



PREMIUM TRAINING

Gravity Forms -->> <http://webdesign.com/gravity>

** SPECIAL DISCOUNT **

This special discount is ONLY good for attendees of this class. Please do not pass this coupon over to other people. If this coupon code is found to be abused, it will be nullified and removed. (Do not put this code on your website, Facebook, or Twitter.)

Coupon Code: **WEBDESIGN50**

This will give you 50% off the Developer License of Gravity Forms. The coupon expires July 1st.

Gravity Forms -->> <http://webdesign.com/gravity>

DAY TWO

Modal Windows

[http://codecanyon.net/item/lightbox-evolution-for-wordpress/119478?
=&BenjaminBradley](http://codecanyon.net/item/lightbox-evolution-for-wordpress/119478?=&BenjaminBradley)

By using a lightbox plugin, we can begin to use Gravity Forms inside a modal window. One of the easiest ways is by linking to another page that we create using Builder and Gravity Forms.

Step 1 - Create a new layout, with an HTML module. Inside that module we will place the shortcode for the Gravity Forms form we want to use.

Step 2 - Assign that layout to a page

Step 3 - Link to that page using some lightbox "markup". Example:

```
<a href="http://wpzoo.com/modal-test?lightbox[width]=525&lightbox[height]=500" class="lightbox">Go Here</a>
```

Also check out this link for another way of doing this: <http://www.gravityhelp.com/creating-a-modal-form-with-gravity-forms-and-fancybox/>

Code for adding additional fields to User Profiles

```
function my_new_contactmethods($contactmethods) {  
    // Add Twitter  
    $contactmethods['twitter'] = 'Twitter';  
    // Add Facebook  
    $contactmethods['facebook'] = 'Facebook';  
    return $contactmethods;  
}  
add_filter('user_contactmethods','my_new_contactmethods',10,1);
```

Code for removing fields to User Profiles

```
function hide_profile_fields($contactmethods) {  
    unset($contactmethods['aim']);  
    unset($contactmethods['jabber']);  
    unset($contactmethods['yim']);  
    return $contactmethods;  
}  
add_filter('user_contactmethods','hide_profile_fields',10,1);
```

Pre-populate fields out of the User Profiles

The first bit of code will go in your active theme's functions.php file:

```
// populate the field with "user_firstname" as the population parameter with
the "first_name" of the current user
add_filter('gform_field_value_user_firstname', create_function("", '$value =
populate_usermeta(\'first_name\'); return $value;' ));

// populate the field with "user_lastname" as the population parameter with
the "last_name" of the current user
add_filter('gform_field_value_user_lastname', create_function("", '$value =
populate_usermeta(\'last_name\'); return $value;' ));

// this function is called by both filters and returns the requested user
meta of the current user
function populate_usermeta($meta_key){
    global $current_user;
    get_currentuserinfo();

    foreach($current_user as $key => $value){
        if($key == $meta_key)
            return $value;
    }

    return '';
}
```

What types of info can you call? Anything that exists in the user profile! (Even stuff you add at a later time.

**first_name, last_name, user_login, user_pass, user_email, user_url,
user_registered, user_status, display_name, nickname, description, aim,
yim, jabber**

All you have to do is replace any of these, or add these fields to the code above and you are good to go.

Don't store data on your server once you send it somewhere else

This code goes in your functions.php file of your active WordPress theme.

```
add_action('gform_post_submission_1', 'remove_form_entry', 10, 2);
function remove_form_entry($entry, $form){
    global $wpdb;

    $lead_id = $entry['id'];
    $lead_table = RGFormsModel::get_lead_table_name();
    $lead_notes_table = RGFormsModel::get_lead_notes_table_name();
    $lead_detail_table = RGFormsModel::get_lead_details_table_name();
    $lead_detail_long_table = RGFormsModel::get_lead_details_long_table_name
();

    //Delete from detail long
    $sql = $wpdb->prepare(" DELETE FROM $lead_detail_long_table
                            WHERE lead_detail_id IN(
                                SELECT id FROM $lead_detail_table WHERE
lead_id=%d
                            )", $lead_id);
    $wpdb->query($sql);

    //Delete from lead details
    $sql = $wpdb->prepare("DELETE FROM $lead_detail_table WHERE lead_id=%d",
$lead_id);
    $wpdb->query($sql);

    //Delete from lead notes
    $sql = $wpdb->prepare("DELETE FROM $lead_notes_table WHERE lead_id=%d",
$lead_id);
    $wpdb->query($sql);

    //Delete from lead
    $sql = $wpdb->prepare("DELETE FROM $lead_table WHERE id=%d", $lead_id);
    $wpdb->query($sql);
}
```

Allow for a unique transaction number to be auto-populated:

This code goes in the active theme's functions.php file:

```
add_filter("gform_field_value_uuid", "uuid");
function uuid($prefix = '')
{
    $chars = md5(uniqid(mt_rand(), true));
    $uuid = substr($chars,0,8) . '-';
    $uuid .= substr($chars,8,4) . '-';
    $uuid .= substr($chars,12,4) . '-';
    $uuid .= substr($chars,16,4) . '-';
    $uuid .= substr($chars,20,12);
    return $prefix . $uuid;
}
```

--

Design and Layout

<http://rkt.gs/cssguide>

CSS Ready Classes

http://www.gravityhelp.com/documentation/page/CSS_Ready_Classes

IMPORTANT

gform_field_value_***
gform_hidden

Awesome Support

<http://www.gravityhelp.com/forums/>

Showing fields based on the logged in/out status of a user

This is a multistep process.

Step 1 - Create a multiple choice field. You can call it “Is user logged in?” with the options of Yes and No.

Step 2 - In the Advanced tab we need to hide this form by using the class ***gform_hidden***. Plus we need to populate this field dynamically. So lets add a parameter named ***loginmask***.

Step 3 - Now we will create a special field that has the conditional logic that looks at the above login variable field.

Step 4 - Add some code to the active child theme’s functions.php file

```
add_filter('gform_field_value_loginmask', 'gf_is_user_logged_in');
function gf_is_user_logged_in($value){
    return is_user_logged_in() ? 1 : 0;
}
```

Step 5 - Now you can choose to show fields based on logged in users. Possibilities are amazing here.