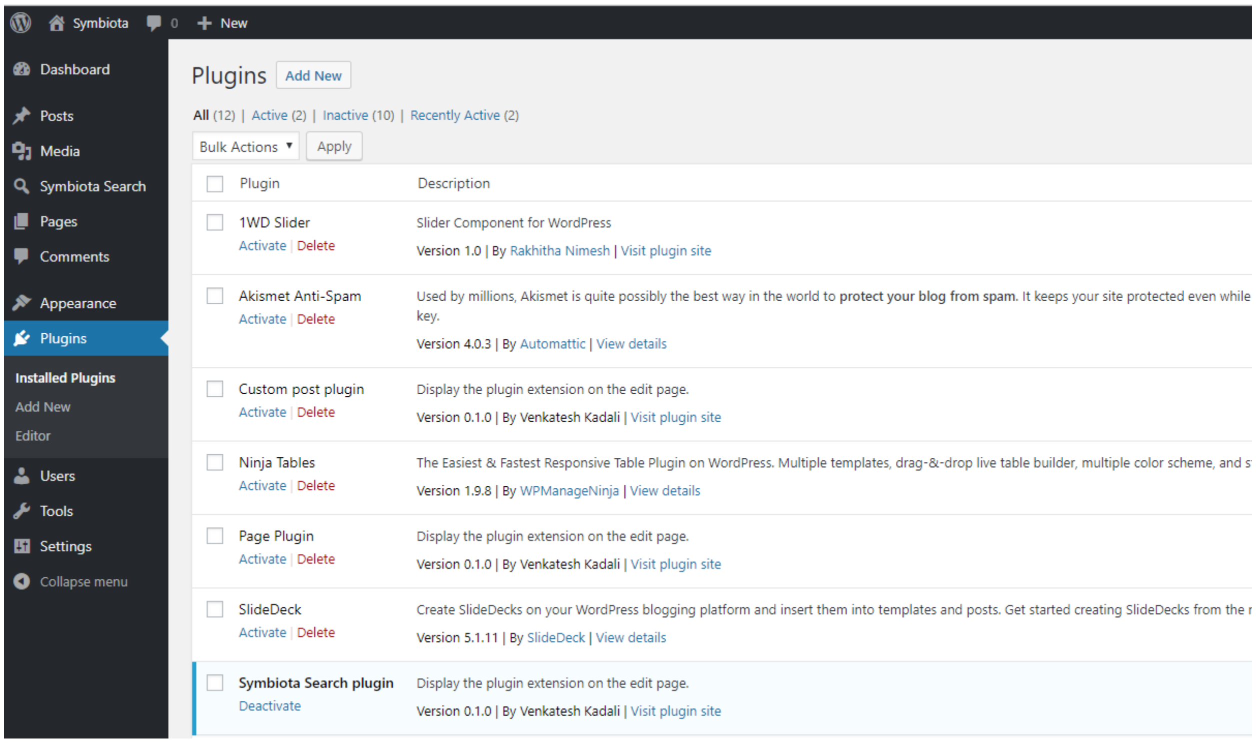
WORDPRESS OVERVIEW

3.1 WordPress

WordPress is a highly popular free open-source content management system used to power many websites, blogs, home pages and social communities. It is based on PHP and MYSQL. There are over 29.4% of the top 10 million websites using WordPress as of January 2018 [16] [17]. WordPress support more than 60 million websites, making it statistically the most popular website management system in use on Web. WordPress allows users to share and create blogs, home pages or websites with content ranging from handcrafted, personal anecdotes to world-changing movements. Inexperienced tech people can also use it very easily and, tech experienced people can customize it in several ways.

3.2 WordPress Plugins

WordPress plugins extend the functionality or add new features to WordPress websites. They make it easy for users to add features to their website without knowing the single line of code. There is a saying that goes around the WordPress community: “there’s a plugin for that”. The vast number of plugins is one of the reasons for the popularity of WordPress. At this time of writing, there are over 50,316 plugins in the official WordPress plugin directory. The plugins are frequently developed by individual developers offering custom functions and features that allow users to adjust their sites to their specific needs. This customization ranges from content display features, to search engine optimization, content management systems, and the addition of widgets and navigation bars. There are plugins for social media syndication, picture galleries, and payment systems. WordPress includes a plugins page which lists installed plugins. Figure 3.1 shows the installed WordPress plugins for a WordPress website which the user can activate and deactivate [8].



Plugin	Description
1WD Slider Activate | Delete	Slider Component for WordPress Version 1.0 | By Rakhitha Nimesh | Visit plugin site
Akismet Anti-Spam Activate | Delete	Used by millions, Akismet is quite possibly the best way in the world to **protect your blog from spam.** It keeps your site protected even while key. Version 4.0.3 | By Automattic | View details
Custom post plugin Activate | Delete	Display the plugin extension on the edit page. Version 0.1.0 | By Venkatesh Kadali | Visit plugin site
Ninja Tables Activate | Delete	The Easiest & Fastest Responsive Table Plugin on WordPress. Multiple templates, drag-&-drop live table builder, multiple color scheme, and s Version 1.9.8 | By WPManageNinja | View details
Page Plugin Activate | Delete	Display the plugin extension on the edit page. Version 0.1.0 | By Venkatesh Kadali | Visit plugin site
SlideDeck Activate | Delete	Create SlideDecks on your WordPress blogging platform and insert them into templates and posts. Get started creating SlideDecks from the r Version 5.1.11 | By SlideDeck | View details
Symbiota Search plugin Deactivate	Display the plugin extension on the edit page. Version 0.1.0 | By Venkatesh Kadali | Visit plugin site

Figure 3.1: WordPress plugins in WordPress website where the user can activate and deactivate

3.3 WordPress Plugins Functionality

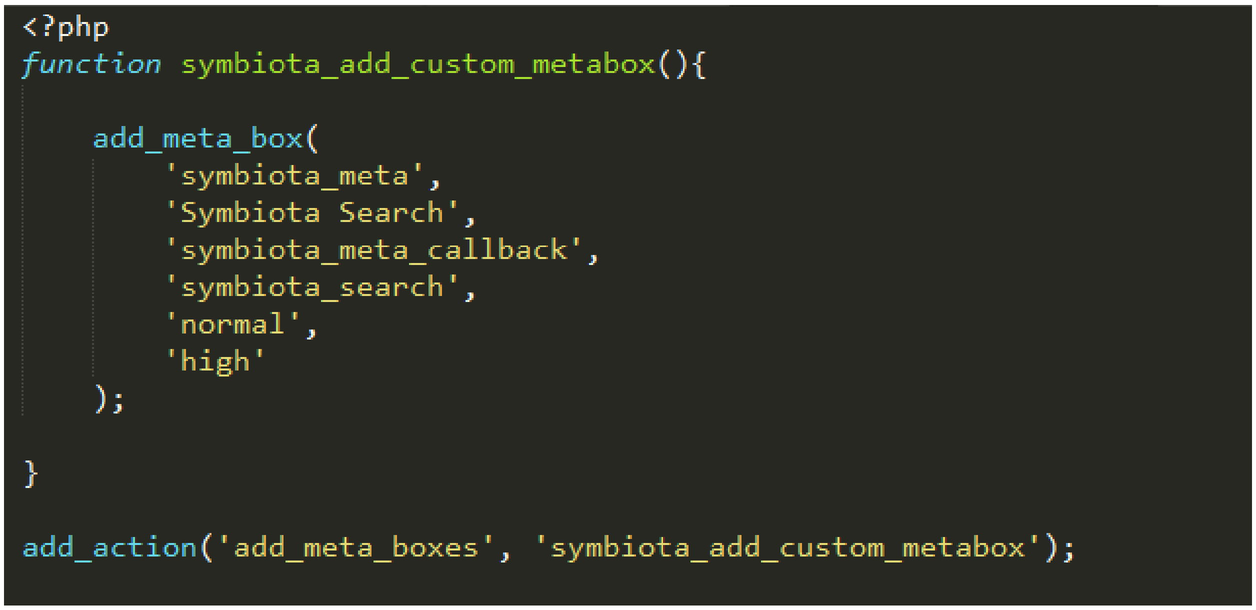
WordPress is designed so that developers can add their own code into it. The WordPress plugin API offers a set of hooks allowing developers to modify existing WordPress functionality or add new functionality. There are two kinds of hooks:

3.3.1 Actions

An action in WordPress is triggered at specific time when WordPress is running and lets the user to take an action. Publishing a post, changing themes, modifying database data are examples of actions.

The basic steps to add an action are:

- Create a PHP function to execute a specific event in the plugin file shown in Figure 3.2.
- Using the add_action() event, hook the above PHP function.
- Add the PHP function in a plugin file and activate it.



```
<?php
function symbiota_add_custom_metabox(){

    add_meta_box(
        'symbiota_meta',
        'Symbiota Search',
        'symbiota_meta_callback',
        'symbiota_search',
        'normal',
        'high'
    );

}

add_action('add_meta_boxes', 'symbiota_add_custom_metabox');
```

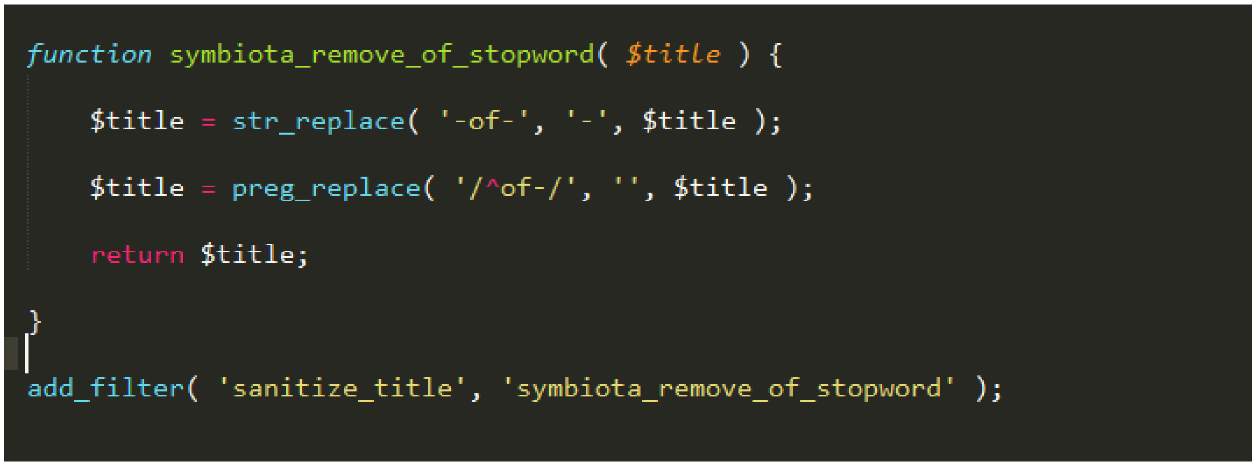
Figure 3.2: Adding a meta-box using add_action()

3.3.2 Filters

A filter in WordPress allows the user to modify WordPress data before it is sent to database or browser. Filters sit between the database and the browser, and between the browser and the database. Adding custom code to the end of the blogpost or customizing the excerpts display are the examples of filters.

Basic steps to add a filter are:

- Create a PHP function that filters the data, as shown in Figure 3.3.
- Hook the filter to WordPress, by calling add_filter() function.
- Add the PHP function in a plugin file and activate it.



```
function symbiota_remove_of_stopword( $title ) {

    $title = str_replace( '-of-', '-', $title );

    $title = preg_replace( '/^of-/', '', $title );

    return $title;

}

add_filter( 'sanitize_title', 'symbiota_remove_of_stopword' );
```

Figure 3.3: Cleaning up a webpage's permalink using add_filter()

The WordPress plugins installed in a site are registered in its WordPress database. A user can activate and deactivate them at any time. On each visit, WordPress connects to its database, loads the core software, and then loads the active plugins. All of this code is processed on the server and then sent to user's browser.

CHAPTER 4

SYMBIOTA SEARCH WORDPRESS PLUGIN ARCHITECTURE

The Symbiota Search Architecture consists of four parts [12].

1) Custom Post Type
2) Meta box
3) New Page and Short code
4) Search Functionality and Twig templates using Timber plugin

4.1 Custom Post Type

A custom post type is a custom defined piece of content. Using custom post types, one can define any type of content in WordPress. The custom post type is the feature that transforms WordPress from a blogging platform to a fully-fledged content management system. The custom post type can be anything, not just public-facing pieces of content.

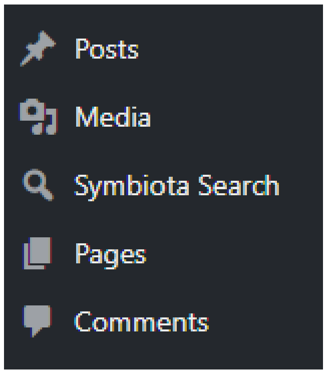


Figure 4.1: Custom post type

In Figure 4.1, Symbiota Search is a custom post type created in this project. Since plugins create their own custom post types, Symbiota search created a custom post type called Symbiota Search. Posts, Pages, Comments, and Media are the built-in post types. This custom post type has many functionalities which can be available to a developer when he makes his own product or interface. The developer may not have a lot of time to create a website or blog, so he uses these plugins or custom post types to make work easier and to save time.

4.1.1 Creating a custom post type

There are many plugins to create a custom post type. The Symbiota search custom post type is created from scratch. There is only one function needed to create a custom post type – register_post_type().

```
<?php register_post_type($post_type, $args); ?>
```

The register_post_type() function accepts two parameters:

- $post_type – The name of the post type. IT should contain only lowercase letters, no spaces, and a max length of 20 characters.
- $args – An array of arguments that define the post type and various options in WordPress.

```
<?php
    function symbiota_post_type_search(){
     $singular = __('Symbiota Search');
     $plural = __('Symbiota Search');
     $slug1 = str_replace( ' ', '_', strtolower( $singular ) );

     $labels = array(
          'name'                   =>$plural,
          'singular_name'          =>$singular,
          'add_new'           =>'Add New',
          'add_new_item'      =>'Add New ' . $singular,
          'edit'                   =>'Edit',
          'edit_item'              =>'Edit ' . $singular,
          'new_item'               =>'New' . $singular,
          'view'                   =>'View ' . $singular,
          'view_item'              =>'View ' . $singular,
     );

     $args = array(
          'labels'            => $labels,
          'public'            => true,
          'menu_position'     => 10,
          'menu_icon'         => 'dashicons-search',
          'can_export'        => true,
          'delete_with_user'  => false,
          'hierarchical'      => true,
          'capability_type'   => 'post',
          'map_meta_cap'      => true,
          'rewrite'           => array(
               'slug' => 'symbiota',
               'with_front' => true,
               'pages' => true,
               'feeds' => true,
          ),
          'supports'          => array(
                'title' ,
            )
     );

     register_post_type($slug1, $args);
    }


    add_action('init', 'symbiota_post_type_search'); ?>
```

Code to create a custom post type

The register post type is hooked to the init action at the end using the add_action() function. The add_action() function hooks a function to a specific action. Here the add_action() function hooks the symbiota_post_type_search function (which contains register_post_type) to the init action. Labels and args are used to create a custom post type. The labels property is mainly used to create a new post or delete a new post. There are many parameters in the labels property which allows the developer to configure the visibility of the custom post type in the frontend and backend.

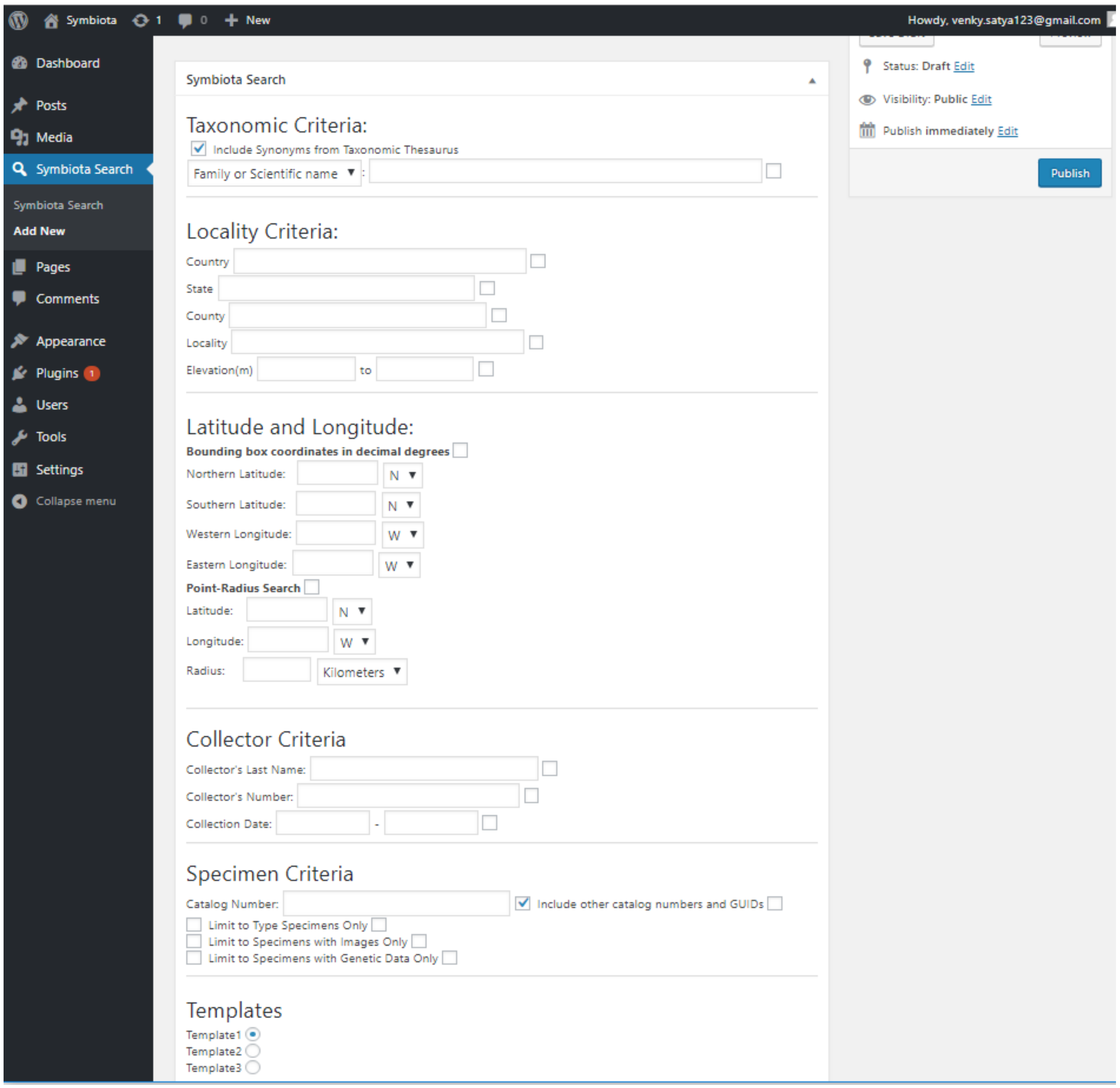
The most important one in the args parameter is public parameter, which sets the values of other properties in one call.

- If it is set true, the post will be included in searches, the UI will be shown, it will show up in the admin bar and so on.
- If it is set to false, the post type is excluded from searches, it will not show up in the UI, and it will be hidden in the menus, etc.

4.2 Meta box

Meta boxes are used for adding additional information to your posts, pages, and content. WordPress includes many meta boxes on the Add New Post or Add New Page screens. Creating meta boxes is a crucial part of WordPress plugin development. A custom post type in WordPress always needs to add some additional sort of data to it which can be done with the help of meta boxes[10].

Figure 4.2 is a Symbiota search plugin meta box representation. The meta box consists of many search fields related to taxonomic criteria, geographic criteria, collector criteria, specimen criteria and template selection. Each field consists of a check box where the user can select the fields he needs. The templates block consists of radio buttons where the user can select one template that can be displayed in the output. There are three different templates which shows the output in different templates or tables. The publish button on the right-top corner saves all the selected fields into the WordPress database.



Symbiota Search

Taxonomic Criteria:

☑ Include Synonyms from Taxonomic Thesaurus

Family or Scientific name ▼ : ☐

Locality Criteria:

Country ☐
State ☐
County ☐
Locality ☐
Elevation(m) to ☐

Latitude and Longitude:

Bounding box coordinates in decimal degrees ☐
Northern Latitude: N ▼
Southern Latitude: N ▼
Western Longitude: W ▼
Eastern Longitude: W ▼
Point-Radius Search ☐
Latitude: N ▼
Longitude: W ▼
Radius: Kilometers ▼

Collector Criteria

Collector's Last Name: ☐
Collector's Number: ☐
Collection Date: - ☐

Specimen Criteria

Catalog Number: ☑ Include other catalog numbers and GUIDs ☐
☐ Limit to Type Specimens Only ☐
☐ Limit to Specimens with Images Only ☐
☐ Limit to Specimens with Genetic Data Only ☐

Templates

Template1 ●
Template2 ○
Template3 ○

Figure 4.2: The Meta box in Symbiota search plugin consists of five parts

The meta box creation process is broken down into three parts:

1) Adding the meta box
2) Rendering the meta box
3) Saving the data

4.2.1 Adding the meta box

Meta boxes can be created in a plugin using the add_meta_box() function in WordPress. This function accepts seven parameters.

```
<?php  add_meta_box($id,  Stitle,  $callback,  $page,  $context,
$priority, $callback_args); ?>
```

- $id: The CSS ID attribute for the meta box.
- $title: The title displayed at the top of the meta box.
- $callback: The function that renders the meta box.
- $page: The page where the meta box to be displayed.
- $context: The part of the page where meta box should be displayed.
- $priority: The priority within the context where the meta box should display.
- $callback_args: Arguments to pass into your callback function.

```
<?php
function symbiota_add_custom_metabox(){

     add_meta_box(
          'symbiota_meta',
          'Symbiota Search',
          'symbiota_meta_callback',
          'symbiota_search',
          'normal',
          'high'
     );

}
add_action('add_meta_boxes', 'symbiota_add_custom_metabox');
?>
```

Code to add a meta-box

Adding the add_meta_box() function alone causes errors as it called before WordPress is loaded. So, this function is hooked into a WordPress action hook. It is hooked to the add_meta_boxes hook by using the add_action() function.

4.2.2 Rendering the Meta box

Rendering the meta box consists of adding the fields. This includes an HTML form mixed in with a bit of PHP to display the saved data. the Meta box is rendered by creating a callback function which is included in add_meta_box() function. This callback function is responsible of displaying all the fields inside the metabox.

```
<?php
function symbiota_meta_callback( $post ){
      echo "What you put here, shows up in the meta box";
}
?>
```

4.2.3 Saving the Data

To save the user selected data in the meta box, the save_post WordPress hook is used which work like the action hook.
`<?php add_action('save_post', 'symbiota_meta_save'); ?>`

The symbiota_meta_save() function consists of post parameter which takes care of saving the data into the database. The save_post hook activates after the publish or save draft button has been hit. All the data is saved according to the post parameter in the wp_posts table in the WordPress database. The user has access to all the $_POST data, which includes the meta box fields. Before saving the data, the user must check three things:

1) Check if the post is auto saving.
2) Verify the nonce value which was created earlier.
3) Check to make sure the current user can edit the post.

```
<?php
function symbiota_meta_save($post){
     $is_autosave = wp_is_post_autosave($post);
     $is_revision = wp_is_post_revision($post);
     $is_valid_nonce = (isset($_POST['symbiota_jobs_nonce']) &&
wp_verify_nonce($_POST[   'symbiota_jobs_nonce'   ],   basename(
__FILE__ ) ) ) ? 'true' : 'false';

     if ( $is_autosave || $is_revision || !$is_valid_nonce ) {
          return;
     }
     if( isset($_POST['cb_taxontype'])){
          update_post_meta($post,'cb_taxontype',
          sanitize_text_field($_POST['cb_taxontype']));
     }
}
?>
```

Code to check the above mentioned three things

The code is saved in the database using the update_post_meta() function. It takes three arguments: a post ID, the meta key, and the value.

4.3 Shortcode Generation

A shortcode is a special tag or a WordPress specific code that lets the user do things with little effort. It is entered in a post that gets replaced with different content when viewing the post on the website. It can include files or create objects that would require lots of complicated ugly code in just one line. WordPress introduced the shortcode API six years ago and the shortcodes are now used by many WordPress plugins to allow users to add content to their posts and pages.

Short code is represented inside a square bracket. This code gets replaced with some other code when the page is loaded in a web browser. WordPress allows developers to create their own custom shortcodes. Shortcode names should be all lowercase letters, numbers and underscores. There are many inbuilt short codes like [gallery], [audio], [dailymotion]. The [gallery] short code displays the gallery of the selected images on the website after loading the blog page with this short code. [symbiota id='123'] is a short code used in this plugin. This short code consists of ID as an attribute.

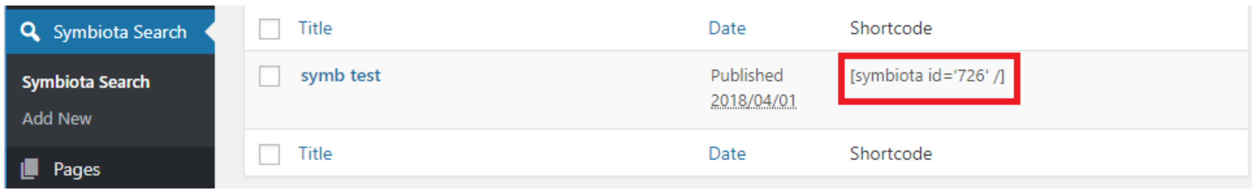
The shortcode API consists of a simple set of functions for creating WordPress shortcodes for use in posts and pages. This API helps developers to make special kinds of content that users can attach to certain pages by adding the matching shortcode into the page text. WordPress makes it very easy for users to create shortcodes. The PHP code required to create a shortcode is:

```
add_shortcode('symbiota', 'symbiota_shortcode_function');

function symbiota_shortcode_function($attr, $content){
    //Required HTML and PHP code to display content on web page.
}
```

Code to create a shortcode

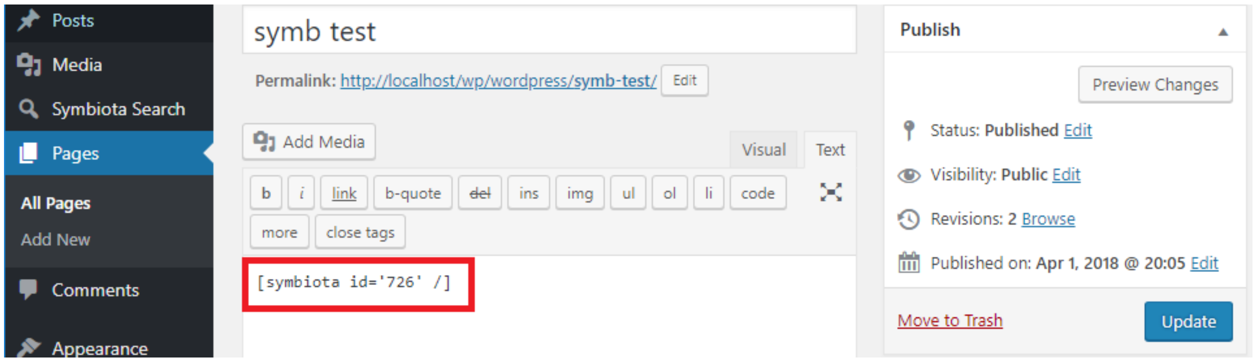
The add_shortcode() function is used to register a shortcode handler. It takes two parameters: the shortcode name and the callback function name. Here in the above code, symbiota is a shortcode and symbiota_shortcode_function is a callback function. Eventually, a shortcode named symbiota is created.



Title	Date	Shortcode
symb test	Published 2018/04/01	[symbiota id='726' /]
Title	Date	Shortcode

Figure 4.3: A shortcode is generated after clicking update button in the meta box

The user can copy the generated shortcode shown in Figure 4.3 and paste it in a new page displayed in Figure 4.4 to display the selected content on his web page.



symb test

Permalink: http://localhost/wp/wordpress/symb-test/ Edit

```
[symbiota id='726' /]
```

Status: Published Edit

Visibility: Public Edit

Revisions: 2 Browse

Published on: Apr 1, 2018 @ 20:05 Edit

Figure 4.4: Pasted the generated shortcode in a new page called symb test

The selected content in the meta box will be displayed on the web page shown in Figure 4.5 when the user clicks update button after pasting the shortcode in the new page.

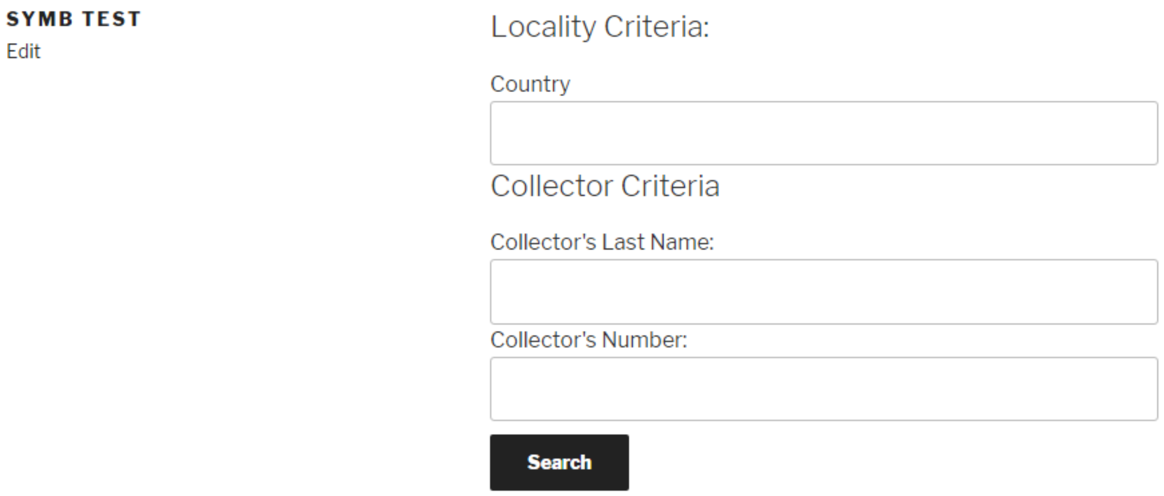


Figure 4.5: The selected content in the meta box in displayed on the web page

4.4 Search Functionality and Twig Templates using Timber plugin

The search functionality in Figure 4.5 similar to the Symbiota search website. The search button in the above figure is linked to the search.php page. This page is linked to Symbiota code and twig templates. Clicking the search button after entering the details in the search bars give all the details to the Symbiota code which returns a php array. An object is created to the php class in Symbiota code to get the php array to search.php page [9].

The twig concept is used in this plugin for creating multiple templates to display the php array in tables. Twig is the flexible, fast and secure template engine for PHP. It is a template engine for the PHP programming language. Twig is used by many Open-Source projects like Symfony, Drupal.

The key features of twig are:

- **Fast**: Twig compiles templates down to plain optimized PHP code. The overhead compared to regular PHP code is minimal.

- **Secure**: Twig has a sandbox mode to evaluate untrusted template code. This allows twig to be used as a template language for applications where users may modify the template design.

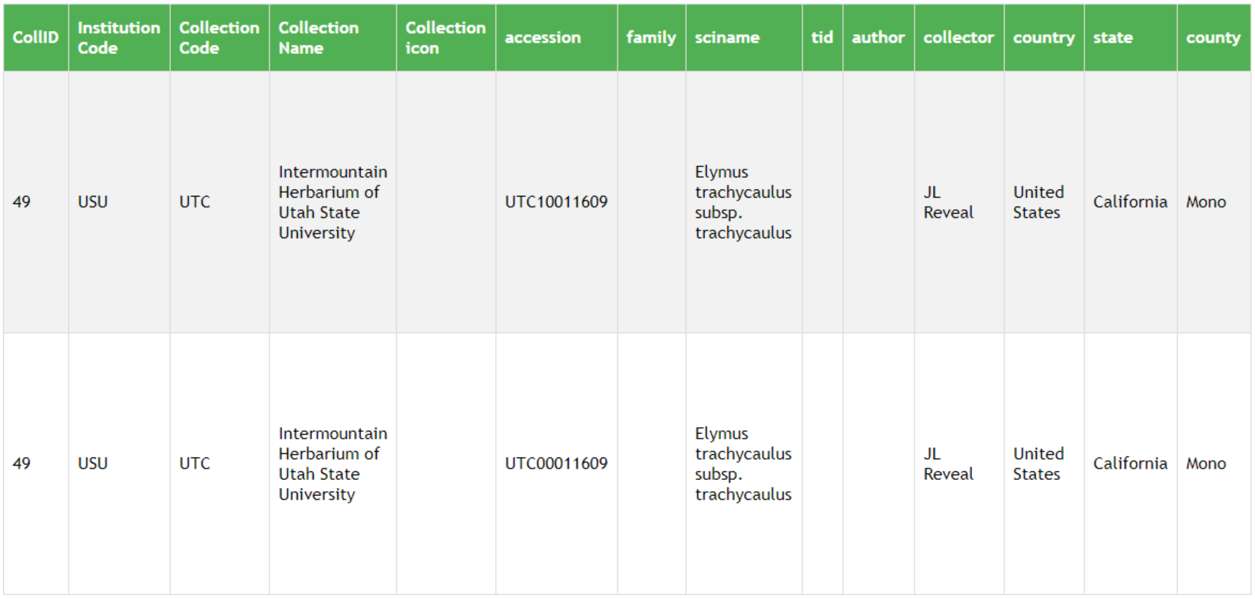
- **Flexible**: Twig is provided with a flexible lexer and parser. This allows the developer to define their own custom tags and filters.

Three different templates are created using the twig template engine. To access the twig script, the Timber plugin is used. The user can select the templates in the meta box. Three templates are added to this plugin by the time of writing this paper.

The Timber plugin separates the design and logic so that the developer can concentrate completely on logic using PHP and twig file can focus totally on HTML and display. The Timber plugin helps in creating fully-customized WordPress templates or themes faster with more sustainable code. Once the Timber plugin is installed in the plugin directory of WordPress, it gives any WordPress theme or plugin the ability to take advantage of the power of Twig and other Timber features.

The three different templates:

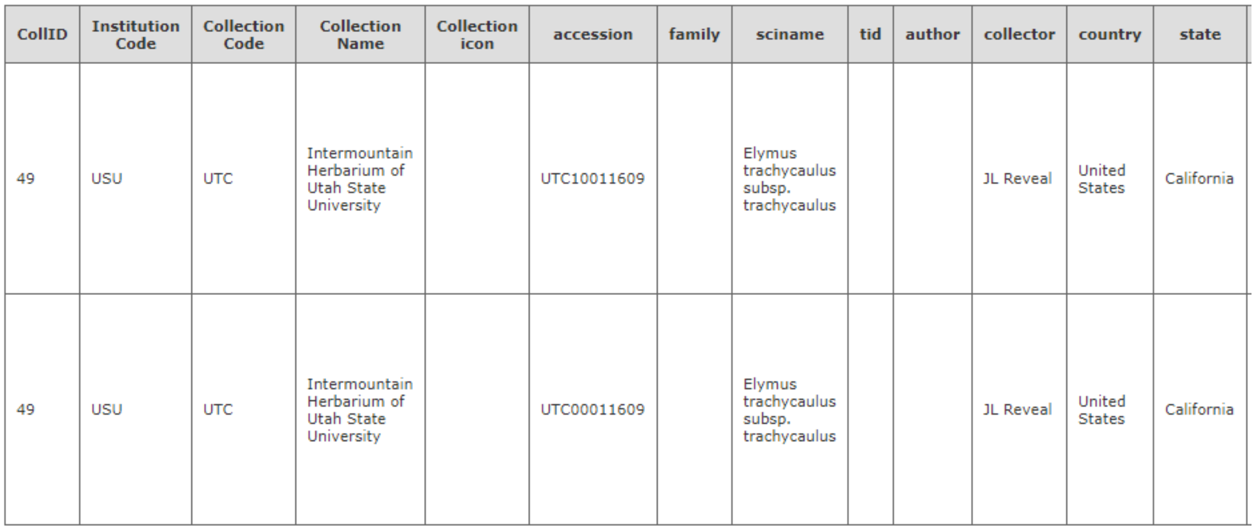
Template 1



CollID	Institution Code	Collection Code	Collection Name	Collection icon	accession	family	sciname	tid	author	collector	country	state	county
49	USU	UTC	Intermountain Herbarium of Utah State University		UTC10011609		Elymus trachycaulus subsp. trachycaulus			JL Reveal	United States	California	Mono
49	USU	UTC	Intermountain Herbarium of Utah State University		UTC00011609		Elymus trachycaulus subsp. trachycaulus			JL Reveal	United States	California	Mono

Figure 4.6: Template 1

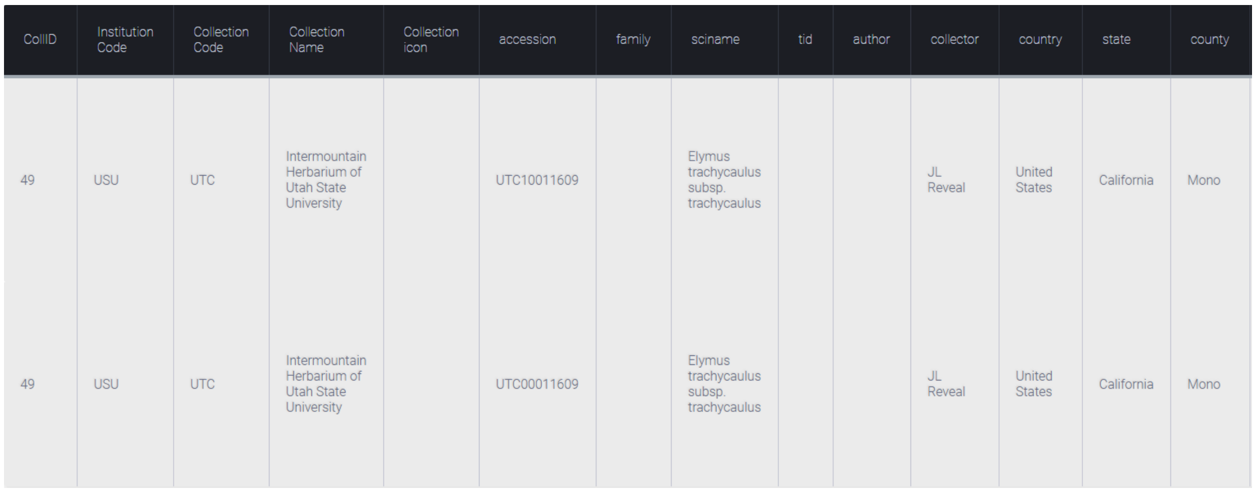
Template 2



CollID	Institution Code	Collection Code	Collection Name	Collection icon	accession	family	sciname	tid	author	collector	country	state
49	USU	UTC	Intermountain Herbarium of Utah State University		UTC10011609		Elymus trachycaulus subsp. trachycaulus			JL Reveal	United States	California
49	USU	UTC	Intermountain Herbarium of Utah State University		UTC00011609		Elymus trachycaulus subsp. trachycaulus			JL Reveal	United States	California

Figure 4.7: Template 2

Template 3



CollID	Institution Code	Collection Code	Collection Name	Collection icon	accession	family	sciname	tid	author	collector	country	state	county
49	USU	UTC	Intermountain Herbarium of Utah State University		UTC10011609		Elymus trachycaulus subsp. trachycaulus			JL Reveal	United States	California	Mono
49	USU	UTC	Intermountain Herbarium of Utah State University		UTC00011609		Elymus trachycaulus subsp. trachycaulus			JL Reveal	United States	California	Mono

Figure 4.8: Template 3

CHAPTER 5

SYMBIOTA VALIDATION WORDPRESS PLUGIN ARCHITECTURE

The architecture of the Symbiota validation plugin is similar to Symbiota search plugin except for the last step. This validation plugin architecture consists of four parts.

1) Custom Post Type
2) Meta box
3) New Page and Short code
4) Form Validation

5.1 Custom Post Type

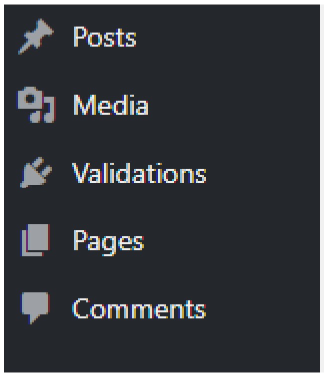
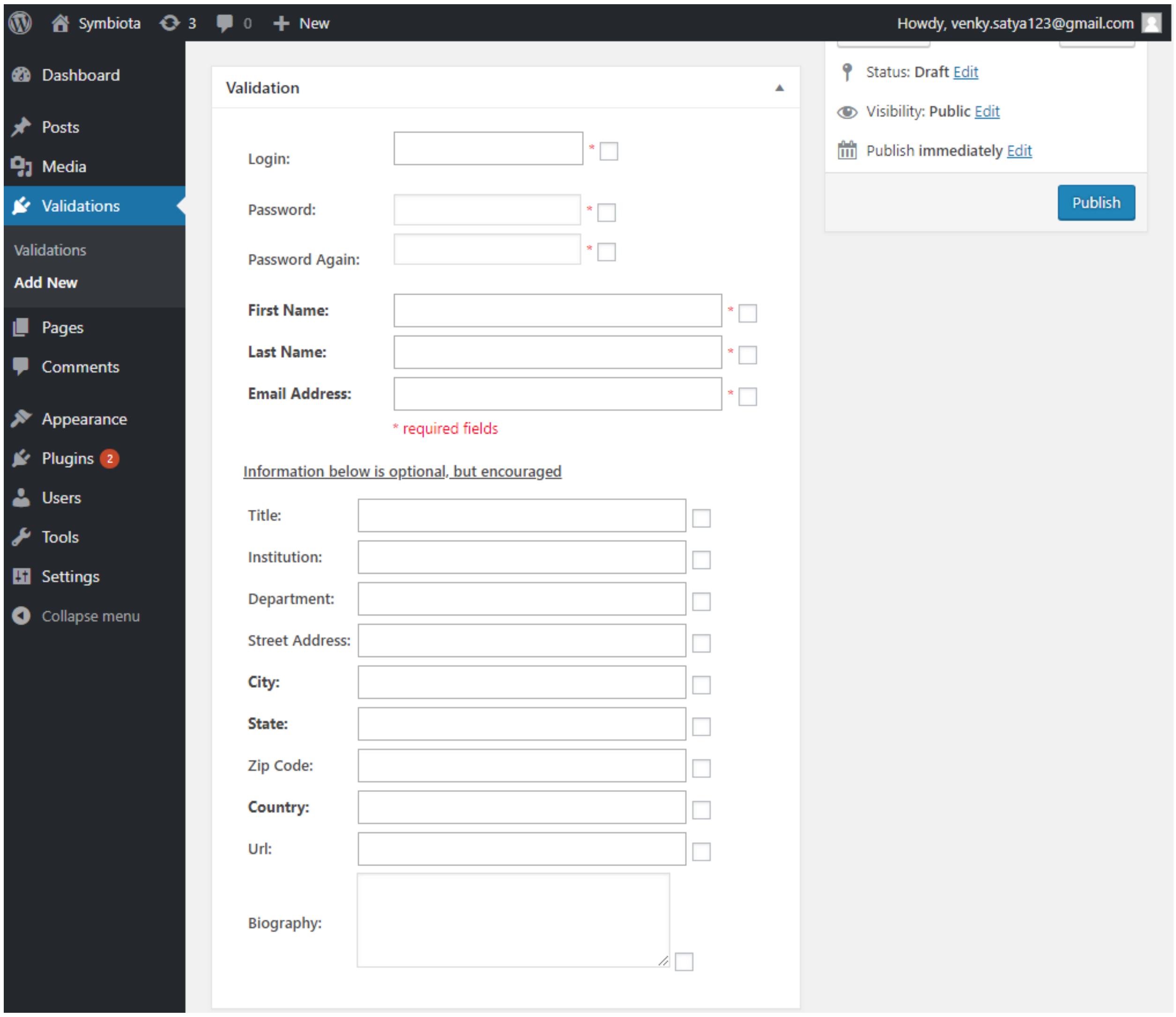


Figure 5.1: Custom post type

In Figure 5.1, Symbiota Validations is the custom post type created in this project. This custom post type will be available to a developer when he makes his own product or interface. Creating a custom post type is similar to what was shown for the first plugin in Section 4.1.1.

5.2 Meta box

The most crucial part of WordPress plugin development is Meta box creation. Figure 5.2 is a representation of meta box in Symbiota validation plugin. This meta box consists of a user registration form where a WordPress developer can select and embed them in their WordPress websites. This form is divided into two parts – required fields and optional fields. Meta box creation has three parts which includes adding the meta box, rendering the meta box and saving the data which were described in the Section 4.2.



Login: *

Password: *

Password Again: *

First Name: *

Last Name: *

Email Address: *

* required fields

Information below is optional, but encouraged

Title:

Institution:

Department:

Street Address:

City:

State:

Zip Code:

Country:

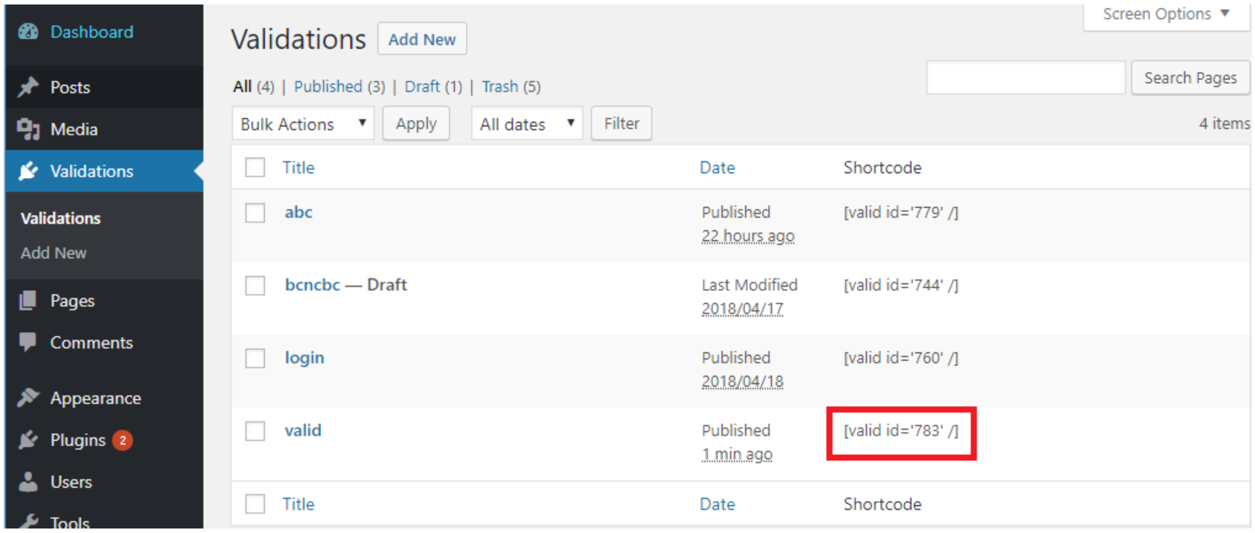
Url:

Biography:

Figure 5.2: Meta box in Symbiota validation plugin

5.3 Shortcode Generation

WordPress features a Shortcode API that can be used to easily create shortcode functionality in plugins. Shortcodes are basically text macro codes that can be inserted into a post, page, or custom post type. Shortcode creation was described in Section 4.3. Figure 5.3 shows a shortcode created after selecting the forms in the meta box. [valid] is the shortcode for this plugin. A new page called valid is created in Figure 5.4 in which the generated shortcode is pasted. The selected content in the meta box is displayed on the web page when the user clicks on the update button after pasting the shortcode in the new page as shown in Figure 5.5.

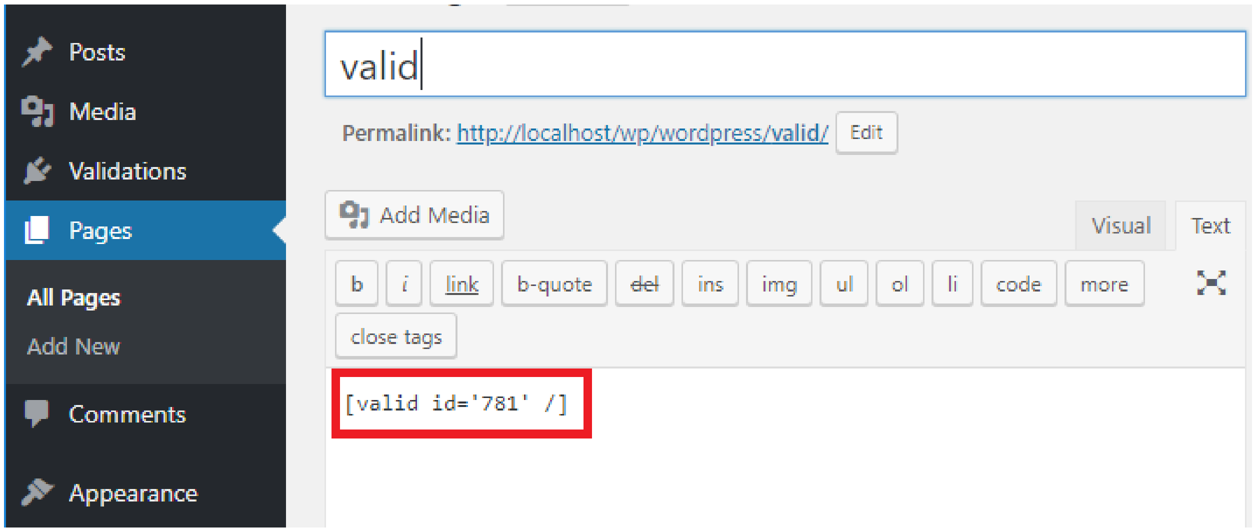


All (4) | Published (3) | Draft (1) | Trash (5)

4 items

	Title	Date	Shortcode
☐	abc	Published 22 hours ago	[valid id='779' /]
☐	bcncbc — Draft	Last Modified 2018/04/17	[valid id='744' /]
☐	login	Published 2018/04/18	[valid id='760' /]
☐	valid	Published 1 min ago	[valid id='783' /]
☐	Title	Date	Shortcode

Figure 5.3: A shortcode is generated after clicking on the update button in the meta box

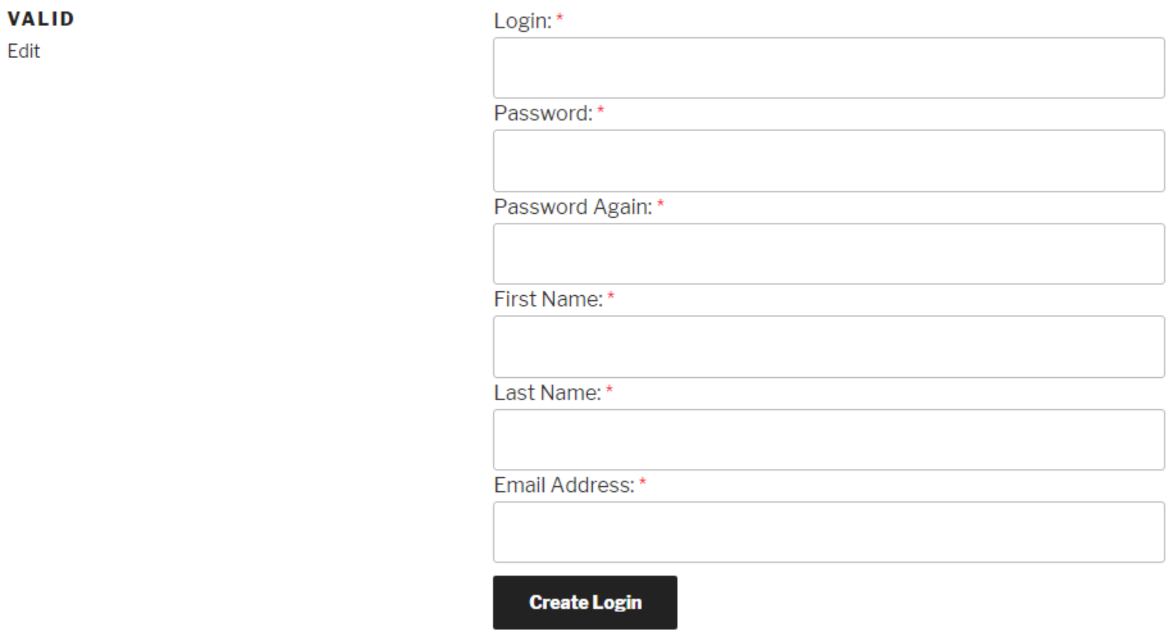


valid

Permalink: http://localhost/wp/wordpress/valid/

```
[valid id='781' /]
```

Figure 5.4: Paste the generated shortcode in a new page called valid


VALID
Edit

Login: *

Password: *

Password Again: *

First Name: *

Last Name: *

Email Address: *

Figure 5.5: The selected content in the meta box is displayed on the web page

5.4 Form Validation

Form validation is used to register a new user in Symbiota database. This form is directly connected to the Symbiota database so that the user can login if he already has an account or else he can create a new account by filling the form. Whenever the user clicks on Create Login button by entering all the details, the user is registered into the Symbiota database and a Symbiota page gets opened.

CHAPTER 6

SUMMARY

The Symbiota software project is a combination of many PHP files. This code can be reduced in size and complexity by separating the graphical user interface from the core functionality. This can be done by adding an MVC framework to Symbiota. WordPress is a free and open source content management system which is easy for the users to create and manage their content without any technical knowledge.

The main goal of this project is to refactor Symbiota code in a way that helps WordPress developers to easily embed the Symbiota features into their own website and also should answer the question whether the separation of Symbiota's user interface with its core structure is feasible. This project consists of two plugins which helps WordPress developers to add features like specimen search and form validation to their website without knowing the single line of code.

The Symbiota search plugin allows users to choose required fields of the specimen search engine to add them to their website. This plugin helps users to create posts with search bars and allows the users to update the created posts whenever needed. A user can select different templates to visualize the queried data. The Symbiota validation plugin helps WordPress developers to add the Sign in and Sign up forms to their WordPress website by just selecting the required forms. Eventually, by using these plugins, users can read from and write to the Symbiota database without knowing the single line of Symbiota code.

These plugins show that the separation of Symbiota's user interface from its core structure is feasible. These plugins are also helpful for Symbiota developers who want to add a Symbiota interface into their website. Future work includes developing more plugins which helps in providing more Symbiota features to Symbiota developers and developing plugins which supports the Joomla and Drupal content management systems as they have a market share of 7% and 5%.

REFERENCES

[1] Symbiota specimen search Engine in Open Herbarium website [Online]. Available: http://openherbarium.org/portal/collections/harvestparams.php

[2] Symbiota Biodiversity Management Software code in GitHub [Online]. Available: https://github.com/Symbiota

[3] Symbiota Working Group in IDIGBIO website [Online]. Available: https://www.idigbio.org/wiki/index.php/Symbiota_Working_Group

[4] Symbiota Overview, May 11, 2017 [Online]. Available: http://symbiota.org/docs/wp-content/uploads/Symbiota_Primer_2017.pdf

[5] Coder Cruise 2018, PHP Documentation [Online]. Available: http://php.net/docs.php

[6] WordPress – Content Management System [Online]. Available: https://wordpress.org/

[7] WordPress -Wikipedia, the free encyclopedia [Online]. Available: https://en.wikipedia.org/wiki/WordPress

[8] WordPress Plugins – Extend your WordPress experience with 54,953 plugins [Online]. Available: https://wordpress.org/plugins/

[9] Twig template engine for PHP Documentation [Online]. Available: https://twig.symfony.com/

[10] Custom WordPress Meta Box Creation [Online]. Available: https://code.tutsplus.com/tutorials/how-to-create-custom-wordpress-writemeta-boxeswp20336

[11] Symbiota – Mobilizing Biodiversity Data, Feb 12, 2018 [Online]. Available: http://symbiota.org/docs/wp-content/uploads/Symbiota_Dec_27_2017.pdf

[12] WordPress Codex – Online Manual for WordPress and a living repository for WordPress information and documentation [Online]. Available: https://codex.wordpress.org/

[13] Symbiota – Promoting Bio-Collaboration [Online]. Available: http://symbiota.org/docs/

[14] Corinna Fries, Edward E. Gilbert, Nico M. Franz, "Symbiota – A virtual platform for creating voucher-based biodiversity information communities", Jun 24, 2014 [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4092327/

[15] Paolo Galli, Fabrizio Stefani, Francesca Benzoni, Aldo Zullini, "Introduction of Alien Host-parasite Complexes in a Natural Environment and the Symbiota Concept", Mar 12, 2005 [Online]. Available: https://link.springer.com/article/10.1007/s10750-005-3645-0

[16] Beginner's Guide for WordPress – wpbeginner [Online]. Available: http://www.wpbeginner.com/

[17] Brad Williams, David Damstra, Hal Stern, "Professional WordPress Design and Development", 2013 [Online]. Available: http://www.stilson.net/documentation/Professional%20WordPress,%202nd%20Edition.pdf

[18] XAMPP and WordPress Installation locally on Windows, Dec 27, 2017 [Online]. Available: https://themeisle.com/blog/install-xampp-and-wordpress-locally/

APPENDICES

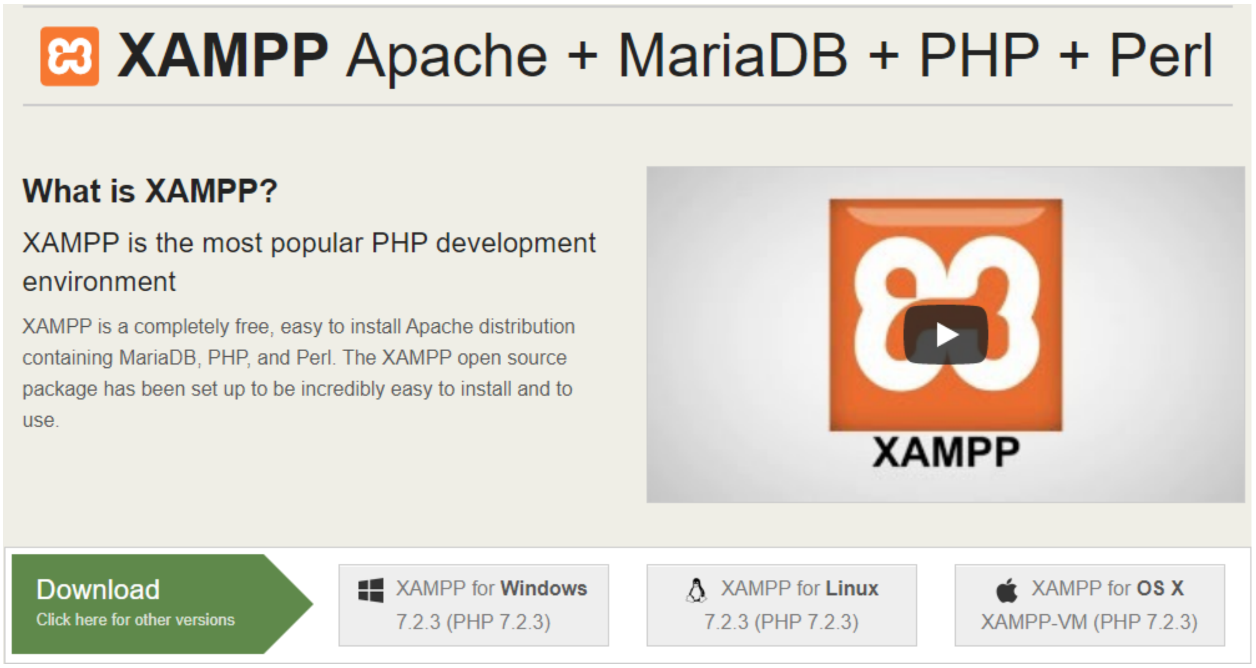
APPENDIX A

A.1 Installation of XAMPP server and WordPress on Microsoft Windows

To create a local WordPress site, web server software (Apache) setup, PHP and MySQL are needed. PHP is a programming language and MySQL is a database management software. Both are needed to run WordPress. Installing and setting up this software separately is a difficult task for users. XAMPP makes it easy for the users to build WordPress websites locally. Creating local WordPress sites is a common practice among developers and site owners. It allows the developers to test WordPress without creating an actual website on Internet. The developer can try different WordPress plugins and themes as the local websites are only visible to the developer [18].

XAMPP: XAMPP stands for cross-platform, Apache, MySQL, PHP and Perl. The main function of it is to create a local web server for testing purposes. Since XAMPP is cross-platform, it works on Mac, Windows and Linux. This documentation shows XAMPP installation in Windows 10.

Step1: Download and install XAMPP on a computer


XAMPP Apache + MariaDB + PHP + Perl

What is XAMPP?

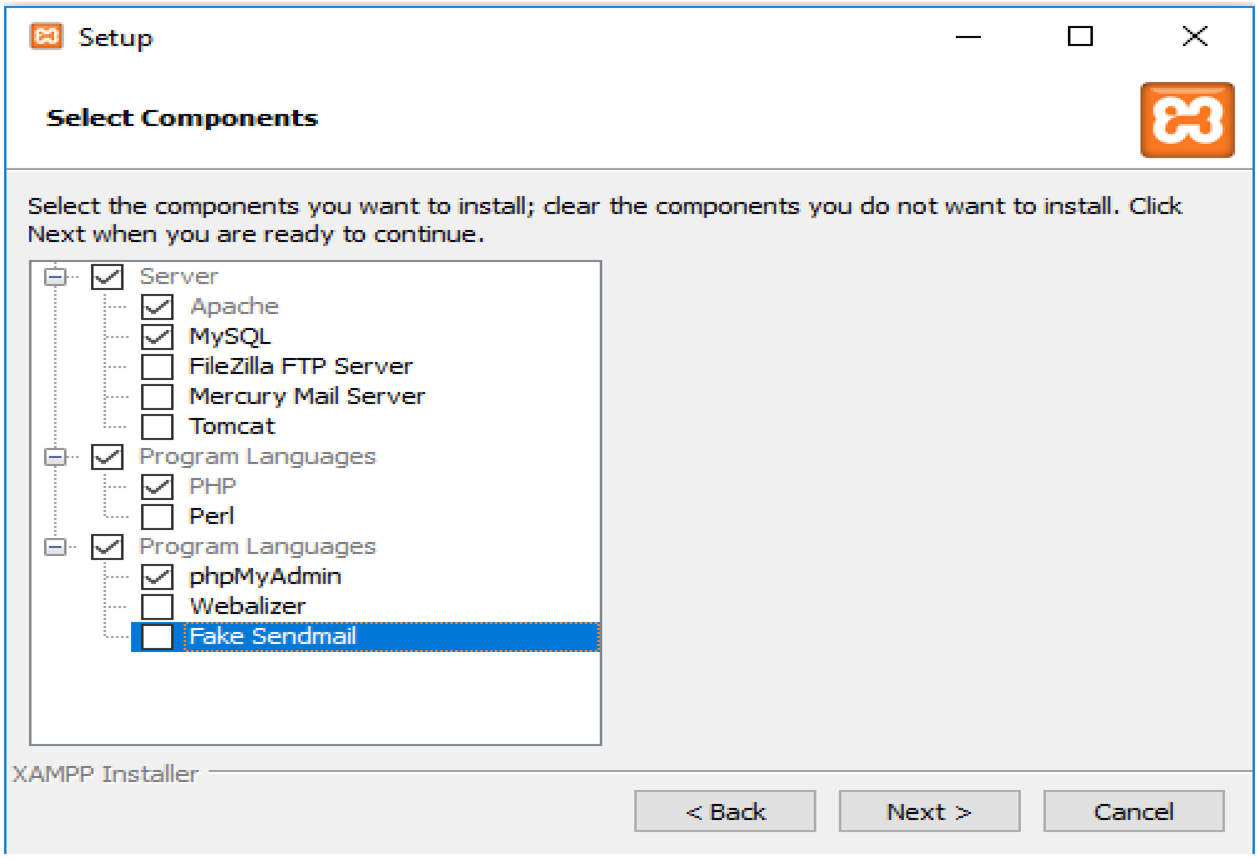
XAMPP is the most popular PHP development environment

XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.

XAMPP
Download
Click here for other versions
XAMPP for Windows 7.2.3 (PHP 7.2.3)
XAMPP for Linux 7.2.3 (PHP 7.2.3)
XAMPP for OS X XAMPP-VM (PHP 7.2.3)

Figure A.1: Download XAMPP Windows installer file from apachefriends.org and run the file.

To install XAMPP and WordPress, MySQL and PHPMyAdmin are needed. So, select those in the next screen.



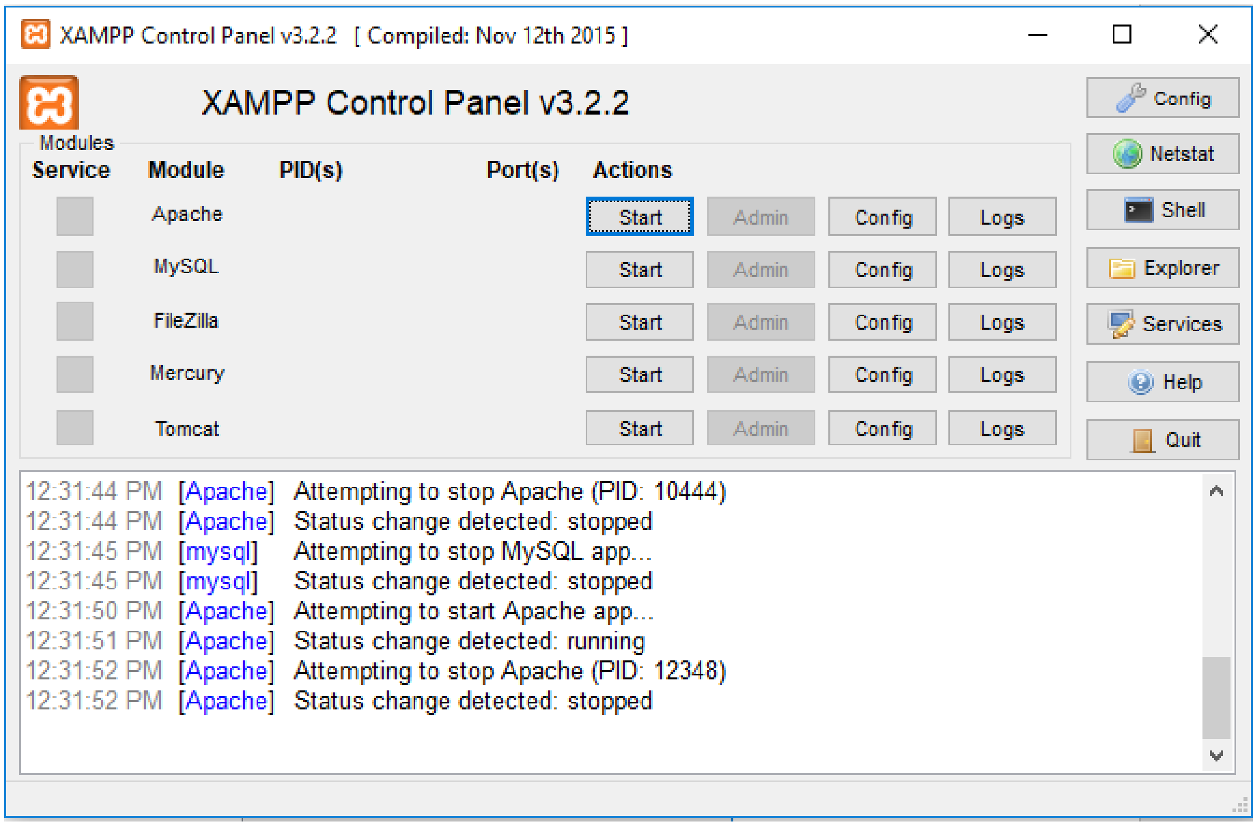
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- [✓] Server
  - [✓] Apache
  - [✓] MySQL
  - [ ] FileZilla FTP Server
  - [ ] Mercury Mail Server
  - [ ] Tomcat
- [✓] Program Languages
  - [✓] PHP
  - [ ] Perl
- [✓] Program Languages
  - [✓] phpMyAdmin
  - [ ] Webalizer
  - [ ] Fake Sendmail

Figure A.2: Select Apache and MySQL to install

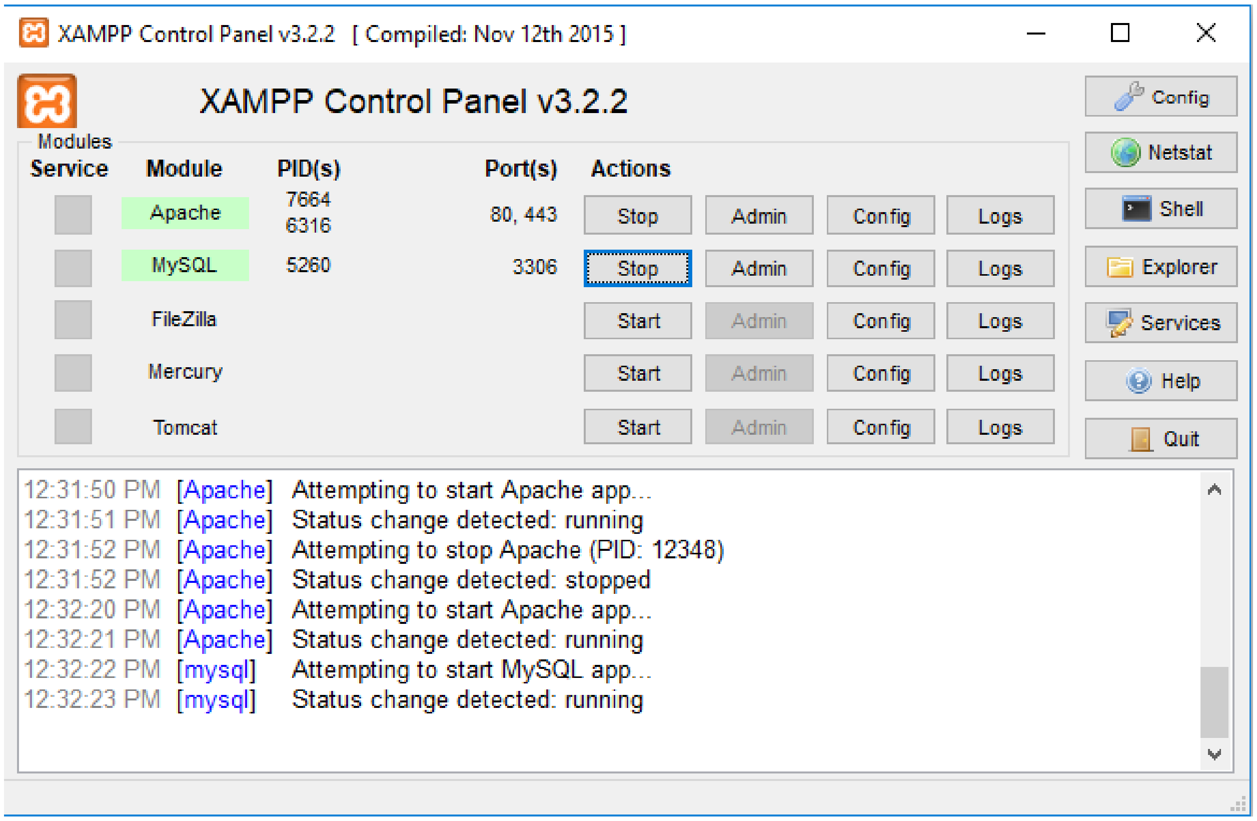
Step2: Start the modules and test the server.

Apache and MySQL are the two modules that are needed to run to install XAMPP and WordPress properly. The XAMPP control panel is the screen used to start those modules.



```
12:31:44 PM  [Apache]   Attempting to stop Apache (PID: 10444)
12:31:44 PM  [Apache]   Status change detected: stopped
12:31:45 PM  [mysql]    Attempting to stop MySQL app...
12:31:45 PM  [mysql]    Status change detected: stopped
12:31:50 PM  [Apache]   Attempting to start Apache app...
12:31:51 PM  [Apache]   Status change detected: running
12:31:52 PM  [Apache]   Attempting to stop Apache (PID: 12348)
12:31:52 PM  [Apache]   Status change detected: stopped
```

Figure A.3: XAMPP control panel before starting the Apache and MySQL modules

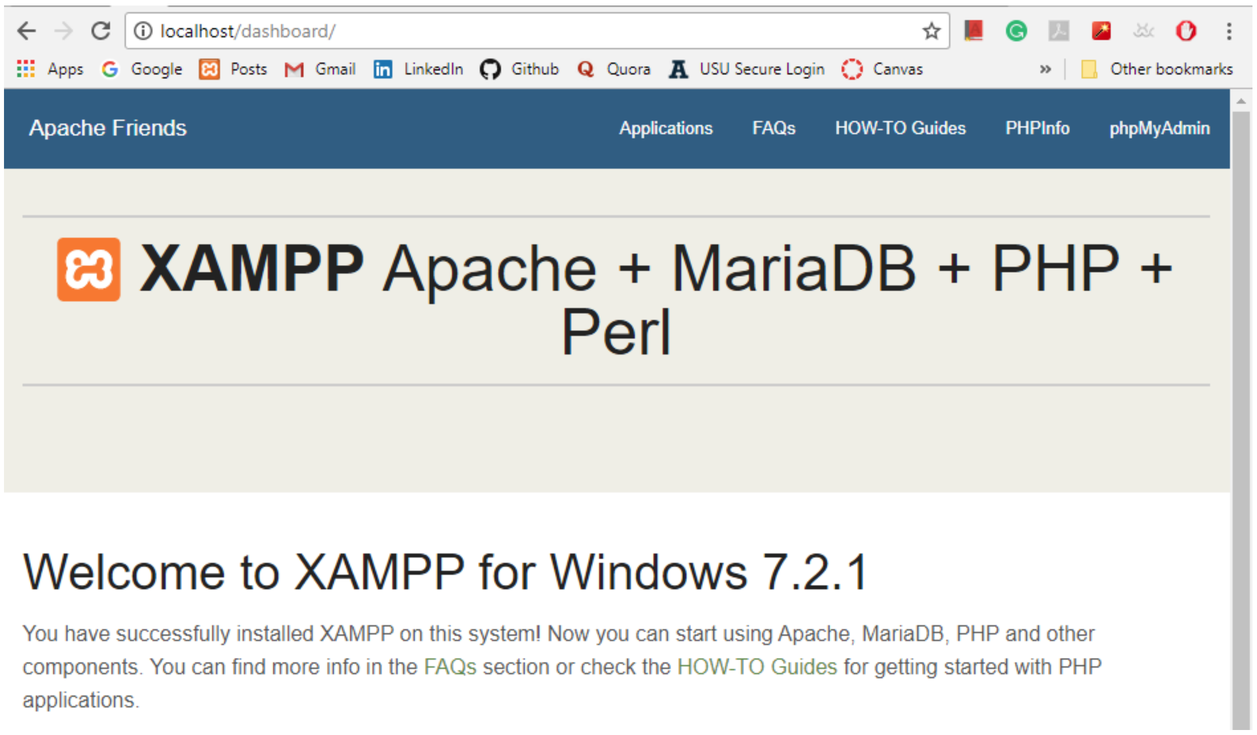

XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]
XAMPP Control Panel v3.2.2
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
	Apache	7664 6316	80, 443	Stop	Admin	Config	Logs
	MySQL	5260	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
12:31:50 PM  [Apache]   Attempting to start Apache app...
12:31:51 PM  [Apache]   Status change detected: running
12:31:52 PM  [Apache]   Attempting to stop Apache (PID: 12348)
12:31:52 PM  [Apache]   Status change detected: stopped
12:32:20 PM  [Apache]   Attempting to start Apache app...
12:32:21 PM  [Apache]   Status change detected: running
12:32:22 PM  [mysql]    Attempting to start MySQL app...
12:32:23 PM  [mysql]    Status change detected: running
```

Figure A.4: XAMPP control panel after starting the Apache and MySQL modules.

The status of Apache and MySQL turn to green after launching them. Now the user should be able to test that the local server is working by going to http://localhost/ in the web browser.



XAMPP Apache + MariaDB + PHP + Perl

Welcome to XAMPP for Windows 7.2.1

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.

Figure A.5: Testing the local server by going to localhost in web browser

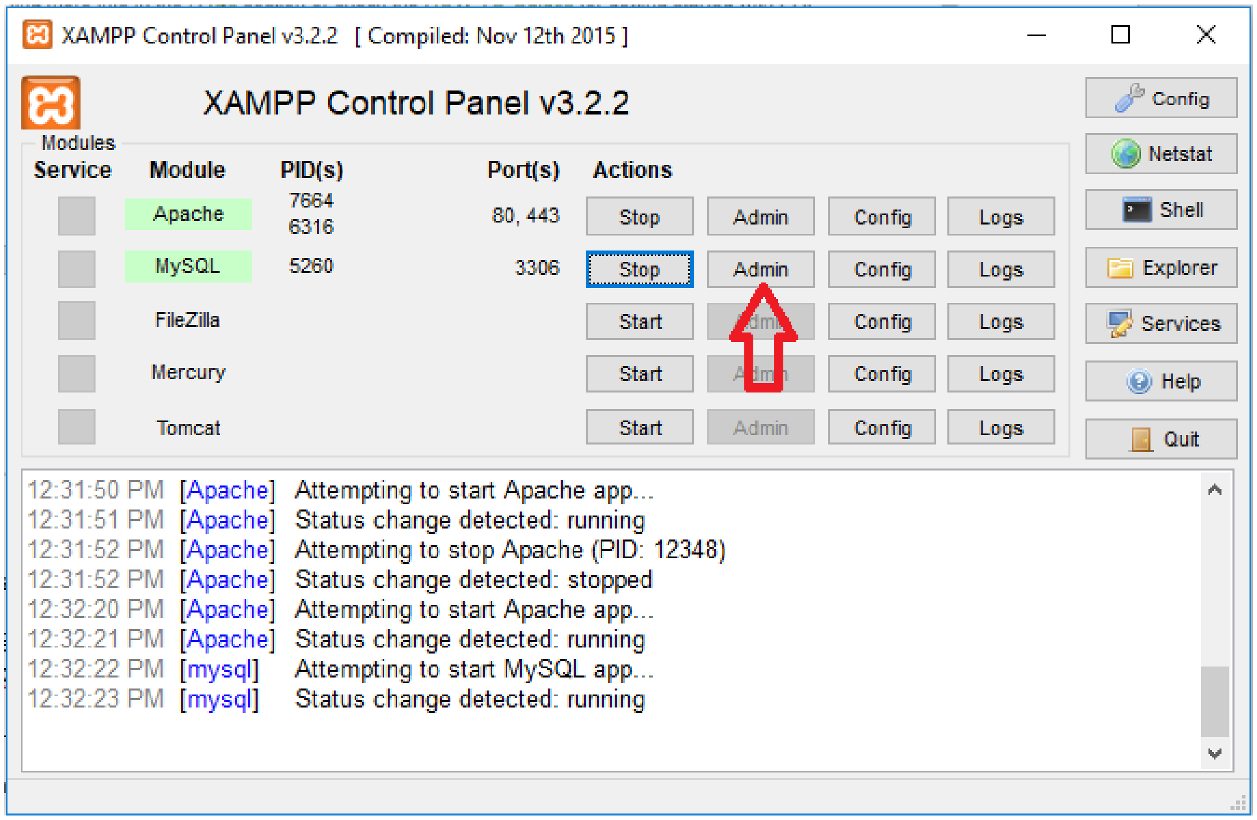
If the XAMPP sever is working properly, then WordPress can be installed while running the XAMPP.

Step 3: Download and add WordPress files

Visit wordpress.org to download the latest version of WordPress. Extract the zip file. Create a new folder, WP in htdocs and move the WordPress file into it.
On Windows it would be C:\xampp\htdocs

Step 4: Create a database for WordPress

To create a database for the WordPress, launch PHPMyAdmin page by clicking on MySQL admin in XAMPP control panel.

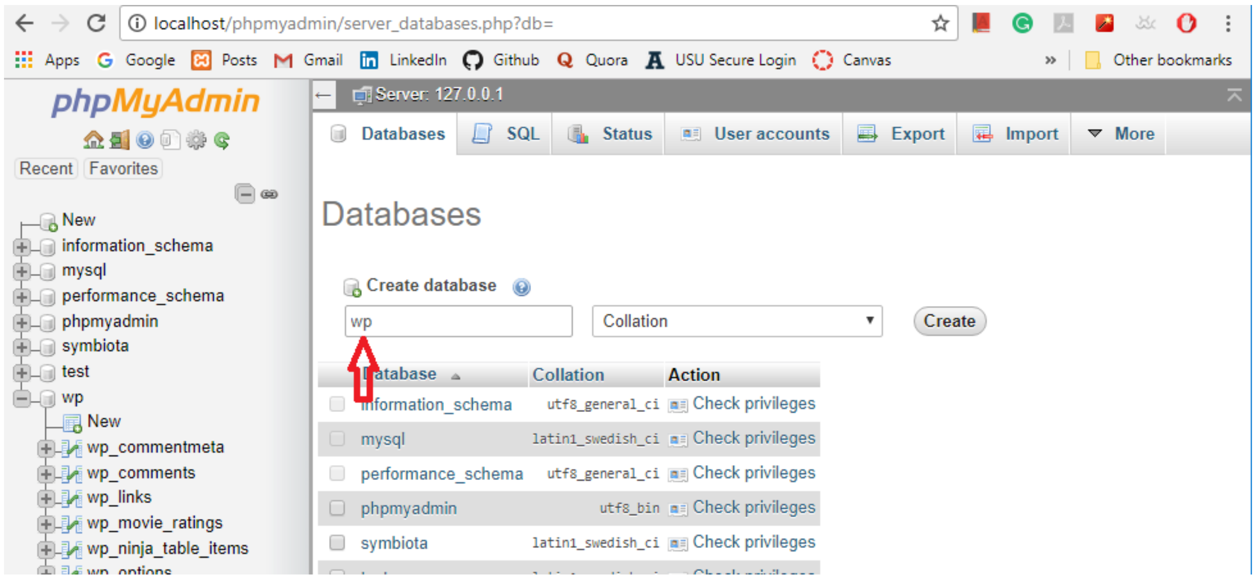

XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]
XAMPP Control Panel v3.2.2
Config
Netstat
Shell
Explorer
Services
Help
Quit
Modules

Service	Module	PID(s)	Port(s)	Actions			
	Apache	7664 6316	80, 443	Stop	Admin	Config	Logs
	MySQL	5260	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
12:31:50 PM  [Apache]   Attempting to start Apache app...
12:31:51 PM  [Apache]   Status change detected: running
12:31:52 PM  [Apache]   Attempting to stop Apache (PID: 12348)
12:31:52 PM  [Apache]   Status change detected: stopped
12:32:20 PM  [Apache]   Attempting to start Apache app...
12:32:21 PM  [Apache]   Status change detected: running
12:32:22 PM  [mysql]    Attempting to start MySQL app...
12:32:23 PM  [mysql]    Status change detected: running
```

Figure A.6: XAMPP Control Panel - Click on MySQL admin to launch PHPMyAdmin page

Click on databased at the top and enter a name of your database and click create to create a database.



New
information_schema
mysql
performance_schema
phpmyadmin
symbiota
test
wp
New
wp_commentmeta
wp_comments
wp_links
wp_movie_ratings
wp_ninja_table_items
wp_options

Databases

Create database

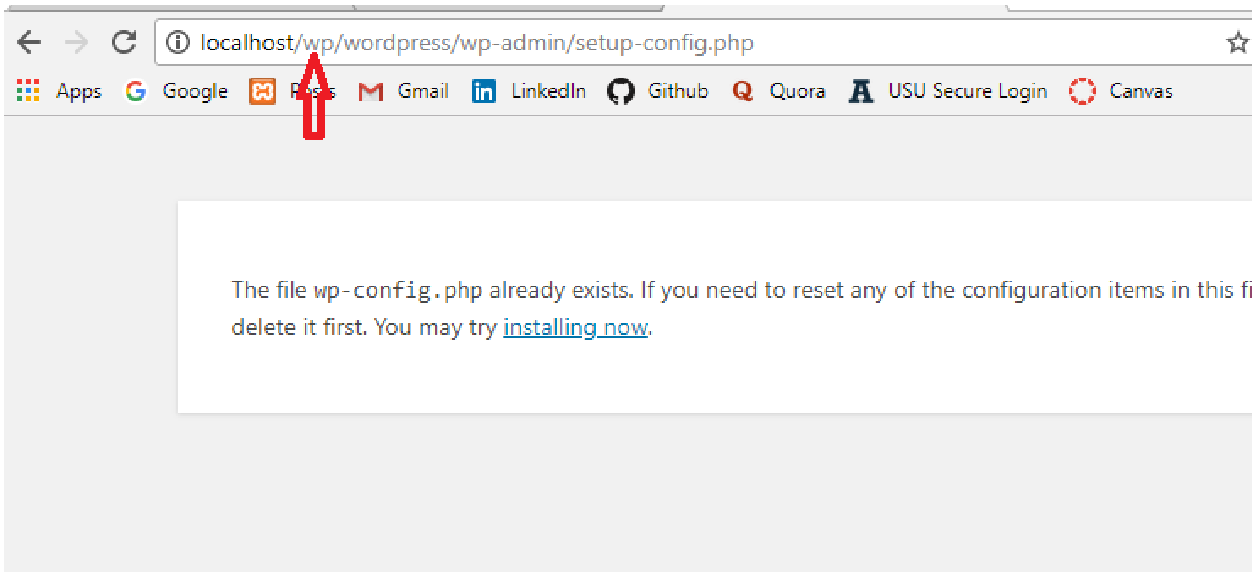
wp

Database	Collation	Action
information_schema	utf8_general_ci	Check privileges
mysql	latin1_swedish_ci	Check privileges
performance_schema	utf8_general_ci	Check privileges
phpmyadmin	utf8_bin	Check privileges
symbiota	latin1_swedish_ci	Check privileges

Figure A.7: Create a database named wp in PHPMyAdmin page

Step 5: Run the WordPress installer

To run the WordPress installer, the user must visit the test site which is http://localhost/FOLDERNAME:



The file wp-config.php already exists. If you need to reset any of the configuration items in this fi
delete it first. You may try installing now.

Figure A.8: The test site to run the WordPress installer

On the next screen, the user should enter the database name, username and password details.
Database name: Name of the database created in PHPMyAdmin ("wp")
Username: "root"
Password: leave blank
A brand-new WordPress install will be seen on the browser after completing the above process.

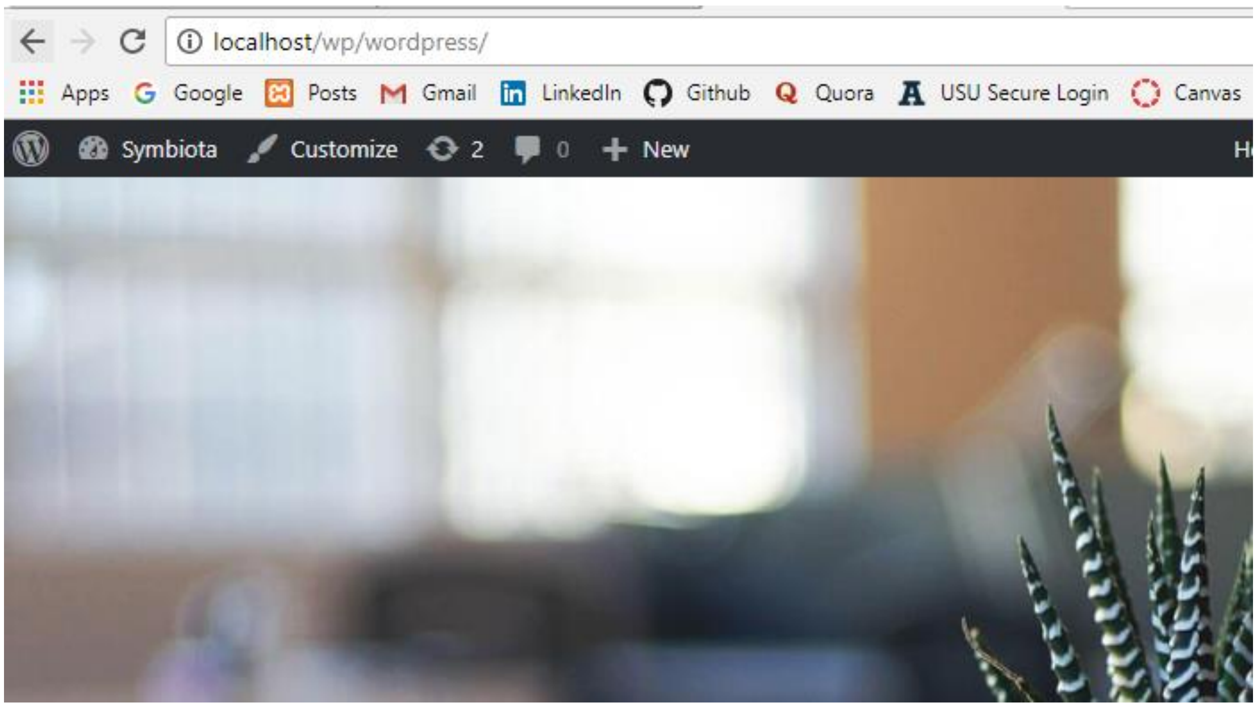
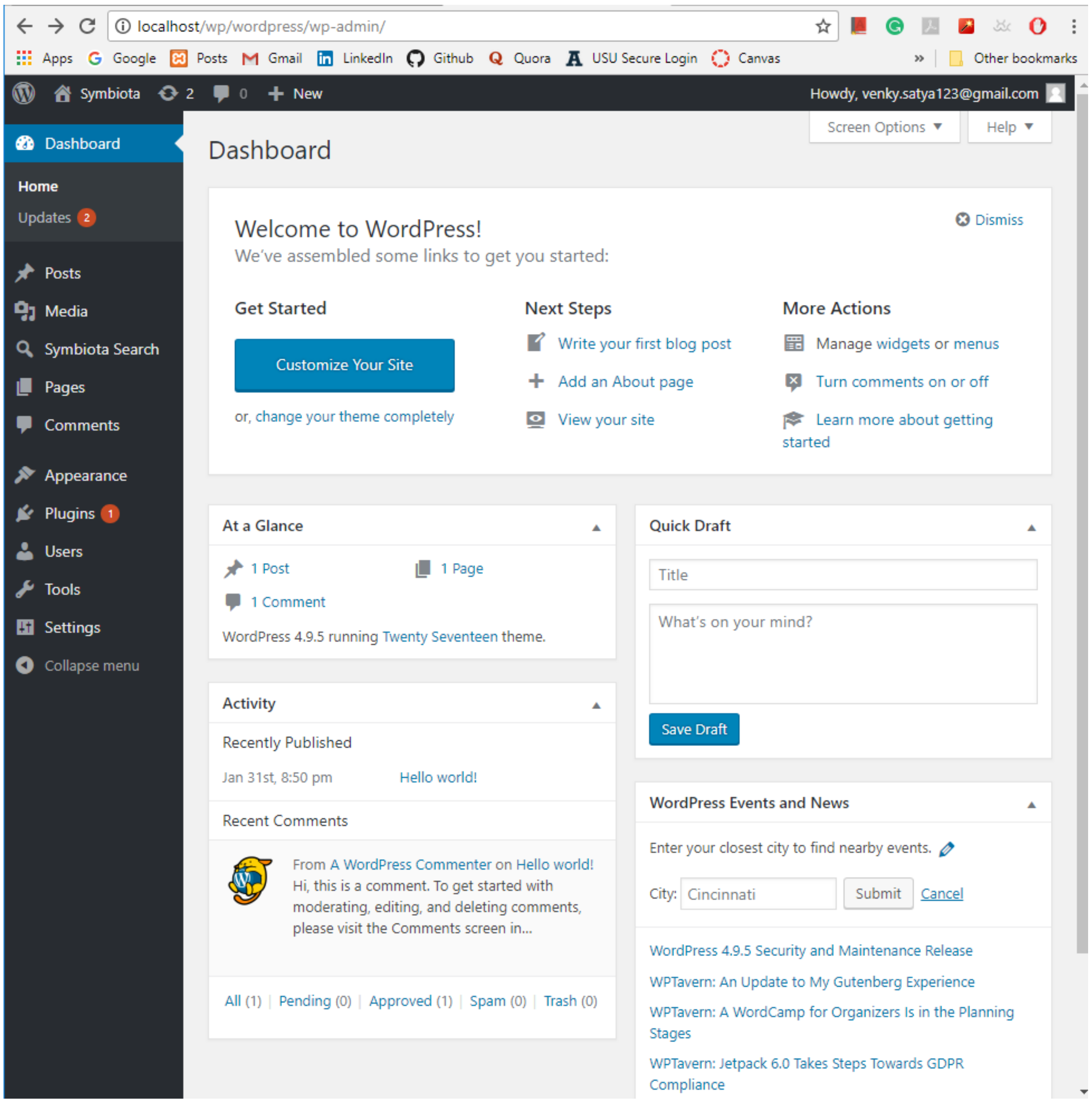


Figure A.9: WordPress website



Dashboard

Welcome to WordPress!

We've assembled some links to get you started:

Get Started

or, change your theme completely

Next Steps

Write your first blog post

Add an About page

View your site

More Actions

Manage widgets or menus

Turn comments on or off

Learn more about getting started

At a Glance

1 Post 1 Page

1 Comment

WordPress 4.9.5 running Twenty Seventeen theme.

Activity

Recently Published

Jan 31st, 8:50 pm Hello world!

Recent Comments

From A WordPress Commenter on Hello world!
Hi, this is a comment. To get started with moderating, editing, and deleting comments, please visit the Comments screen in…

All (1) | Pending (0) | Approved (1) | Spam (0) | Trash (0)

Quick Draft

Title

What's on your mind?

WordPress Events and News

Enter your closest city to find nearby events.

City: Cincinnati

WordPress 4.9.5 Security and Maintenance Release

WPTavern: An Update to My Gutenberg Experience

WPTavern: A WordCamp for Organizers Is in the Planning Stages

WPTavern: Jetpack 6.0 Takes Steps Towards GDPR Compliance

Figure A.10: The WordPress dashboard to create new WordPress websites or plugins or themes

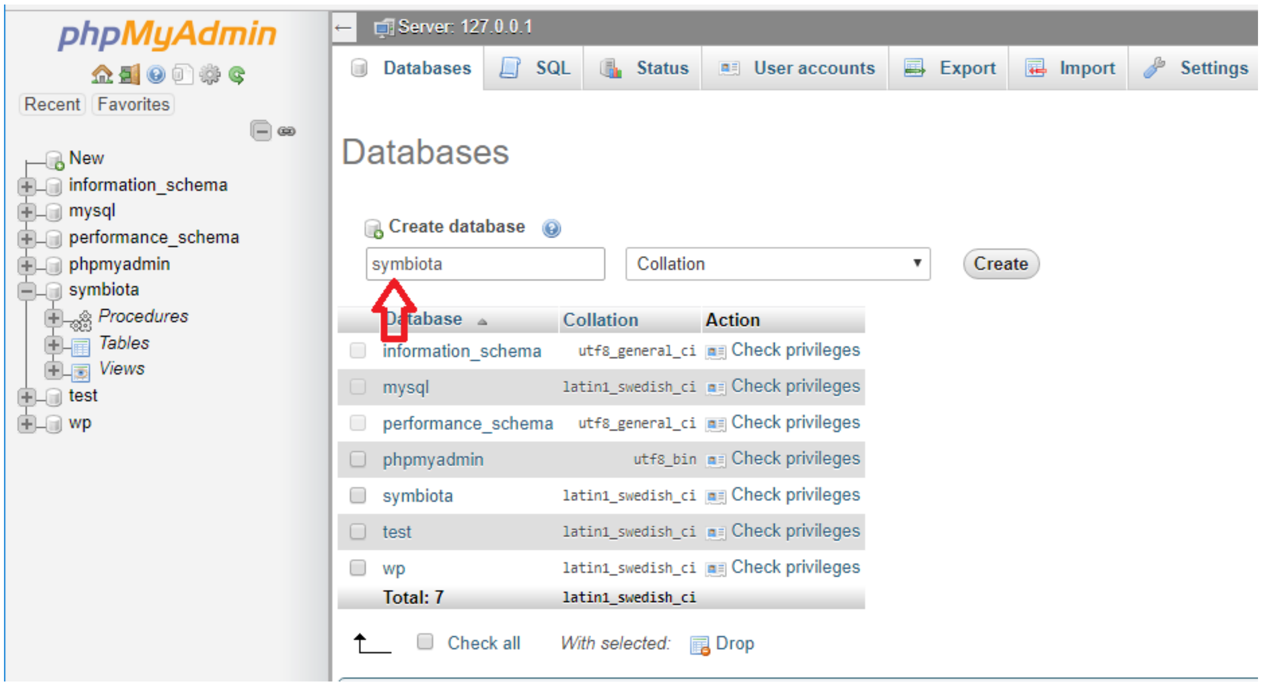
APPENDIX B

B.1 Symbiota Installation:

Step 1: Download Symbiota code from GitHub repository.
Link: https://github.com/Symbiota/Symbiota
Command Line Checkout: git clone https://github.com/Symbiota/Symbiota.git

Step 2: Install Symbiota database schema

a) Create a new database symbiota in PHPMyAdmin.



Databases

Create database

symbiota | Collation | Create

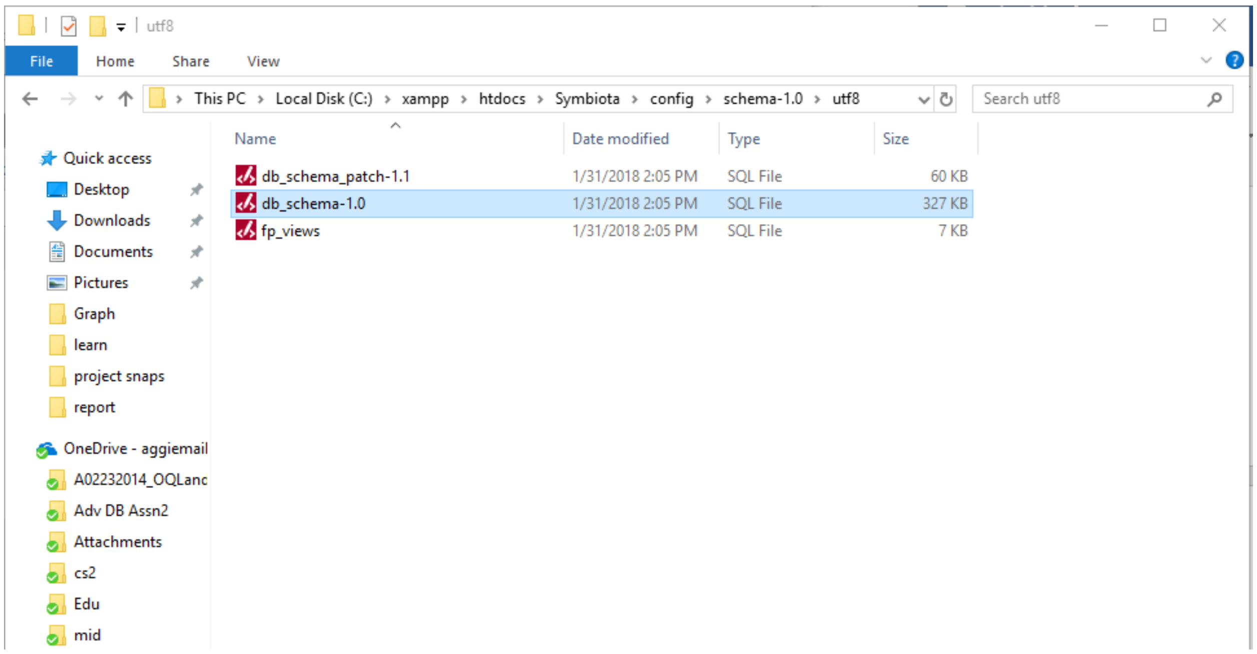
Database	Collation	Action
information_schema	utf8_general_ci	Check privileges
mysql	latin1_swedish_ci	Check privileges
performance_schema	utf8_general_ci	Check privileges
phpmyadmin	utf8_bin	Check privileges
symbiota	latin1_swedish_ci	Check privileges
test	latin1_swedish_ci	Check privileges
wp	latin1_swedish_ci	Check privileges
Total: 7	latin1_swedish_ci	

Check all *With selected:* Drop

Figure B.1: Create a database symbiota in PHPMyAdmin

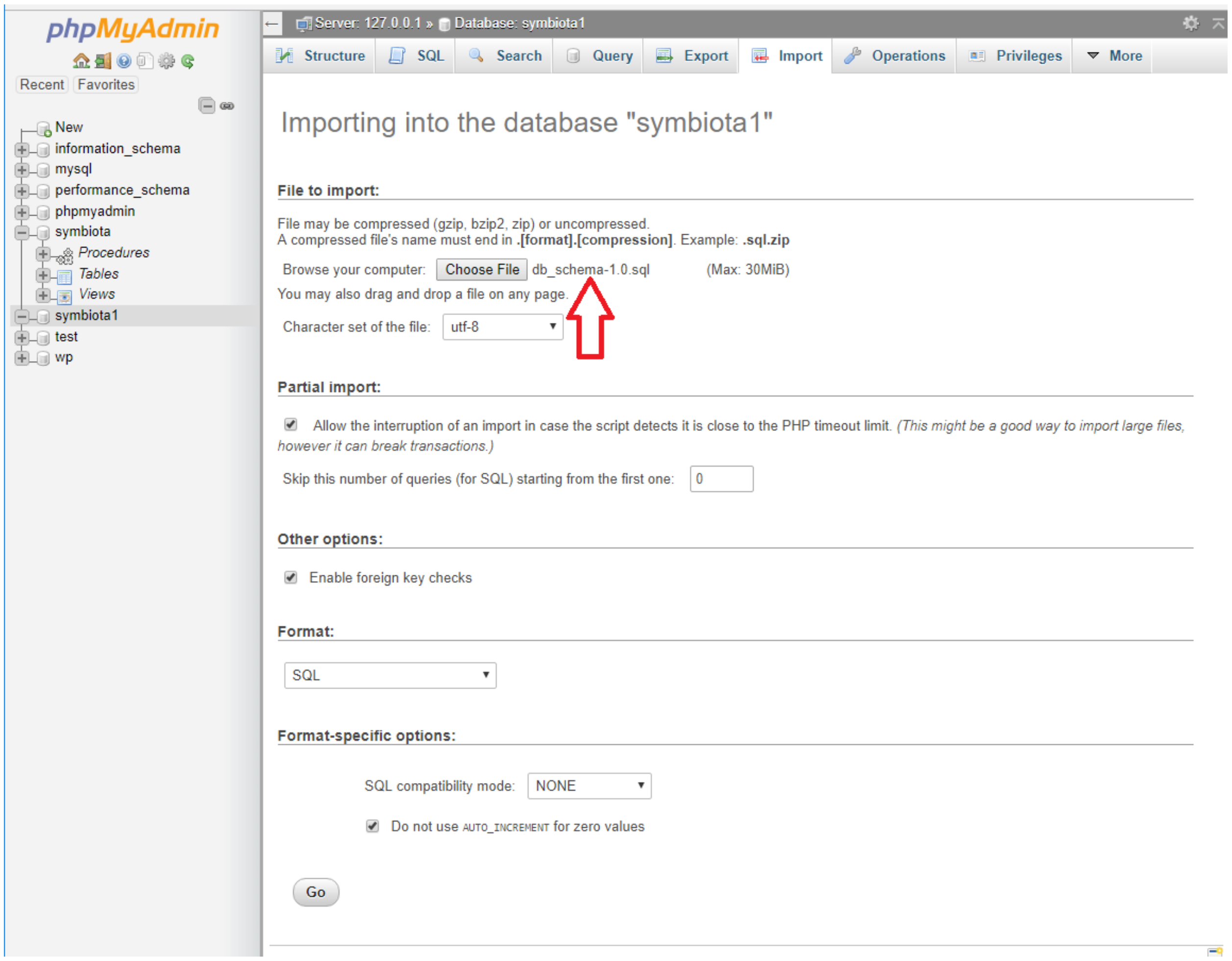
b) Create read-only and read/write users for the Symbiota database

c) Load database schema from scripts. Schema definition files are in <symbiota base folder>/config/schema-1.0/utf-8/. Run db_schema-1.0.sql to install core table structure. By default, the database is assumed to be configured to a UTF8 character set.



Name	Date modified	Type	Size
db_schema_patch-1.1	1/31/2018 2:05 PM	SQL File	60 KB
db_schema-1.0	1/31/2018 2:05 PM	SQL File	327 KB
fp_views	1/31/2018 2:05 PM	SQL File	7 KB

Figure B.2: Location of db_schema-1.0.sql file



Importing into the database "symbiota1"

File to import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: Choose File db_schema-1.0.sql (Max: 30MiB)

You may also drag and drop a file on any page.

Character set of the file: utf-8

Partial import:

✓ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a good way to import large files, however it can break transactions.)*

Skip this number of queries (for SQL) starting from the first one: 0

Other options:

✓ Enable foreign key checks

Format:

SQL

Format-specific options:

SQL compatibility mode: NONE

✓ Do not use AUTO_INCREMENT for zero values

Figure B.3: Run the db_schema-1.0.sql file in the Import option available in PHPMyAdmin to install the core table structure

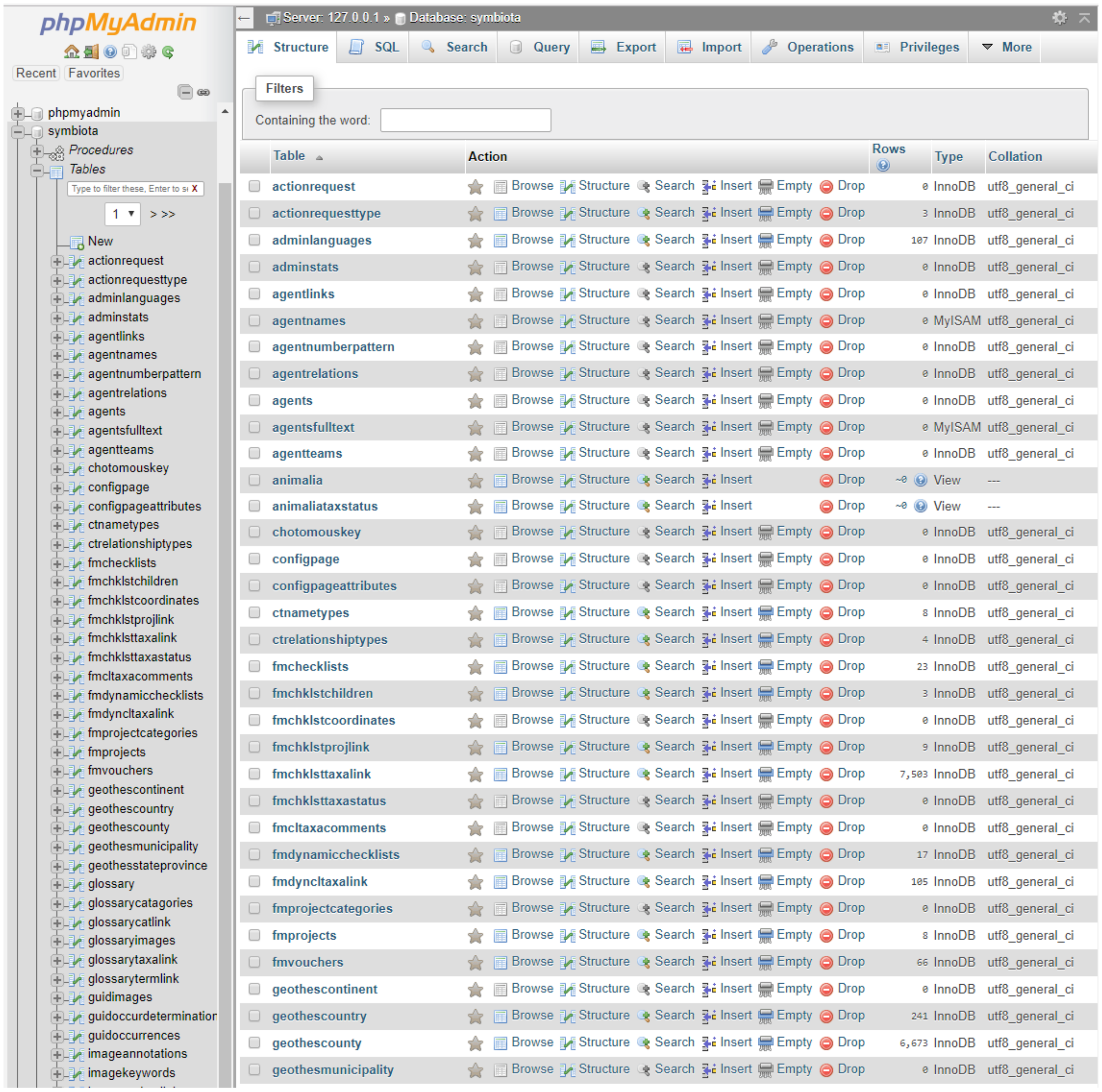


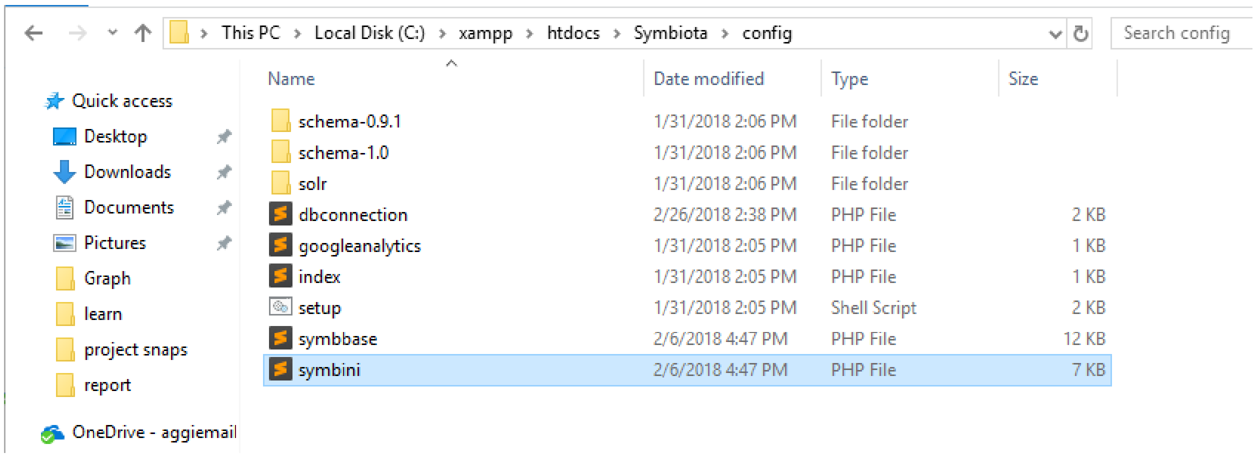
Table	Action	Rows	Type	Collation
actionrequest	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
actionrequesttype	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_general_ci
adminlanguages	Browse Structure Search Insert Empty Drop	107	InnoDB	utf8_general_ci
adminstats	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
agentlinks	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
agentnames	Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci
agentnumberpattern	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
agentrelations	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
agents	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
agentsfulltext	Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci
agentteams	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
animalia	Browse Structure Search Insert Drop	~0	View	---
animaliataxstatus	Browse Structure Search Insert Drop	~0	View	---
chotomouskey	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
configpage	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
configpageattributes	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
ctnametypes	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8_general_ci
ctrelationshiptypes	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci
fmchecklists	Browse Structure Search Insert Empty Drop	23	InnoDB	utf8_general_ci
fmchklstchildren	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_general_ci
fmchklstcoordinates	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
fmchklstprojlink	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8_general_ci
fmchklsttaxalink	Browse Structure Search Insert Empty Drop	7,503	InnoDB	utf8_general_ci
fmchklsttaxastatus	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
fmcltaxacomments	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
fmdynamicchecklists	Browse Structure Search Insert Empty Drop	17	InnoDB	utf8_general_ci
fmdyncltaxalink	Browse Structure Search Insert Empty Drop	105	InnoDB	utf8_general_ci
fmprojectcategories	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
fmprojects	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8_general_ci
fmvouchers	Browse Structure Search Insert Empty Drop	66	InnoDB	utf8_general_ci
geothescontinent	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci
geothescountry	Browse Structure Search Insert Empty Drop	241	InnoDB	utf8_general_ci
geothescounty	Browse Structure Search Insert Empty Drop	6,673	InnoDB	utf8_general_ci
geothesmunicipality	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci

Figure B.4: The core table structure of symbiota database after running the db_schema-1.0.sql

d) Run database patch scripts to bring the database up to current structure. The scripts should be run in the correct order.

Step 3: Configure the Symbiota portal by modifying the following configuration files.

a) Symbiota configuration – rename
Rename the /config/ symbini_template.php to /config/symbini.php



This PC › Local Disk (C:) › xampp › htdocs › Symbiota › config

Search config

Quick access
Desktop
Downloads
Documents
Pictures
Graph
learn
project snaps
report
OneDrive - aggiemail

Name	Date modified	Type	Size
schema-0.9.1	1/31/2018 2:06 PM	File folder	
schema-1.0	1/31/2018 2:06 PM	File folder	
solr	1/31/2018 2:06 PM	File folder	
dbconnection	2/26/2018 2:38 PM	PHP File	2 KB
googleanalytics	1/31/2018 2:05 PM	PHP File	1 KB
index	1/31/2018 2:05 PM	PHP File	1 KB
setup	1/31/2018 2:05 PM	Shell Script	2 KB
symbbase	2/6/2018 4:47 PM	PHP File	12 KB
symbini	2/6/2018 4:47 PM	PHP File	7 KB

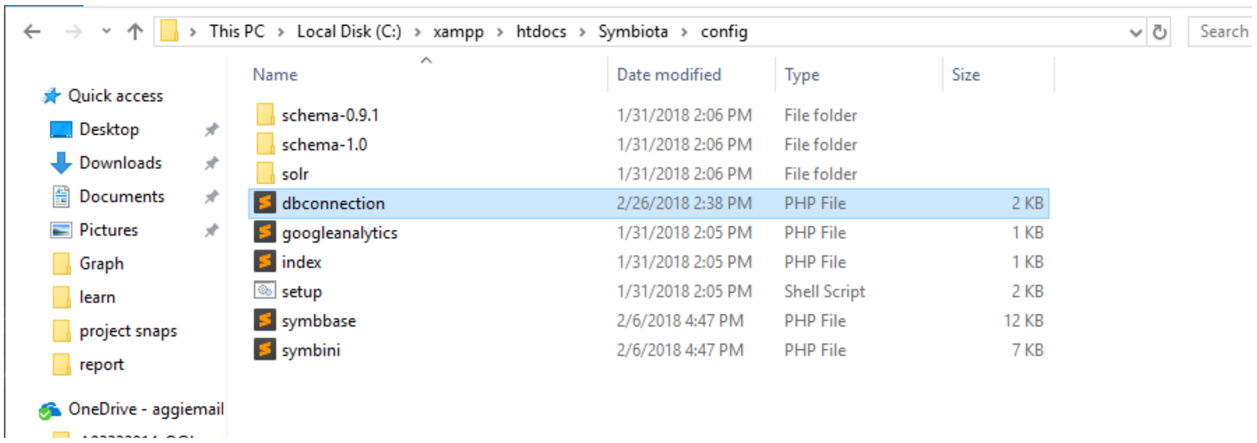
Figure B.5: Rename symbini_template.php to symbini.php

Modify the variables in symbini.php according the project environment. Change the client_root and server_root variables in symbini.php.

```
$CLIENT_ROOT = '/Symbiota';                    //URL path to project root
folder

$SERVER_ROOT = 'C:/xampp/htdocs/Symbiota';
```

b) Database connection – rename
Rename /config/dbconnection_template.php to /config/dbconnection.php



This PC › Local Disk (C:) › xampp › htdocs › Symbiota › config

Search

Quick access
Desktop
Downloads
Documents
Pictures
Graph
learn
project snaps
report
OneDrive - aggiemail

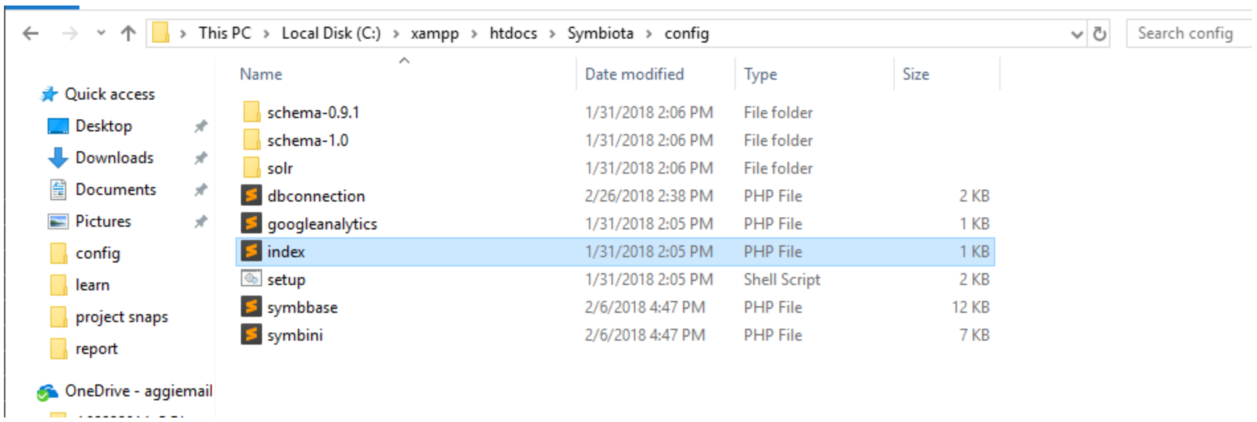
Name	Date modified	Type	Size
schema-0.9.1	1/31/2018 2:06 PM	File folder	
schema-1.0	1/31/2018 2:06 PM	File folder	
solr	1/31/2018 2:06 PM	File folder	
dbconnection	2/26/2018 2:38 PM	PHP File	2 KB
googleanalytics	1/31/2018 2:05 PM	PHP File	1 KB
index	1/31/2018 2:05 PM	PHP File	1 KB
setup	1/31/2018 2:05 PM	Shell Script	2 KB
symbbase	2/6/2018 4:47 PM	PHP File	12 KB
symbini	2/6/2018 4:47 PM	PHP File	7 KB

Figure B.6: Rename /config/dbconnection_template.php to /config/dbconnection.php

Modify with read only and read/write logs, username, password, database, port number and schema names.

```
static $SERVERS = array(
    array(
                    'type' => 'readonly',
                    'host' => 'localhost',
                    'username' => 'root',
                    'password' => '',
                    'database' => 'symbiota',
                    'port' => '3306',
                    'charset' => 'utf8'              //utf8, latin1, latin2, etc
     ),
     array(
                    'type' => 'write',
                    'host' => 'localhost',
                    'username' => 'root',
                    'password' => '',
                    'database' => 'symbiota',
                    'port' => '3306',
                    'charset' => 'utf8'
     )
  );
```

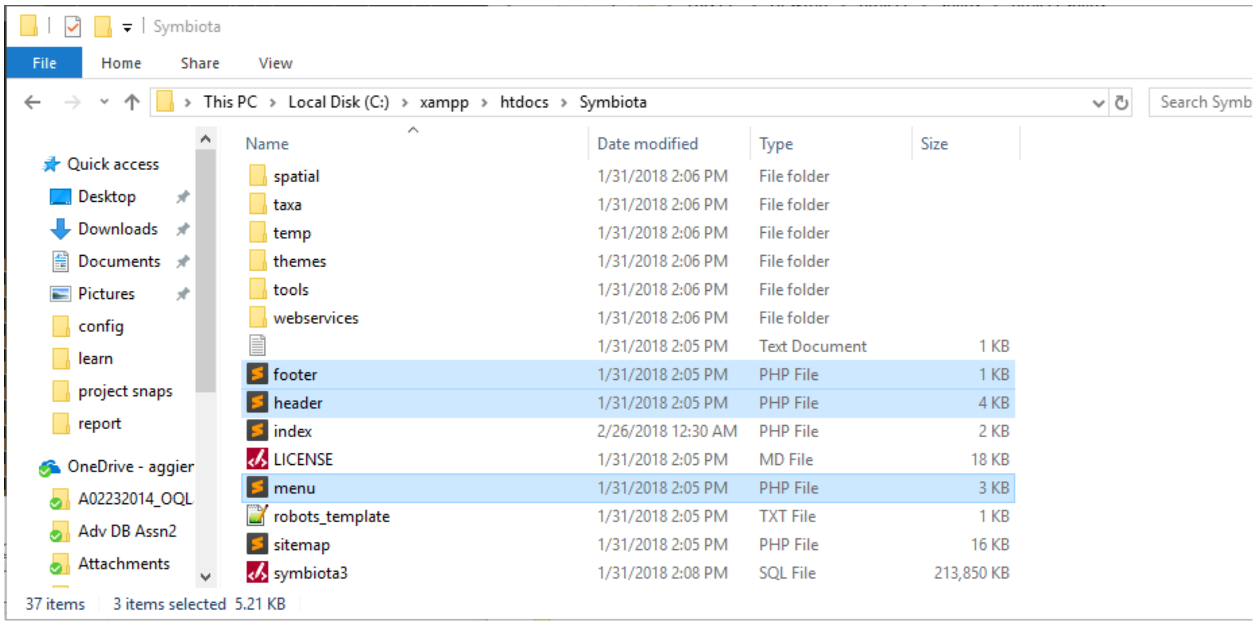
c) Homepage – rename
Rename /index_template.php to index.php. This is the homepage to which will needs introductory text graphics etc.


This PC › Local Disk (C:) › xampp › htdocs › Symbiota › config

Name	Date modified	Type	Size
schema-0.9.1	1/31/2018 2:06 PM	File folder	
schema-1.0	1/31/2018 2:06 PM	File folder	
solr	1/31/2018 2:06 PM	File folder	
dbconnection	2/26/2018 2:38 PM	PHP File	2 KB
googleanalytics	1/31/2018 2:05 PM	PHP File	1 KB
index	1/31/2018 2:05 PM	PHP File	1 KB
setup	1/31/2018 2:05 PM	Shell Script	2 KB
symbbase	2/6/2018 4:47 PM	PHP File	12 KB
symbini	2/6/2018 4:47 PM	PHP File	7 KB

Figure B.7: Rename /index_template.php to index.php

d) Home page
header.php and footer.php are used by all pages to establish uniform layout and menu.php is needed if a left menu is preferred. Within the base installation folder, rename header_template.php to header.php and footer_template.php to footer.php.



Name	Date modified	Type	Size
spatial	1/31/2018 2:06 PM	File folder	
taxa	1/31/2018 2:06 PM	File folder	
temp	1/31/2018 2:06 PM	File folder	
themes	1/31/2018 2:06 PM	File folder	
tools	1/31/2018 2:06 PM	File folder	
webservices	1/31/2018 2:06 PM	File folder	
	1/31/2018 2:05 PM	Text Document	1 KB
footer	1/31/2018 2:05 PM	PHP File	1 KB
header	1/31/2018 2:05 PM	PHP File	4 KB
index	2/26/2018 12:30 AM	PHP File	2 KB
LICENSE	1/31/2018 2:05 PM	MD File	18 KB
menu	1/31/2018 2:05 PM	PHP File	3 KB
robots_template	1/31/2018 2:05 PM	TXT File	1 KB
sitemap	1/31/2018 2:05 PM	PHP File	16 KB
symbiota3	1/31/2018 2:08 PM	SQL File	213,850 KB

Figure B.8: Rename header_template.php and footer_template.php to header.php and footer.php

- header.php: Within this file, change /images/layout/defaultheader.jpg to /images/layout/header.jpg. Add the header to /images/layout/folder. Change /images/layout/defaultleftstrip.gif as desired. Establishing the header using an image is easy, yet more complex header configurations are possible.

- footer.php: modify as you did with header.php file.

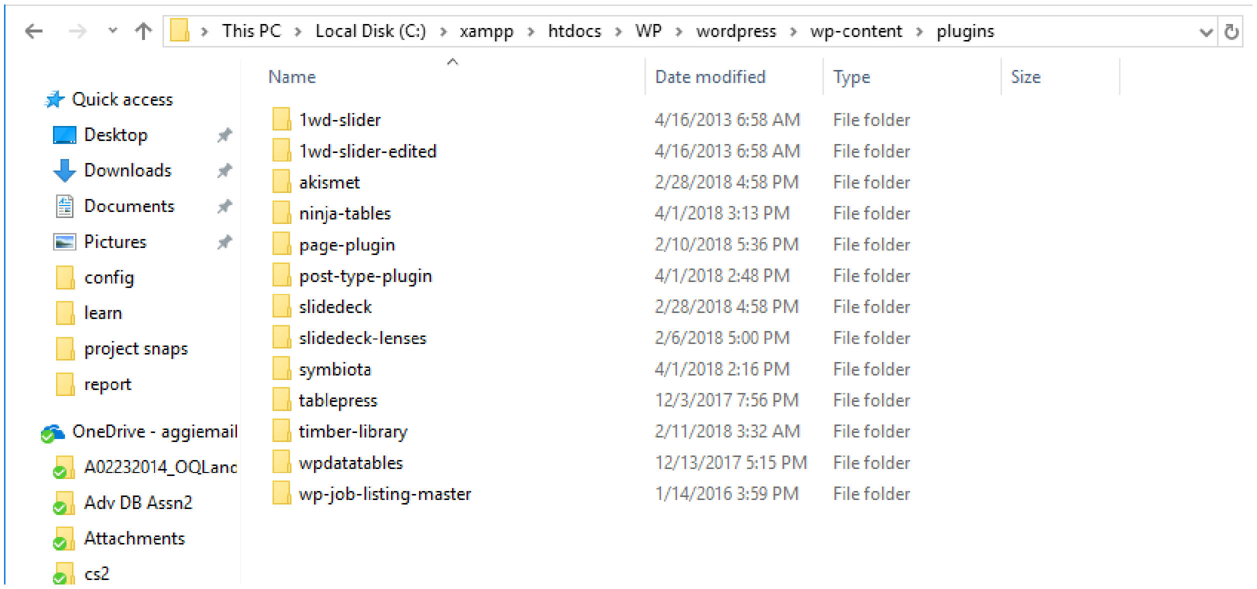
Step 4: File permissions: the web server needs write access to the files and folders mentioned below.

All folders in /temp
/webservices/dwc/rss.xml
/content/collicon/
/content/dwca/
/content/logs/

APPENDIX C

C.1 WordPress plugin Installation:

Plugins provides additional functionality to the WordPress application. To install a plugin, the user must put plugin files into the wp-content/plugins/ directory. The user can activate it or deactivate it after keeping them in the plugins directory.

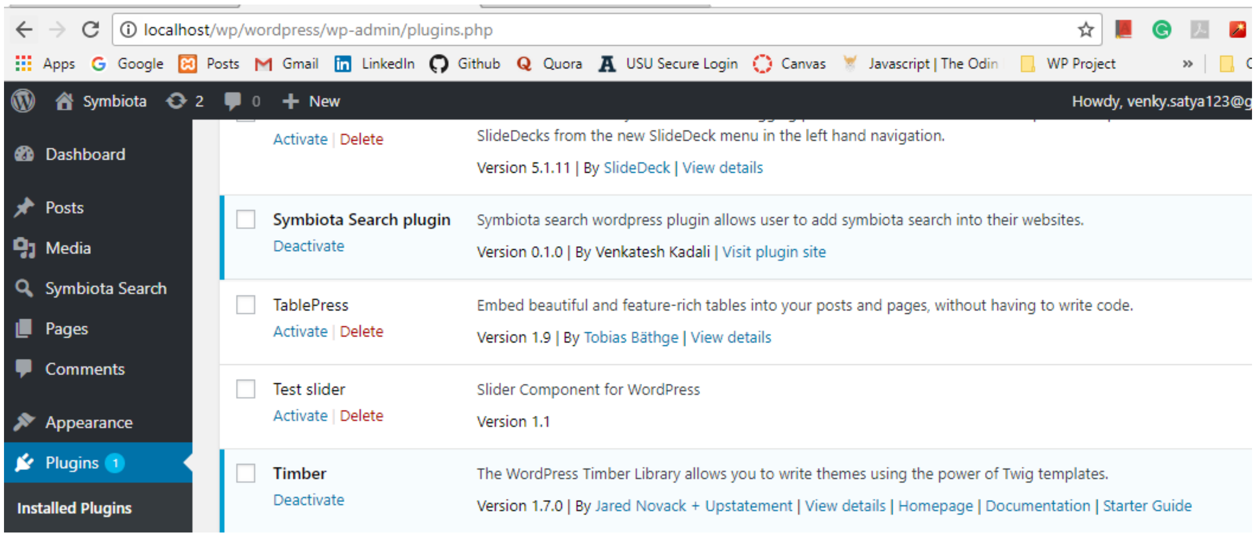


This PC › Local Disk (C:) › xampp › htdocs › WP › wordpress › wp-content › plugins

Quick access
Desktop
Downloads
Documents
Pictures
config
learn
project snaps
report
OneDrive - aggiemail
A02232014_OQLanc
Adv DB Assn2
Attachments
cs2

Name	Date modified	Type	Size
1wd-slider	4/16/2013 6:58 AM	File folder	
1wd-slider-edited	4/16/2013 6:58 AM	File folder	
akismet	2/28/2018 4:58 PM	File folder	
ninja-tables	4/1/2018 3:13 PM	File folder	
page-plugin	2/10/2018 5:36 PM	File folder	
post-type-plugin	4/1/2018 2:48 PM	File folder	
slidedeck	2/28/2018 4:58 PM	File folder	
slidedeck-lenses	2/6/2018 5:00 PM	File folder	
symbiota	4/1/2018 2:16 PM	File folder	
tablepress	12/3/2017 7:56 PM	File folder	
timber-library	2/11/2018 3:32 AM	File folder	
wpdatatables	12/13/2017 5:15 PM	File folder	
wp-job-listing-master	1/14/2016 3:59 PM	File folder	

Fig C.1: /wp-content/plugins folder consisting of several plugins



localhost/wp/wordpress/wp-admin/plugins.php

Apps Google Posts Gmail LinkedIn Github Quora USU Secure Login Canvas Javascript | The Odin WP Project

Symbiota 2 0 New Howdy, venky.satya123@g

Dashboard
Posts
Media
Symbiota Search
Pages
Comments
Appearance
Plugins 1
Installed Plugins

Activate | Delete	SlideDecks from the new SlideDeck menu in the left hand navigation. Version 5.1.11 | By SlideDeck | View details
Symbiota Search plugin Deactivate	Symbiota search wordpress plugin allows user to add symbiota search into their websites. Version 0.1.0 | By Venkatesh Kadali | Visit plugin site
TablePress Activate | Delete	Embed beautiful and feature-rich tables into your posts and pages, without having to write code. Version 1.9 | By Tobias Bäthge | View details
Test slider Activate | Delete	Slider Component for WordPress Version 1.1
Timber Deactivate	The WordPress Timber Library allows you to write themes using the power of Twig templates. Version 1.7.0 | By Jared Novack + Upstatement | View details | Homepage | Documentation | Starter Guide

Figure C.2: The user can activate and deactivate the plugins after installing them