Final Report

eventPress

Hosting an event made easy.

A wordpress plugin

Benjamin Healey

Features and Specifications

eventPress is a wordpress plugin that has two roles; users and hosts. Users and hosts can make events and display them on their sites. The main function is to be that users can press a button and send their event to a host. The host then receives the event and can approve or deny the event to be shown on the hosts' sites. The host also displays the events in a way that makes it easy for visitors to sort through lots of events by location or other criteria.

The plugin works (in broad strokes) by creating a custom post type called "event", it would then call a jQuery post command to post the event to a dedicated ASP (server file) that then takes the event and creates it local on the hosts' page.

User Stories

"As a magazine website owner, I want (for our magazine website) an events page that collects events from our clients, and other users sites in Hamilton, the page needs to be mobile friendly and display the events in a simple and searchable way. We would also like some easy way of authenticating events and filtering out spam (irrelevant) events, it would also be great to easily contact back all of the people who posted events (we have had problems like that in the past). It would also be cool to potentially have a map but I am not crazy about that layout, you don't go to an event just because it's close by, you go to an event that interests you."

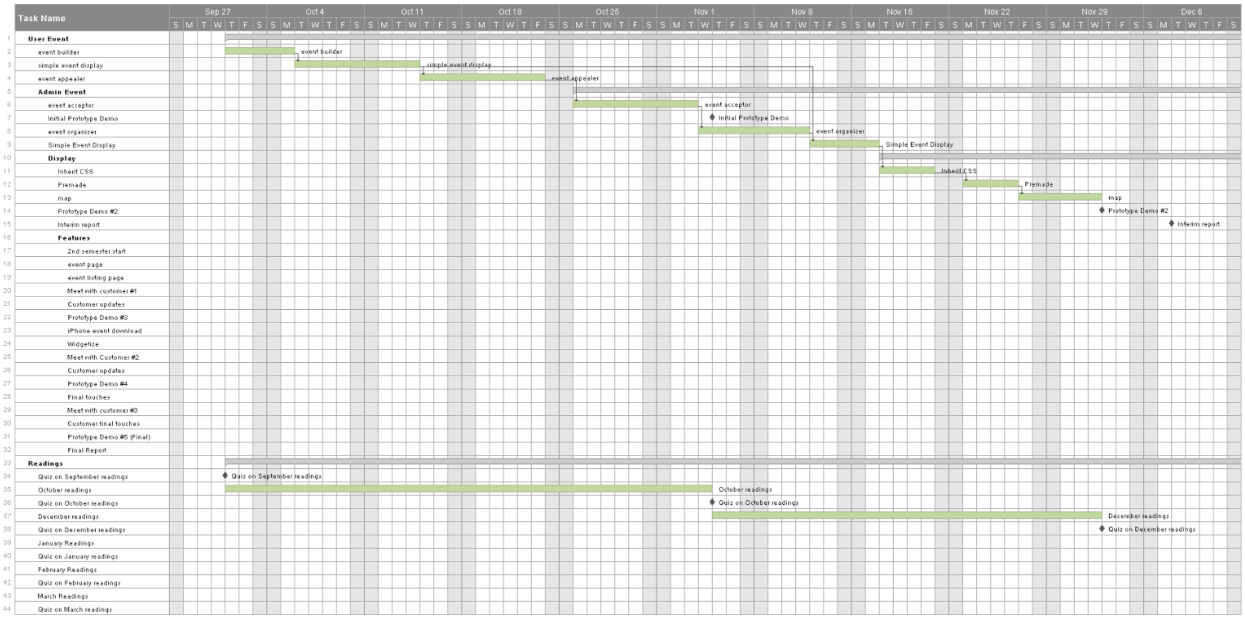
- Martinus Geleynse (Editor of Urbanicity)

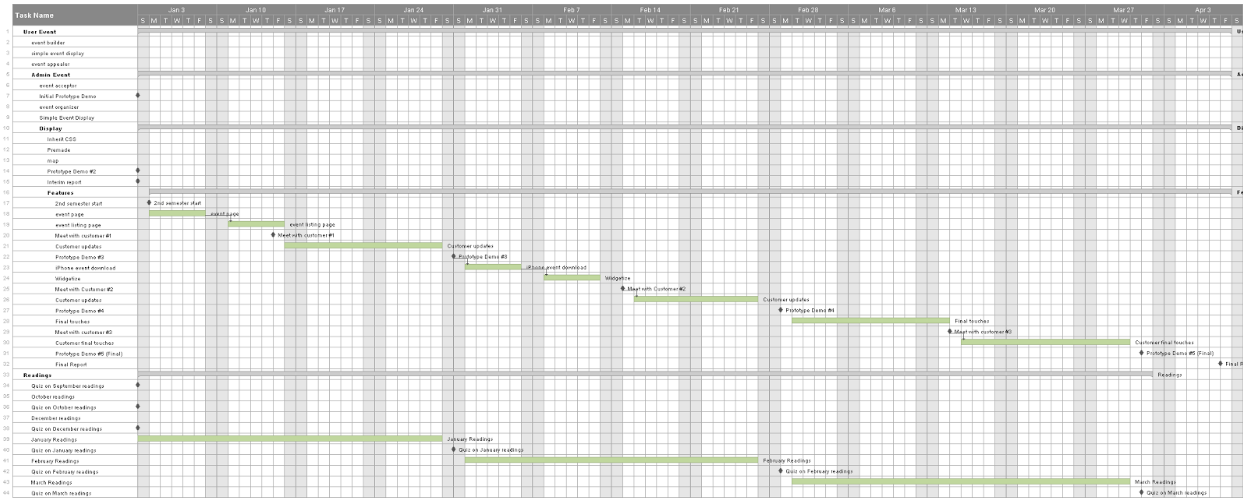
"As a website designer, what we need for our clients is a simple calendar/events plugin that is highly usable. It needs to not have an overwhelming amount of fields. It also needs to be naturally fitting in the site it's implemented in and then easily customized, if needed. I think it would be really helpful to have the fields load based on what you have previously input, why make it more confusing than it needs to be."

- Martinus Geleynse (CEO of Urbanicity Omnimedia)

At the interim report (December 2015) Mr. Geleynse and I had discussed extending a different plugin to include the functionality we want. At the time this seemed wise but it proved too difficult to adapt to another plugin and seemed easier to create our own plugin from scratch.

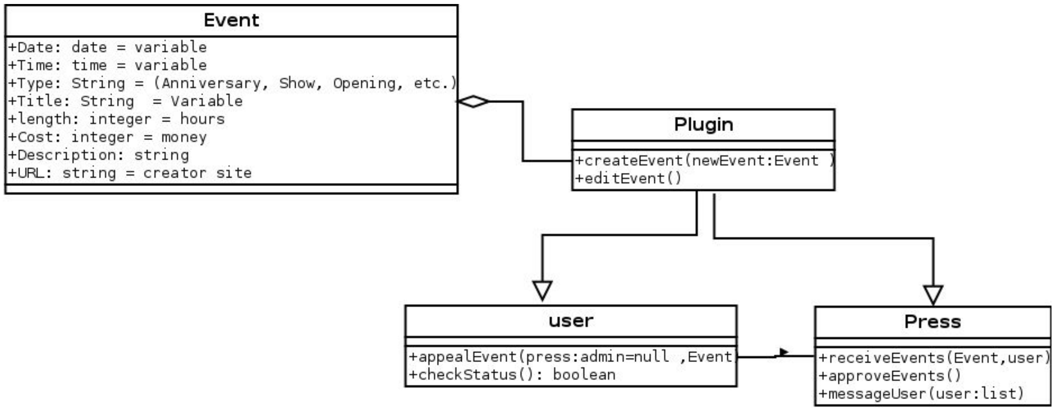
Gantt Chart



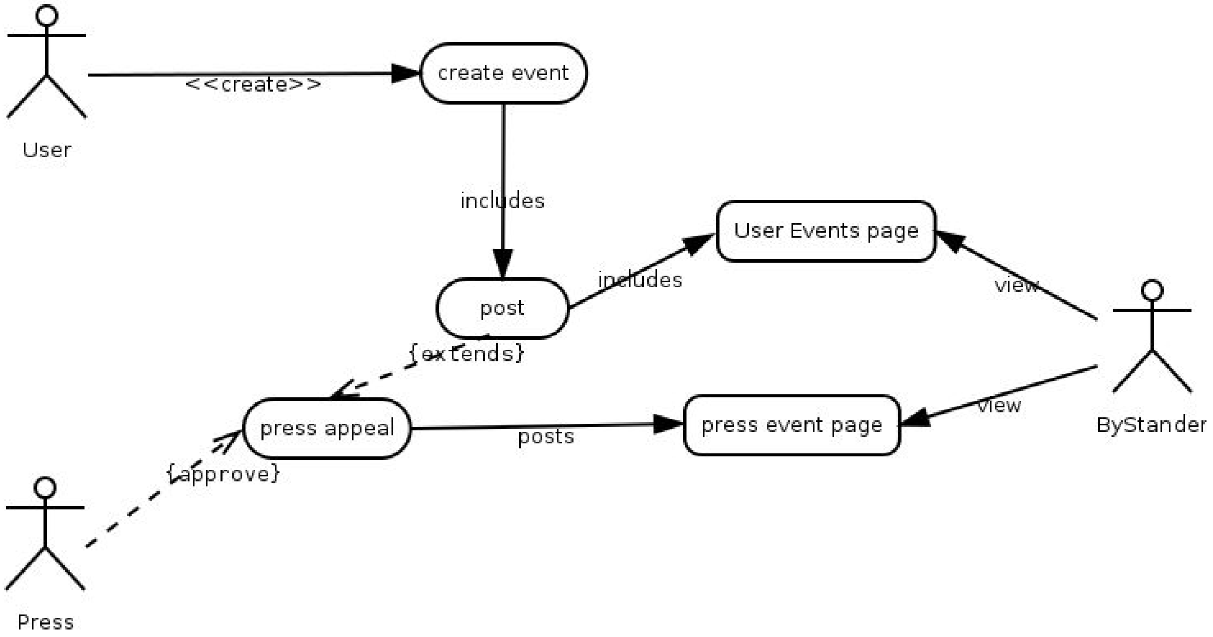


UML use-case diagrams

Class diagram



Use-Case Diagram



Source Code

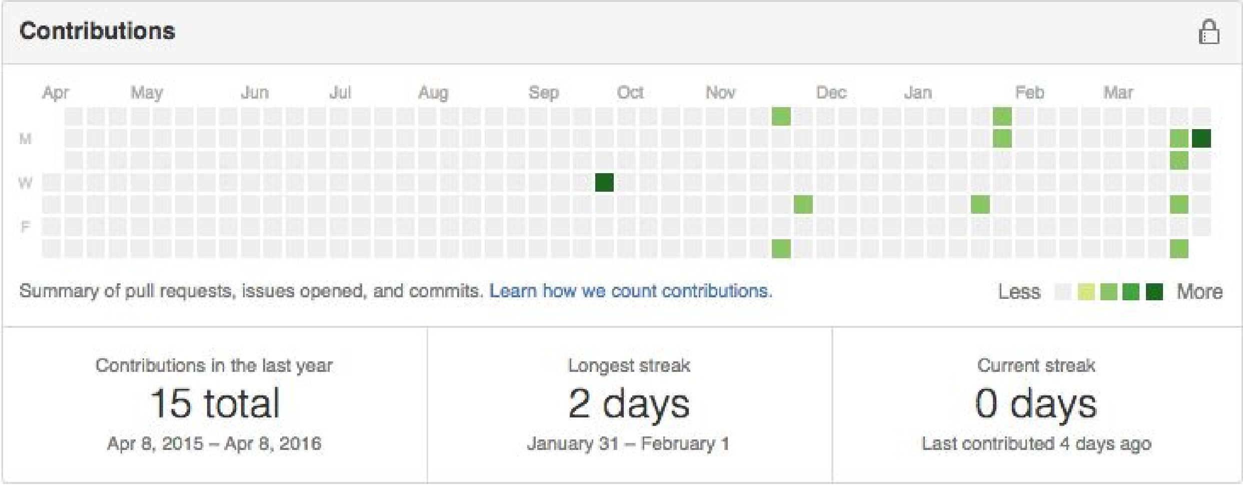
event_press.php

```
<?php
    /*
    Plugin Name: Event Press
    Plugin URI: www.cs.redeemer.ca/~bhealey
    Description: Plugin for creating, displaying, and sharing events.
    Author: Benjamin Healey
    Author URI: www.cs.redeemer.ca/~bhealey
    Version: 1.1
  */
require_once dirname( __FILE__ ).'/includes/ep-main.php';

register_activation_hook (__FILE__, array ('bh_ep_main', 'bh_ep_install')); //on
install, do stuff
register_deactivation_hook (__FILE__, array ('bh_ep_main', 'bh_ep_uninstall'));
//uninstall action hook

add_action( 'init', 'bh_ep_event_post_type' ); //custom post type, bh_ep_event
add_action( 'add_meta_boxes', 'bh_ep_add_event_info_metabox' ); //add info to custom
post type
add_action( 'save_post', 'bh_ep_save_event_info' ); //save functionality for new info
on bh_ep_event
add_action('admin_menu', 'bh_ep_menu'); //add menu
add_action('init', 'import_script');
//IMPORT SHARE JavaScript File
function import_script(){
       wp_enqueue_script('script', plugins_url('js/send.js', __FILE__),
array('jquery'));
}
//create event post type
function bh_ep_event_post_type(){
       //array of arguments to be passed to register
       $labels = array(
              'name'               => 'Events',
              'singular_name'      => 'Event',
              'menu_name'          => 'Events',
              'name_admin_bar'     => 'Event',
              'add_new'            => 'Add New Event',
              'add_new_item'       => 'Add New Event' ,
              'new_item'           => 'New Event',
              'edit_item'          => 'Edit Event',
              'view_item'          => 'View Event',
              'all_items'          => 'All Events',
              'search_items'       => 'Search Events',
              'parent_item_colon'  => 'Parent Events:',
              'not_found'          => 'No events found.',
              'not_found_in_trash' => 'No events found in Trash.'
       );
       $support = array(
              'title',
              'editor',
              'thumbnail'
       );
       $args = array(
              'labels'             => $labels,
                     'description'        => __( 'Description.'),
              'public'             => true,
              'publicly_queryable' => true,
              'show_ui'            => true, //default UI
              'show_in_menu'       => true, //show in menu
              'query_var'          => "event",
              'menu_icon'       => 'dashicons-calendar-alt',
              'rewrite'            => array( 'slug' => 'event' ),
              'has_archive'        => true,
              'hierarchical'       => false,
              'menu_position'      => null,
              'supports'        => $support
       );
       register_post_type('bh_ep_event', $args);
}
function bh_ep_add_event_info_metabox() {
       add_meta_box(
                     'bh-ep-event-info-metabox',
              __('Event Info'),
              'bh_ep_render_event_info_metabox',
              'bh_ep_event',
             'side',
             'core'
       );
       add_meta_box(
                    'bh-ep-send-metabox',
             __('Share'),
             'bh_ep_render_send_metabox',
             'bh_ep_event',
             'normal',
             'core'
       );
}
function bh_ep_render_send_metabox( $post) {
       if (get_post_status($post) == 'auto-draft' || get_post_status($post) ==
'draft'){
             $button_text = 'Publish and Share';
       } else if (get_post_status($post) == 'publish'){
             $button_text = 'Share';
       } else {
             $button_text = 'Cannot Share in this state';
       }

        ?>
       <!-- on click call JQeury function   -->
       <p id="state">Not Sent</p>
        <button class="button button-large" onclick="sendEvent()"
id="bh-ep-event-send"><?php _e($button_text); ?></button>
<script>
function sendEvent() {
       //foo for testing, "test.asp" for testing
       var posting =
$.post('http://cs.redeemer.ca/~bhealey/wp-content/plugins/event_press/includes/test.as
p', "foo");
       document.getElementById("state").innerHTML = posting;
}
</script>
<?php
       // jQuery.post(); JSON to server
             //create Urbanicity page as server
}

function bh_ep_render_event_info_metabox( $post ) {
    // generate a nonce field
    //ensures info comes from current site
    wp_nonce_field( basename( __FILE__ ), 'bh-ep-event-info-nonce' );

    // check for previously saved event data
    $event_start_date = get_post_meta( $post->ID, 'event-start-date', true );
    $event_end_date = get_post_meta( $post->ID, 'event-end-date', true );
    $event_venue = get_post_meta( $post->ID, 'event-venue', true );
    // if there is previously saved value then retrieve it, else set it to the current
time
    $event_start_date = ! empty( $event_start_date ) ? $event_start_date : time();
       //add checks, only publishable if criteria met
    // if the end date is not present event ends on the same day
    $event_end_date = ! empty( $event_end_date ) ? $event_end_date :
$event_start_date;

    ?>
       <!-- change to select, drop-down menus  -->
<label for="bh-ep-event-start-date"><?php _e( 'Event Start Date:' ); ?></label>
        <input class="widefat bh-ep-event-date-input" id="bh-ep-event-start-date"
type="text" name="bh-ep-event-start-date" placeholder="Format: May 17, 1994"
value="<?php echo date( 'F d, Y', $event_start_date ); ?>" />
       <!-- change to select, drop down menus  -->
<label for="bh-ep-event-end-date"><?php _e( 'Event End Date:'); ?></label>
        <input class="widefat bh-ep-event-date-input" id="bh-ep-event-end-date"
type="text" name="bh-ep-event-end-date" placeholder="Format: May 17, 1994"
value="<?php echo date( 'F d, Y', $event_end_date ); ?>" />
       <!-- change to select, list of regions  -->
       <!-- add location variable, address -->
<label for="bh-ep-event-venue"><?php _e( 'Event Venue:'); ?></label>
        <input class="widefat" id="bh-ep-event-venue" type="text"
name="bh-ep-event-venue" placeholder="eg. Times Square" value="<?php echo
$event_venue; ?>" />
<?php
}

//save custom post type
function bh_ep_save_event_info( $post_id ) {
    // checking if the post being saved is an 'event',
    // if not, then return
    if ( 'bh_ep_event' != $_POST['post_type'] ) {
        return;
    }

    // checking for the 'save' status
    $is_autosave = wp_is_post_autosave( $post_id );
    $is_revision = wp_is_post_revision( $post_id );
    $is_valid_nonce = ( isset( $_POST['bh-ep-event-info-nonce'] ) && (
wp_verify_nonce( $_POST['bh-ep-event-info-nonce'], basename( __FILE__ ) ) ) ) ? true :
false;

    // exit depending on the save status or if the nonce is not valid
    if ( $is_autosave || $is_revision || ! $is_valid_nonce ) {
        return;
    }

    // checking for the values and performing necessary actions
    if ( isset( $_POST['bh-ep-event-start-date'] ) ) {
        update_post_meta( $post_id, 'event-start-date', strtotime(
$_POST['bh-ep-event-start-date'] ) );
    }

    if ( isset( $_POST['bh-ep-event-end-date'] ) ) {
        update_post_meta( $post_id, 'event-end-date', strtotime(
$_POST['bh-ep-event-end-date'] ) );
    }

    if ( isset( $_POST['bh-ep-event-venue'] ) ) {
        update_post_meta( $post_id, 'event-venue', sanitize_text_field(
$_POST['bh-ep-event-venue'] ) );
    }
}

function bh_ep_menu(){
       //custom menu
       //what page is, Title, capability, slug, function (to be called),
       add_menu_page('EventPress Page', 'EventPress', 'manage_options', __FILE__.
'bh_ep_settings', 'bh_ep_settings');
}

function bh_ep_settings(){
//change all of this
//visual components options
?>

<form action="">
<fieldset>
<h4>Personal information:</h4>
       Contact Email:<br>
<input type="email" name="press_email" value="">
<br>
       Site name:<br>
<input type="text" name="lastname" value="">
<br><br>
</fieldset>

<fieldset>
<h4>Event Criteria</h4>
       Location<br>
<input type="text" name="press_email" value="">
<br>
</fieldset>
<input type="submit" value="Submit">
</form>
<?php
}
?>
```

uninstall.php

```
<?php
register_activation_hook (_FILE_, 'bh_ep_activate'); //on install, activate uninstall
option

function bh_ep_activate(){
//register uninstall function
       register_uninstall_hook(_FILE_, 'bh_ep_uninstaller');

}

function bh_ep_uninstaller(){
       //delete anything that looks good
       delete_option('bh_ep_options');//deactivate plugin
}

?>
```

Revision control log/Commit Log



Contributions

Summary of pull requests, issues opened, and commits. Learn how we count contributions.

Contributions in the last year
15 total
Apr 8, 2015 – Apr 8, 2016

Longest streak
2 days
January 31 – February 1

Current streak
0 days
Last contributed 4 days ago

Acceptance Tests; Results

Press/admin

1. Events
   a. can users create events?
   b. can the press receive events from users?
   c. can they sort through the events?
   d. can they approve or deny the sent event?
2. Functions
   a. can the press display the events in a visually intuitive way?
      i. is it mobile friendly
   b. can the press message all of the users?
   c. can the press authenticate users?

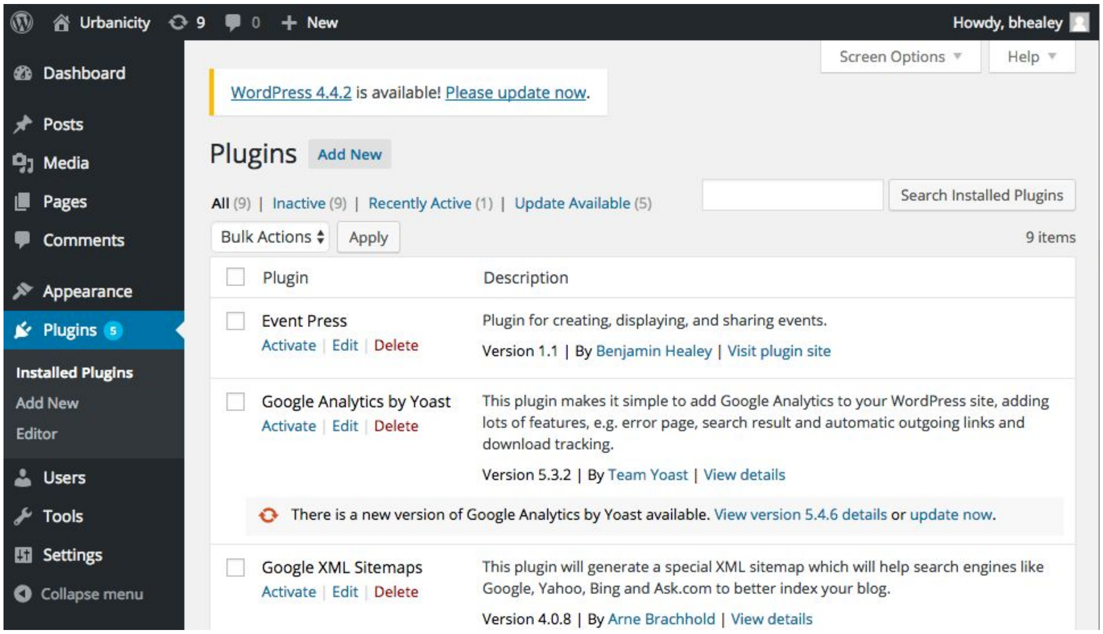
Users

1. can the user share with the press, of the correct criteria?

2. can the users and press communicate?

In the end my plugin registers a custom post type called “event” it allows users to create “events” which are essentially posts with custom meta-data. The send/receive functionality does not work.

User interface screenshots



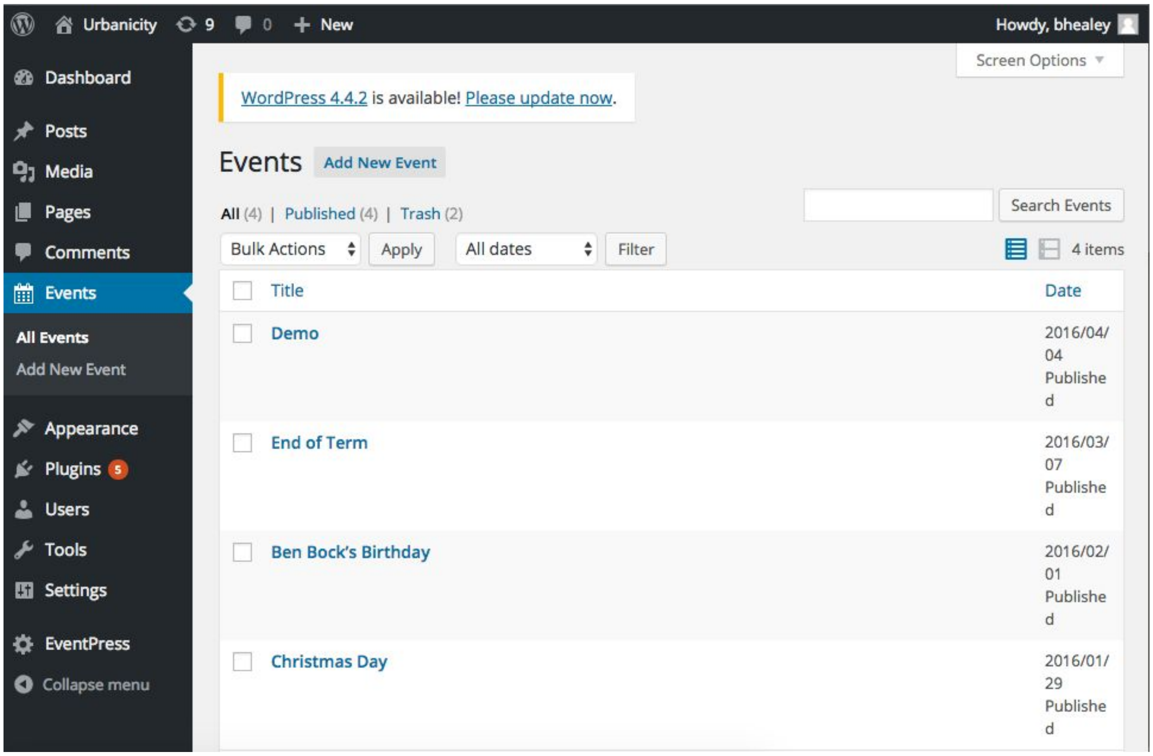
WordPress 4.4.2 is available! Please update now.

Plugins

All (9) | Inactive (9) | Recently Active (1) | Update Available (5)

9 items

□ Plugin	Description
□ Event Press Activate | Edit | Delete	Plugin for creating, displaying, and sharing events. Version 1.1 | By Benjamin Healey | Visit plugin site
□ Google Analytics by Yoast Activate | Edit | Delete	This plugin makes it simple to add Google Analytics to your WordPress site, adding lots of features, e.g. error page, search result and automatic outgoing links and download tracking. Version 5.3.2 | By Team Yoast | View details
	There is a new version of Google Analytics by Yoast available. View version 5.4.6 details or update now.
□ Google XML Sitemaps Activate | Edit | Delete	This plugin will generate a special XML sitemap which will help search engines like Google, Yahoo, Bing and Ask.com to better index your blog. Version 4.0.8 | By Arne Brachhold | View details



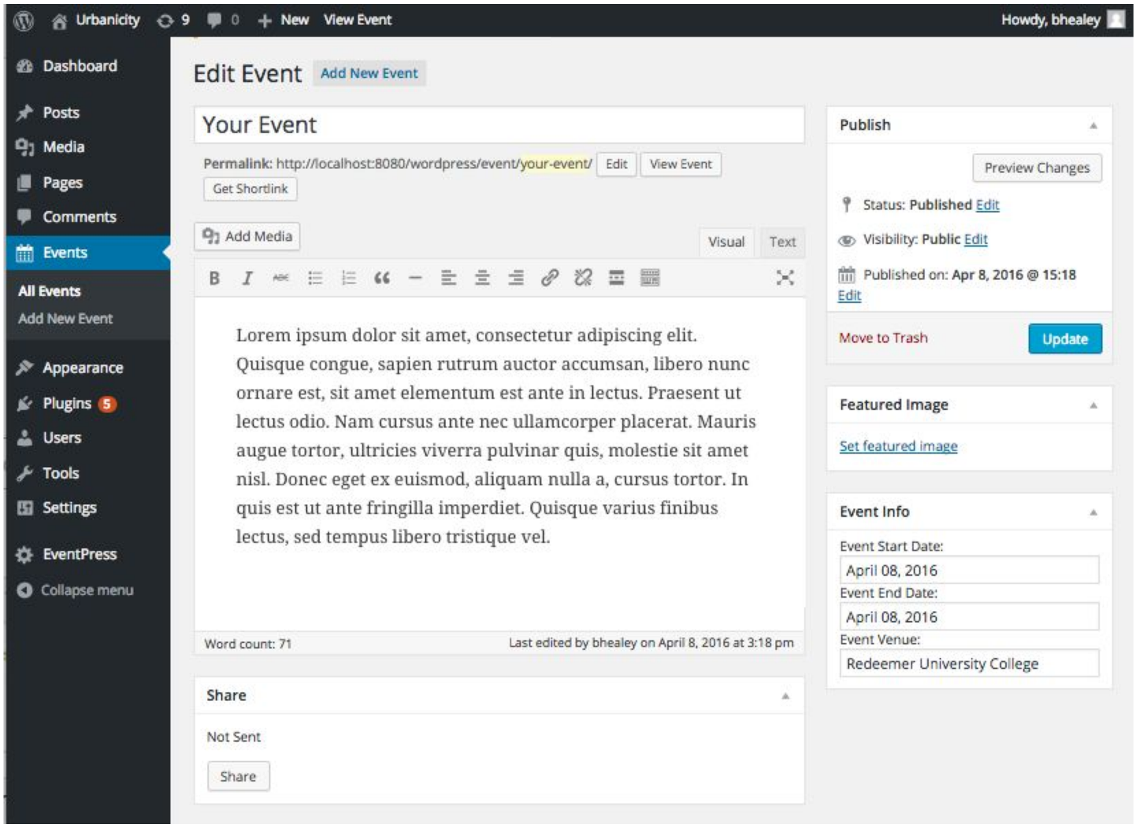
WordPress 4.4.2 is available! Please update now.

Events

All (4) | Published (4) | Trash (2)

4 items

□ Title	Date
□ Demo	2016/04/04 Published
□ End of Term	2016/03/07 Published
□ Ben Bock’s Birthday	2016/02/01 Published
□ Christmas Day	2016/01/29 Published



Your Event

Permalink: http://localhost:8080/wordpress/event/your-event/

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque congue, sapien rutrum auctor accumsan, libero nunc ornare est, sit amet elementum est ante in lectus. Praesent ut lectus odio. Nam cursus ante nec ullamcorper placerat. Mauris augue tortor, ultricies viverra pulvinar quis, molestie sit amet nisl. Donec eget ex euismod, aliquam nulla a, cursus tortor. In quis est ut ante fringilla imperdiet. Quisque varius finibus lectus, sed tempus libero tristique vel.

Word count: 71

Last edited by bhealey on April 8, 2016 at 3:18 pm

Status: Published Edit

Visibility: Public Edit

Published on: Apr 8, 2016 @ 15:18 Edit

Event Start Date: April 08, 2016

Event End Date: April 08, 2016

Event Venue: Redeemer University College

Not Sent

Conclusions

In this project so far I have not completed the functionality that I originally set out to have done. In hindsight it may have been easier to add my desired functionality to an existing plugin, however this is easier now that I understand the wordpress architecture.

Future work

The main desired functionality is the first priority, which is easily achievable. I understand how the core functionality is supposed to work, it just needs a bit more work to be functional. Then future work would be adding other features. I will probably finish the core functionality and then make it open source, and include a list of other ideas for things that I think would make it good. For example, allow multiple host users to register and auto-location feature for when creating an event.

Overview of Design Norms

Aesthetic norms interact with this project. One of the core things with my project was designing something that looks good. That’s why the wordpress architecture was an obvious choice because I am already building off of something that is proven to be aesthetically pleasing and functional. This norm is an interesting one because there exists lots of documentation for what makes something “good looking” but as humans by our very nature we are creative beings, it seemed odd that some creation is labeled as bad. I realized though that aesthetics is more than if something is good looking but its functionality speaks to its beauty. For me the iPhone is a great example of this, I find it’s beauty in how intuitive it is, almost as if it’s being created by my use of it.

Because I am creating a plugin that deals with event creation there is also the social norms that interact with my plugin. I needed to create something that was not only easy to use but I needed to capture the simplest essence of an event, what information we need, yes. But more importantly what is an event?

List of References

Tutsplus
http://code.tutsplus.com/tutorials/create-a-custom-wordpress-plugin-from-scratch--net-2668

Wordpress.org
https://codex.wordpress.org/Writing_a_Plugin

GitHub
https://github.com/Redeemer-CSC390/BenHealey

Smartsheet (Gantt Chart creator tool)
https://app.smartsheet.com/b/home

Williams, Brad, Ozh Richard, and Justin Tadlock. *Professional WordPress Plugin Development*. Indianapolis, IN: Wiley Pub., 2011. Print.

Reflection

I enjoyed this project very much. Learning about the wordpress architecture and plugin development has already proven to be valuable knowledge. I learned allot about conforming to another organization's standards of coding, with the wordpress codex.

I also learned how to adequately judge how much time is required for a project. I now afford way more time, especially if learning a system that I am not familiar with. I learned about how to manage my time when self starting, I do believe this will be easier then this course displayed, in actual software development there will be less deadlines of drastically different types (i.e. having assignments due in other classes that are completely different subjects).

I felt I learned allot about the frustration that will be dealing with managers, and tight deadlines. Especially the vexing problem of trying to explain quickly the progress that's been made that can't be shown.

I have learned also a great lesson in humility, as I go forward I understand how important it is to thoroughly educate myself on the changing field of computer technologies so that I can be viable in the industry, it is vain to think I know everything and wise for me to try to learn as much as possible.