

WORDPRESS DOCUMENTATION

1. INTRODUCTION

WordPress is a way to create websites in an open-source, modular ecosystem. It's known as a Content Management System (CMS), which means you get an intuitive way to create and manage your posts and pages of your website.

There are two ways to do this:

- **Themes** – These dictate the look of your site, although some also provide extra functionality.
- **Plugins** – This is how you extend the functionality of your site, and the plugin ecosystem is huge. You'll be able to find a plugin for any practical application and might need.

Since WordPress is an open-source platform, you're also able to develop your site using any coding skills you have.

2. WordPress Tasks

1.Add Post

This task is to add a blog using custom HTML form. This task was done by three steps,

- Create custom template for blog page
- Create HTML form for accepting inputs
- Submit the data by ajax request and store it into database

Source code: Create template and add the form

```
<?php
/**
 * Template name: blogs
 */
get_header();
?>
<main id="primary" class="site-main <?php echo esc_attr( apply_filters( 'botiga_content_class',
" ) ); ?>">
<meta name="viewport" content="width=device-width, initial-scale=1">
<form type="post" action="<?php echo site_url() ?>/wp-admin/admin-ajax.php" id="addPost">
<h3 class="ml-2 mt-2"><i>Add Blogs</i></h3>
<p class="ml-2 mt-2"><label> Post Title <br><span><input type="text" name="title" value=""
size="80" required="required" class="form-control w-50 p-2 ml-2"></span></label></p>
<?php
if( $terms = get_terms( array( 'taxonomy' => 'category', 'orderby' => 'name' ) ) ) :
echo '<span><select name="categoryfilter" class="form-control w-25 p-2 ml-3" ><option
value="">Select category...</option>';
```

```

foreach ( $terms as $term ) :

echo '<option value="" . $term->term_id . ">' . $term->name . '</option>'; // ID of the category
as the value of an option

endforeach;

echo '</select></span>';

endif;

?><p><span>

<p class="ml-2 mt-2"><label>Post Content<br><span>

<textarea name="body" cols="10" rows="5" required="required" class="form-control w-100 p-
3 ml-2"></textarea>

</span></label>

<input type="hidden" name="action" value="save_ajax_data"/>

<br><button type="submit" class="form-control w-25 p-2 ml-2 btn btn-primary" name="submit-
post">Add</button></p>

</form>

<br>

<br>

</main>

<?php
get_footer();t

```

Custom.js

```

jQuery(function($){
    $('#addPost').submit(function(){
        // console.log("testttt");
        var addPost = $('#addPost');
        $.ajax({
            url:addPost.attr('action'),
            data:addPost.serialize(), // form data
            type:addPost.attr('method'), // POST

```

```

beforeSend:function(xhr){

addPost.find('button').text('posting...'); // changing the button label
},
success:function(data){
addPost.find('button').text('post'); // changing the button label back
$('#response').html(data);
// insert data
}
});
return false;

});

});

```

function.php

```

add_action('wp_ajax_save_ajax_data', 'saveAjaxData'); // wp_ajax_{ ACTION HERE }

add_action('wp_ajax_nopriv_save_ajax_data', 'saveAjaxData');


function saveAjaxData(){

$title = $_REQUEST['title'];

$body = $_REQUEST['body'];

$new_post = array(

    'post_title' => $title,

    'post_content' => $body,

    'post_status' => 'publish',

    'post_date' => date('Y-m-d H:i:s'),

```

```

        'post_author' => $user_ID,
        'post_type' => 'post',

        'post_category' => array($_REQUEST['categoryfilter'])

    );

?>
<?php

$post_id = wp_insert_post($new_post);
if(isset($post_id) )
{
    // echo "Post added";

    // the query

    $wpb_all_query = new WP_Query(array('post_type'=>'post', 'post_status'=>'publish',
'posts_per_page'=>-1)); ?>

<?php if ( $wpb_all_query->have_posts() ) : ?>

<!-- <ul> -->

<!-- the loop -->

<?php while ( $wpb_all_query->have_posts() ) : $wpb_all_query->the_post(); ?>

<a href="<?php the_permalink(); ?>"><h4><?php the_title(); ?></h4></a>

<?php
the_content(); ?>

```

```
<?php wp_link_pages( array( 'before' => '<div class="page-links">' . __( 'Pages:', 'minamaze'
), 'after' => '</div>', ) ); ?>
```

```
<?php endwhile; ?>
```

```
<!-- end of the loop -->
```

```
<!-- </ul> -->
```

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

```
<?php else : ?>
```

```
<p><?php _e( 'Sorry, no posts matched your criteria.' ); ?></p>
```

```
<?php endif;
```

```
}
```

```
else{
```

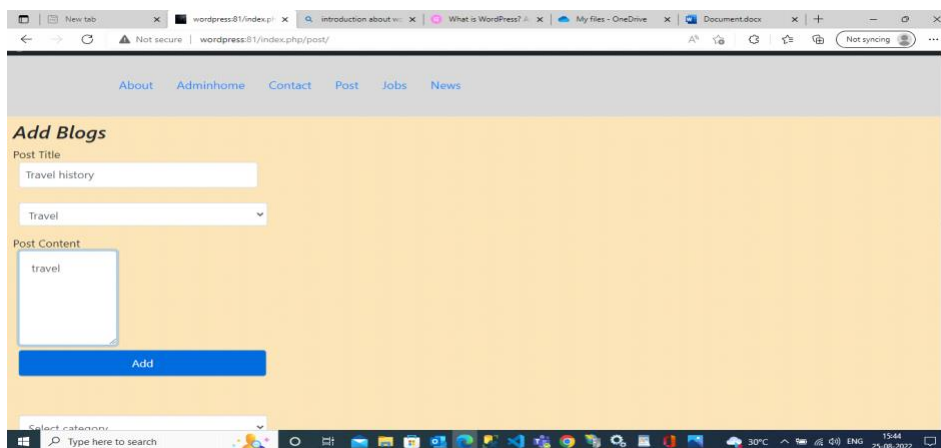
```
echo "Post is not added";
```

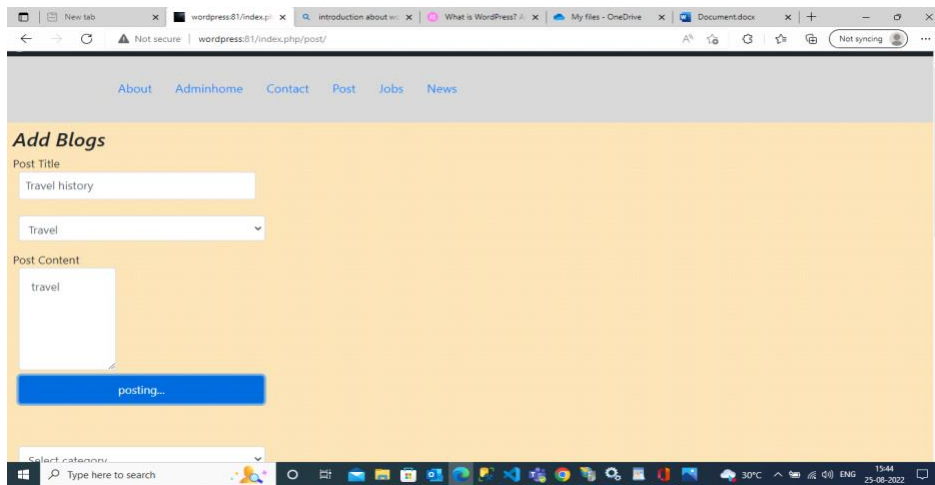
```
}
```

```
wp_die();
```

```
}
```

Screenshots





2.Post filtering by category

This task is to filter blogs by their category and display the result. This was done by adding a dropdown with the categories and filtering it by button click.

Source Code: Adding dropdown to blog template

```
<form action="<?php echo site_url() ?>/wp-admin/admin-ajax.php" method="POST"
id="filter">
```

```
<?php
```

```
if( $terms = get_terms( array( 'taxonomy' => 'category', 'orderby' => 'name' ) ) ) :
```

```
    echo '<select name="categoryfilter" class="form-control w-25 p-2 ml-3" ><option
value="">Select category...</option>';
```

```
    foreach ( $terms as $term ) :
```

```
        echo '<option value="" . $term->term_id . ">" . $term->name . '</option>'; // ID of the
category as the value of an option
```

```
    endforeach;
```

```
    echo '</select>';
```

```

        endif;

    ?>

<button class="form-control w-25 p-2 ml-3 mt-2 btn btn-info">Apply filter</button>

    <input type="hidden" name="action" value="myfilter">

</form>

<h2>Post</h2>

<div id="response">

<?php

    the_content(); ?>

<?php wp_link_pages( array( 'before' => '<div class="page-links">' . __( 'Pages:', 'minamaze' ),
'after' => '</div>', ) ); ?>

</article></div>

</main>

<?php

get_footer();

```

function.php

```

add_action('wp_ajax_myfilter', 'misha_filter_function'); // wp_ajax_{ACTION HERE}
add_action('wp_ajax_nopriv_myfilter', 'misha_filter_function');

function misha_filter_function(){

    $args = array(

        'orderby' => 'date', // we will sort posts by date

        'order' => $_POST['date'] // ASC or DESC

    );

    // for taxonomies / categories

    if( isset( $_POST['categoryfilter'] ) )

        $args['tax_query'] = array(

            array(

```



```

        'taxonomy' => 'category',
        'field' => 'id',
        'terms' => $_POST['categoryfilter']
    )
);

$query = new WP_Query( $args );

if( $query->have_posts() ) :

    while( $query->have_posts() ): $query->the_post();
// echo '<h2>' . $query->post->post_title . '</h2>';

?>

<article id="post-id-<?php the_id(); ?>" <?php post_class('clearfiks archive-post'); ?>>

<a href="<?php the_permalink() ?>"><h4><?php the_title(); ?></h4>

<?php the_content(); ?>

<?php wp_link_pages( array( 'before' => '<div class="page-links">' . __( 'Pages:', 'minamaze' ),
'after' => '</div>', ) ); ?></article>

<?php endwhile;

wp_reset_postdata();

else : echo 'No posts found';

endif;

die();

}

```

Custom.js

```

jQuery(function($){
    $('#filter').change(function(){
        var filter = $('#filter');
        $.ajax({
            url:filter.attr('action'),

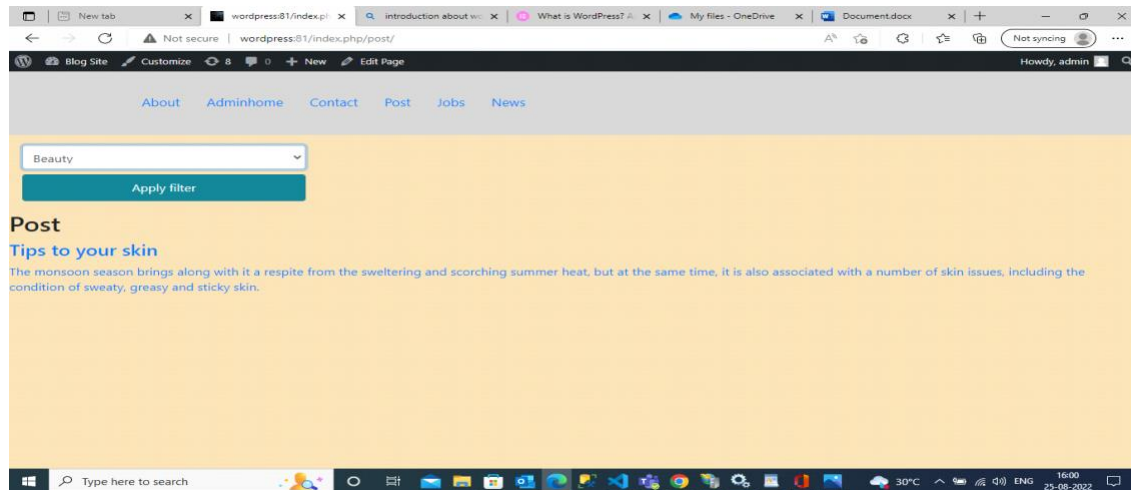
```

```

data:filter.serialize(), // form data
type:filter.attr('method'), // POST
beforeSend:function(xhr){
    filter.find('button').text('Processing...'); // changing the button label
},
success:function(data){
    filter.find('button').text('Apply filter'); // changing the button label back
$('#response').html(data); // insert data
}
});
return false;
});
});

```

Screenshots



3. Autocomplete search Post

This task is to search post. And this will result in matching posts by two letter search. This was done by create the search form and send the matching result by ajax call.

Source code

```
<?php
/**
 * Template name: blogs
 */
get_header();

?><label><strong> <h4 class="ml-2 mt-2">Search Post</h4><span><input type="text"
placeholder="search...." name="keyword" id="keyword" onkeyup="fetch()" class="form-
control w-25 p-2 ml-2"></input></strong>

<div id="datafetch"></div>

<br>

<?php
get_footer();
```

Custom.js

```
function fetch() {
    jQuery.ajax({
        url: my_ajax_object.ajax_url,
        type: 'post',
        data: { action: 'data_fetch', keyword: jQuery('#keyword').val() },
        success: function(data) {
            jQuery('#datafetch').html(data);
        }
    });
}
```

function.php

```
function wpdocs_theme_name_scripts() {  
    wp_enqueue_script( 'ajax-script', get_template_directory_uri() . '/js/custom.js',  
        array('jquery'), );  
    wp_localize_script( 'ajax-script', 'my_ajax_object', array( 'ajax_url' => admin_url(  
'admin-ajax.php' ) ) );  
}
```

```
add_action( 'wp_enqueue_scripts', 'wpdocs_theme_name_scripts' );
```

```
function data_fetch(){  
    $the_query = new WP_Query( array( 'posts_per_page' => -1, 's' => esc_attr(  
        $_POST['keyword'] ), 'post_type' => 'post' ) );  
    if(($_POST['keyword'])!="")  
  
    {  
  
        if(preg_match('/[.\'^£$%&*()]{ @#~?><>,|=_+~\-'}/',$_POST['keyword']))  
        {  
            echo '<strong>Special characters not allowed</strong>';  
        }  
        else  
        {  
            if( $the_query->have_posts() ) :  
                while( $the_query->have_posts() ): $the_query->the_post(); ?>  
                    <strong><a href="<?php echo esc_url( post_permalink() ); ?>"><?php  
the_title();?></a><br>  
                    </strong>
```

```
<?php endwhile;

wp_reset_postdata();

else:

echo "<strong>No Post found</strong>";

endif;

}

}

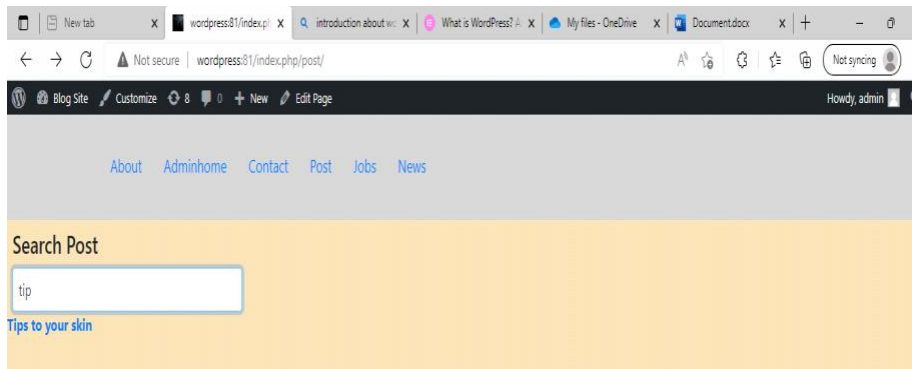
die();

}

add_action('wp_ajax_data_fetch', 'data_fetch');

add_action('wp_ajax_nopriv_data_fetch','data_fetch');
```

Screenshots



4.WP Mail SMTP

WordPress' email feature uses the Hypertext Preprocessor (PHP) mail() function by default. However, it is not the most effective tool as it may trigger spam filters and send error messages to its users.

The Simple Mail Transfer Protocol (SMTP) server is better for WordPress website owners who frequently exchange emails with their visitors. It offers high security and deliverability to ensure properly sent emails.

WP Mail SMTP is the best WordPress SMTP plugin for emails. It configures the `wp_mail()` function to use the correct SMTP host credentials. It also lets you leverage a built-in mail provider to prevent emails from going to the spam folder.

The free version of WP Mail SMTP supports the following email marketing services and SMTP providers:

- SendLayer
- SMTP.com
- Sendinblue
- Mailgun
- SendGrid
- Postmark
- SparkPost
- Gmail

WP Mail SMTP — Configure on the settings page

1. Navigate to the WP MAIL SMTP > Settings page.

From Email

*The email address that emails are sent from.
If you're using an email provider (Yahoo, Outlook.com, etc) this should be your email address for that account.
Please note that other plugins can change this, to prevent this use the setting below.*

☒ **Force From Email**
If checked, the From Email setting above will be used for all emails, ignoring values set by other plugins.

From Name

The name that emails are sent from.











☒ **Force From Name**
If checked, the From Name setting above will be used for all emails, ignoring values set by other plugins.

Return Path ☒ **Set the return-path to match the From Email**
*Return Path indicates where non-delivery receipts - or bounce messages - are to be sent.
If unchecked, bounce messages may be lost.*

2. Fill in the following fields:

- **From Email** - Enter the email you wish to send FROM
- **Force From Email** - Check this box
- **From Name** - Enter a name you'd like to appear as being sent FROM
- **Force From Name** - Check this if you wish to force the FROM name
- **Return Path** - Check this box

Mailer

	 <small>RECOMMENDED</small>	 <small>RECOMMENDED</small>		
<input type="radio"/> Default (none)	<input type="radio"/> SMTP.com	<input type="radio"/> Sendinblue	<input type="radio"/> Mailgun	<input type="radio"/> SendGrid
				
<input type="radio"/> Amazon SES	<input type="radio"/> Gmail	<input type="radio"/> Outlook	<input type="radio"/> Zoho Mail	<input checked="" type="radio"/> Other SMTP

3. Under the Mailer section, choose Other SMTP.

Other SMTP

Use the SMTP details provided by your hosting provider or email service.

To see recommended settings for the popular services, as well as troubleshooting tips, check out our [SMTP documentation](#).

SMTP Host

smtp.dreamhost.com

Encryption

☐ None ☐ SSL ☒ TLS

For most servers TLS is the recommended option. If your SMTP provider offers both SSL and TLS options, we recommend using TLS.

SMTP Port

587

4. Under the Other SMTP section. Make sure to toggle the Authentication button.

- **SMTP Host** - smtp.dreamhost.com
- **Encryption** - Use TLS encryption
- **SMTP Port** - 587

In this example, port 587 with TLS is used. If you would rather use port 465, make sure to use the SSL option instead.

SMTP Username

admin@example.com

SMTP Password

.....

- **Username** - Enter your DreamHost email address
- **Password** - Enter your email address password

An orange rectangular button with rounded corners and a white border, containing the text "Save Settings" in white, bold, sans-serif font. The button is centered within a light gray rectangular background.

- Click the Save Settings button.

5.Custom Registration Form

- a. In this task create the user registration form and send a email notification to user when user profile successfully created.For this created a registration form template and store the data into database.

Source code for registration form template

```
<?php
/*
Template Name: regs
*/
the_post();
get_header();
?>
<div class="wrapper">
<?php
$error= "";
$success = "";

global $wpdb, $PasswordHash, $current_user, $user_ID;
//Only report fatal errors and parse errors.
error_reporting(E_ERROR | E_PARSE);
if(isset($_POST['task']) && $_POST['task'] == 'register') {
    $password1 = $wpdb->escape(trim($_POST['password1']));
    $password2 = $wpdb->escape(trim($_POST['password2']));
```

```

$first_name = $wpdb->escape(trim($_POST['first_name']));
$last_name = $wpdb->escape(trim($_POST['last_name']));
$email = $wpdb->escape(trim($_POST['email']));
$username = $wpdb->escape(trim($_POST['username']));
if( $email == "" || $password1 == "" || $password2 == "" || $username == "" || $first_name == ""
|| $last_name == "" ) {
$error= 'Please don\'t leave the required fields.';
} else if(!filter_var($email, FILTER_VALIDATE_EMAIL)) {
$error= 'Invalid email address.';
} else if(email_exists($email) ) {
$error= 'Email already exist.';
} else if($password1 <> $password2 ){
$error= 'Password do not match.';
}
else
$user_id = wp_insert_user( array ('first_name' => apply_filters('pre_user_first_name',
$first_name), 'last_name' => apply_filters('pre_user_last_name', $last_name), 'user_pass' =>
apply_filters('pre_user_user_pass', $password1), 'user_login' =>
apply_filters('pre_user_user_login', $username), 'user_email' =>
apply_filters('pre_user_user_email', $email), 'role' => 'subscriber' ) );
if( is_wp_error($user_id) ) {
$error= 'Error on user creation.';
} else {
do_action('user_register', $user_id);
$success = 'You are successfully register';
}
}
}
?>
<!--display error/success message-->
<div id="message">

```

```

<?php
if(! empty($err) ) :
echo '<p class="error">'. $err.';
endif;
?>

<?php
if($success):
echo '<p class="error">'. $success.';

    $to = 'archananair0808@gmail.com';
    $registeredEmail = $_GET['email'];
    $headers = 'From: ' . $registeredEmail . "\r\n" .
    'Reply-To: ' . $registeredEmail . "\r\n";
    // send mail
    $subject = $email;
    $message = $username."successfully registered";
    $sent = wp_mail($to, $subject, strip_tags($message), $headers);
    endif;
?>

</div>

<form method="post">
<h3>Create Account</h3>
<p><label> Enter First Name<br><span>
<p><input type="text" value="" name="first_name" id="first_name" size="40" class="form-
control w-25 p-2 ml-2" /></p>
<p><label>Enter Last Name</label></p>
<p><input type="text" value="" name="last_name" id="last_name" size="40" class="form-
control w-25 p-2 ml-2"/></p>
<p><label>Enter Email</label></p>
<p><input type="text" value="" name="email" id="email" size="40" class="form-control w-25
p-2 ml-2"/></p>
<p><label>Enter Username</label></p>

```

```

<p><input type="text" value="" name="username" id="username" size="40" class="form-control w-25 p-2 ml-2" /></p>

<p><label>Enter Password</label></p>

<p><input type="password" value="" name="password1" id="password1" size="40" class="form-control w-25 p-2 ml-2"/></p>

<p><label>Retype Password</label></p>

<p><input type="password" value="" name="password2" id="password2" size="40" class="form-control w-25 p-2 ml-2"/></p>

<div class="alignleft"><p><?php if($success != "") { echo $sucess; } ?> <?php if($error!= "") { echo $error; } ?></p></div>

<button type="submit" name="btnregister" class="btn btn-primary w-25 mt-2 ml-2">Submit</button>

<input type="hidden" name="task" value="register" />

</form>

</div>

<?php get_footer() ?>

```

b.Add Image captcha in registration form

For security purpose add a image captcha in the registration form.

Source code for image captcha

captcha.php

```

<?php
    session_start();

    // Generate captcha code
    $random_num = md5(random_bytes(64));
    $captcha_code = substr($random_num, 0, 6);

    // Assign captcha in session
    $_SESSION['CAPTCHA_CODE'] = $captcha_code;

    // Create captcha image
    $layer = imagecreatetruecolor(168, 37);
    $captcha_bg = imagecolorallocate($layer, 247, 174, 71);

```

```
imagefill($layer, 0, 0, $captcha_bg);
$captcha_text_color = imagecolorallocate($layer, 0, 0, 0);
imagestring($layer, 5, 55, 10, $captcha_code, $captcha_text_color);
header("Content-type: image/jpeg");
imagejpeg($layer);?>
```

Template part

```
<strong>
    Type the text in the image to prove
    you are not a robot
</strong>

<div style='margin:15px'>

</div>
    <input type="text" name="input" class="form-control w-25 p-2 ml-2" />
    <input type="hidden" name="flag" value="1"/>
    <!-- <input type="submit" value="Submit" name="submit"/> -->
<div style='margin-bottom:5px'>
    <?php echo $msg; ?>
</div>

<div>
    Can't read the image? Click
    <a href='<?php echo $_SERVER['PHP_SELF']; ?>'>
        here
    </a>
    to refresh!
</div>
```

```
<button type="submit" name="btnregister" class="btn btn-primary mt-3 w-25">Submit</button>
```

validating Captcha

```
if (isset($_POST['input']) && sizeof($_POST['input']) > 0)
{
    // If the captcha is valid

    if ($_POST['input'] == $_SESSION['CAPTCHA_CODE'])
    {
        $msg = '<span style="color:green">SUCCESSFUL!!!</span>';

        $user_id = wp_insert_user( array ('first_name' => apply_filters('pre_user_first_name',
        $first_name), 'user_pass' => apply_filters('pre_user_user_pass', $password1), 'user_login' =>
        apply_filters('pre_user_user_login', $username), 'user_email' =>
        apply_filters('pre_user_user_email', $email), 'role' => 'subscriber' ) );

        if( is_wp_error($user_id) ) {

            $error= 'Error on user creation.';

        } else {

            do_action('user_register', $user_id);

            $success = 'You are successfully registered';

        }
    }
}
```

6.Custom Contact us Form

This task to create custom contact us form. For this created a template and add code to send notification when user submit the form.

Source code for contact page

```
<?php
/*Template Name: mailer*/
```

```

get_header();

?>

<form action="" enctype="multipart/form-data" method="GET">

<h3><i>Contact Us</i></h3>

<p><label> Your name<br><span><input type="text" name="your-name" value="" size="40"
required="required"></span></label></p>

<p><label> Your email<br><span><input type="email" name="your-email" value="" size="40"
required="required"></span></label></p>

<p><label> Subject<br><span><input type="text" name="subject" value=""
size="40"></span></label></p>

<p><label> Your message<br><textarea name="your-message" cols="40" rows="10" aria-
invalid="false"></textarea></label></p>

<p><button type="submit">Submit</button></p>

<?php

    $to = 'archananair0808@gmail.com';

    $registeredEmail = $_GET['your-email'];

    $headers = 'From: ' . $registeredEmail . "\r\n" .
        'Reply-To: ' . $registeredEmail . "\r\n";

    // send mail

    $subject = $_GET['subject'];

    $message = $_GET['your-message'];

    $sent = wp_mail($to, $subject, strip_tags($message), $headers);

    if ($sent) {
        echo "Notification sended";
    } //message sent!

    else {
        echo "Mail sending error";
    } //mes }?>

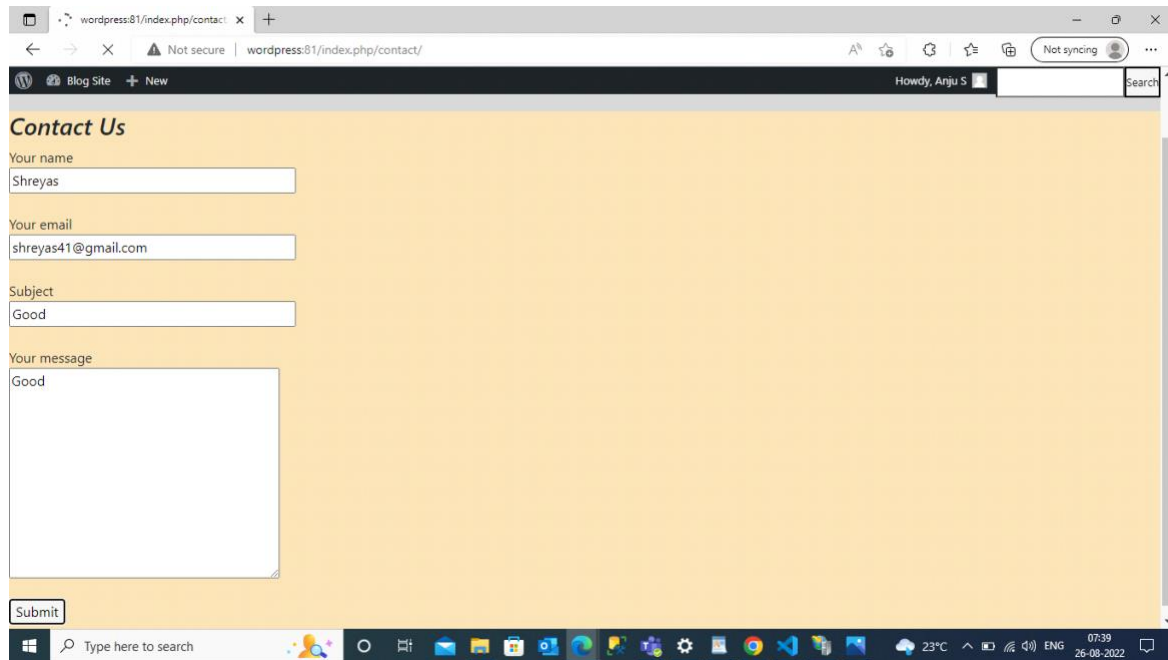
<?php

get_footer();

```

?>

Screenshort



7.Custom post type creation

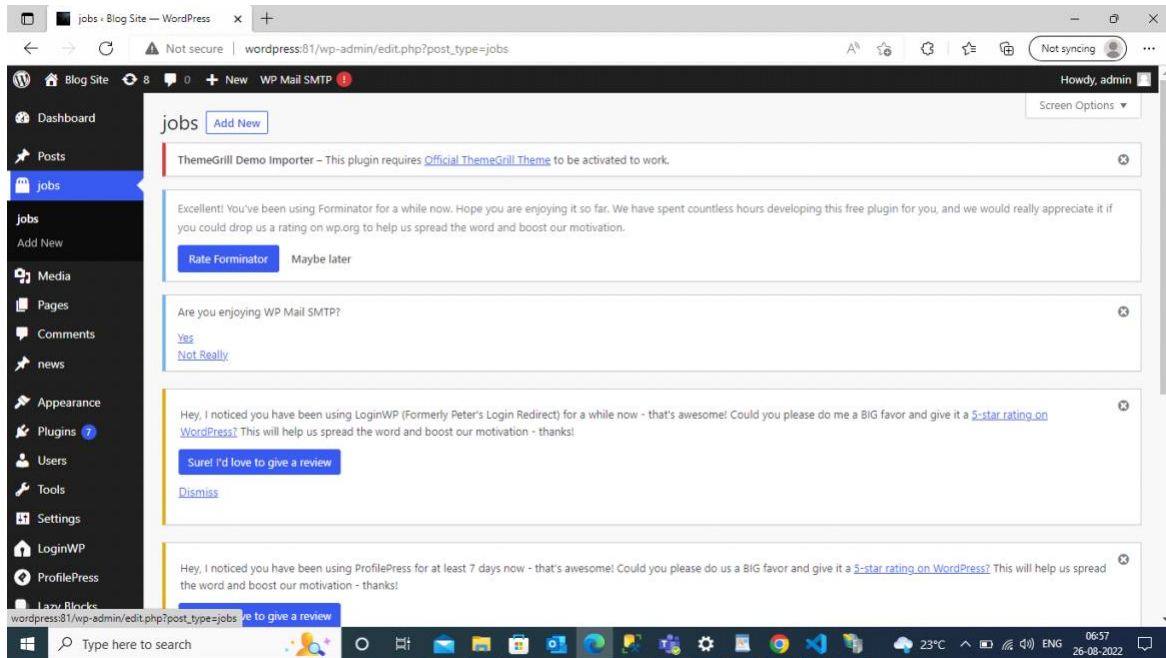
The term of “Post Type” used to determine different types of content that can hold and display in WordPress. By default WordPress already has several basic post types, they are:

- a. post
- b. page
- c. attachment
- d. revision
- e. nav_menu_item
- f. custom_css
- g. customize_changeset

But sometimes there is a need to create your own version of the post. This can be a portfolio, people, goods, etc.

For this register the custom post type by, Function `register_post_type()`.

```
/* Custom Post Type Start */  
function customs_posttype() {  
    register_post_type( 'jobs',  
        // CPT Options  
        array(  
            'labels' => array(  
                'name' => __( 'jobs' ),  
                'singular_name' => __( 'Jobs' )  
            ),  
            'public' => true,  
            'has_archive' => false,  
            'rewrite' => array('slug' => 'jobs'),  
            'menu_icon' => 'dashicons-products',  
            'menu_position' => 6  
        )  
    );  
}  
  
// Hooking up our function to theme setup  
add_action( 'init', 'customs_posttype' );  
  
add_action( 'init', function() {  
    remove_post_type_support( 'jobs', 'editor' );  
}, 99);  
/* Custom Post Type End */
```



8.Adding metafield to custom post type

Adding metafield to custom post type is we can specify the custom fields for post type.And that will be stored.Here created a custompost type news and added a custom field.

Source code

```
/* Custom Post Type Start */
```

```
function create_posttype() {
```

```
    register_post_type( 'news',
```

```
    // CPT Options
```

```
    array(
```

```
        'labels' => array(
```

```
            'name' => __( 'news' ),
```

```
            'singular_name' => __( 'News' )
```

```
        ),
```

```
'public' => true,

'has_archive' => false,

'rewrite' => array('slug' => 'news'),

)

);

}

// Hooking up our function to theme setup

add_action( 'init', 'create_posttype' );

/* Custom Post Type End */


/*Custom Post type start*/

function cw_post_type_news() {

$supports = array(

'title', // post title

'editor', // post content

'author', // post author

'thumbnail', // featured images

'excerpt', // post excerpt

'custom-fields', // custom fields

'comments', // post comments

'revisions', // post revisions

'post-formats', // post formats

);
```

```
$labels = array(

'name' => _x('news', 'plural'),

'singular_name' => _x('news', 'singular'),

'menu_name' => _x('news', 'admin menu'),

'name_admin_bar' => _x('news', 'admin bar'),

'add_new' => _x('Add New', 'add new'),

'add_new_item' => __('Add New news'),

'new_item' => __('New news'),

'edit_item' => __('Edit news'),

'view_item' => __('View news'),

'all_items' => __('All news'),

'search_items' => __('Search news'),

'not_found' => __('No news found.'),

);

$args = array(

'supports' => $supports,

'labels' => $labels,

'public' => true,

'query_var' => true,

'rewrite' => array('slug' => 'news'),

'has_archive' => true,

'hierarchical' => false,

);
```

```

register_post_type('news', $args);

}

add_action('init', 'cw_post_type_news');

/*Custom Post type end*/


add_action('add_meta_boxes',function (){

    add_meta_box(

        'title',

        'New Options',

        'news_custom_fields_html',

        ['news'],

        'normal' ,

        'high'

    );

});

function news_custom_fields_html($sport){

    $cs_meta_val=get_post_meta($sport->ID);

    ?>

<table class="form-table">

    <tr>

        <th><label for="title">Title</label></th>

```

```
<td><input type="text" name="title" id="title" class="regular-text" value="<?php if(
isset( $cs_meta_val['title'])) echo $cs_meta_val['title'][0] ?>"></td>
```

```
</tr>
```

```
</table>
```

```
<?php
```

```
}
```

```
//save meta value with save post hook
```

```
add_action('save_post', 'news_type_save_custom_field_value');
```

```
function news_type_save_custom_field_value( $post_id ){
```

```
    global $post;
```

```
    if(isset($_POST['title'])){
```

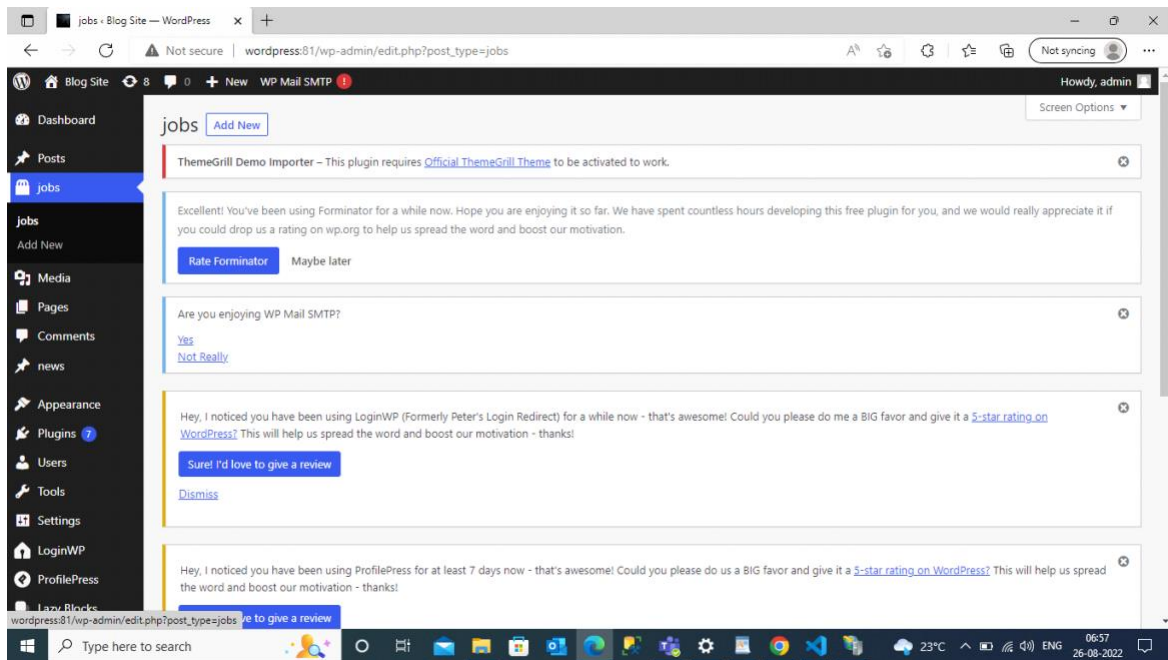
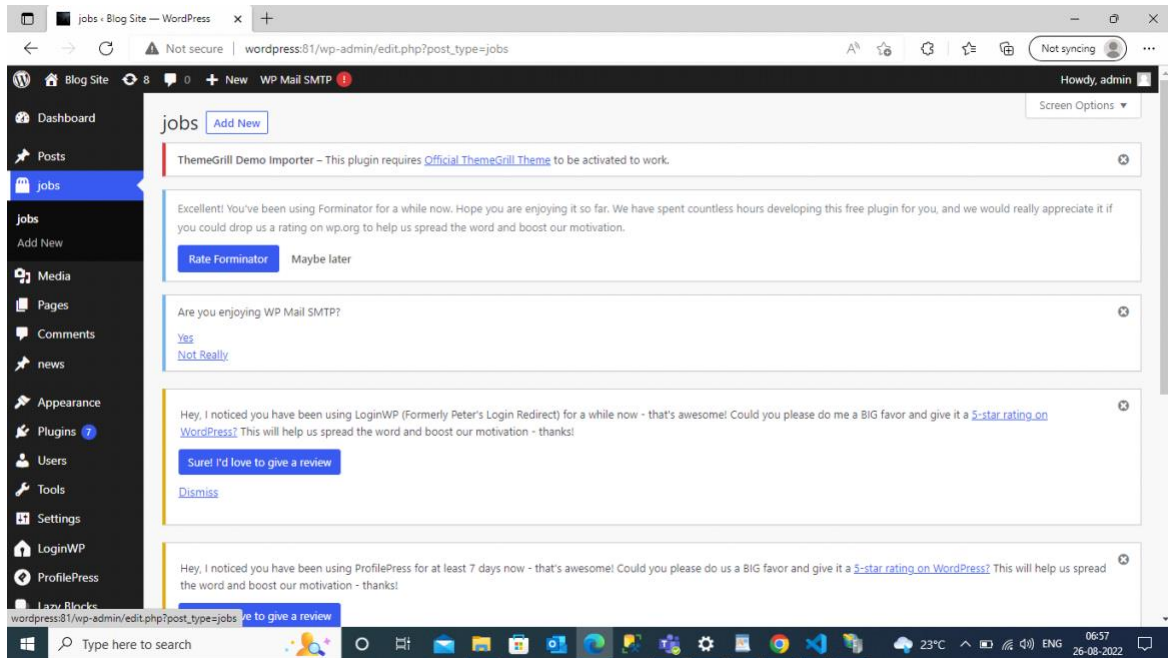
```
        update_post_meta($post_id,'title', sanitize_text_field($_POST['title']));
```

```
    } else {
```

```
        delete_post_meta( $post_id, 'title' );
```

```
    }
```

```
}
```



9.Different menus for different users

In this task display different menus for each users by their role.

Source code

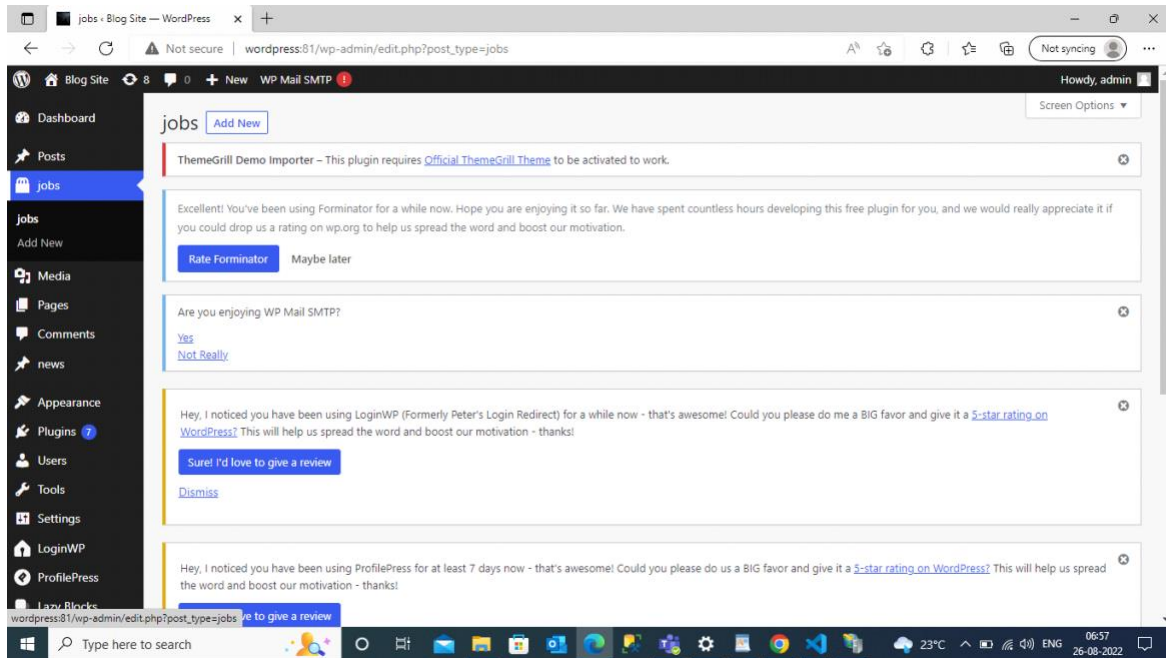
```
function my_wp_nav_menu_args($args = ")
```

```
{
if(is_user_logged_in())
{
    if(current_user_can('administrator'))
    {
        $args['menu'] = 'ad menu';
    }
elseif(current_user_can('subscriber'))
{
    $args['menu'] = 'log-in';
}
elseif(current_user_can('special_user'))
{
    $args['menu'] = 'log-in';
}
elseif(current_user_can('editor'))
{
    $args['menu'] = 'log-in';
}
elseif(current_user_can('reader'))
{
    $args['menu'] = 'log-in';
}
}
else
{
    $args['menu'] = 'logged-out';
}
return $args;
}
```

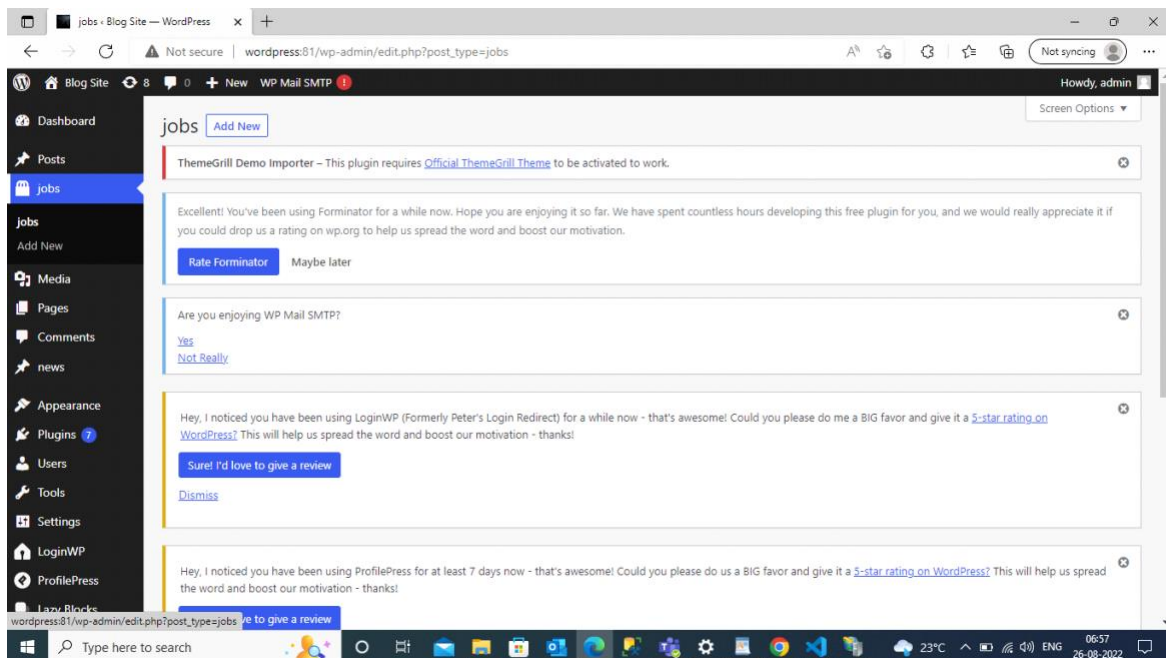


```
add_filter('wp_nav_menu_args', 'my_wp_nav_menu_args');
```

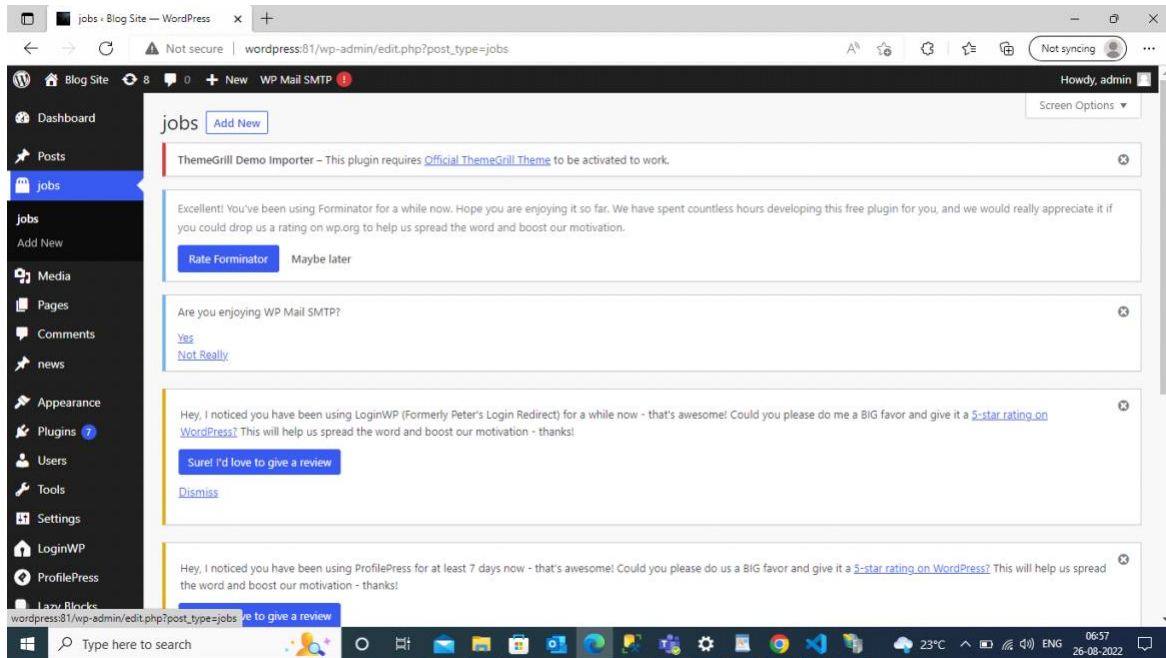
Admin menu



user menu



Index menu



9.Create Special user and give some privileges

Created Special user and give them privileges including reading private post and managing users(add,view and delete users).

source code

```
add_role( 'special_user', __( 'Special User' ), array( 'list_users' => true , 'edit_users' => true , 'read_private_posts' => true , 'create_users' => true , 'delete_users' => true ) );
```

Screenshots

jobs • Blog Site — WordPress

Not secure | wordpress:81/wp-admin/edit.php?post_type=jobs

Howdy, admin

Dashboard Posts jobs **jobs** Media Pages Comments news Appearance Plugins 7 Users Tools Settings LoginWP ProfilePress Lazy Blocks

jobs [Add New](#) [Screen Options](#)

ThemeGrill Demo Importer – This plugin requires [Official ThemeGrill Theme](#) to be activated to work.

Excellent! You've been using Forminator for a while now. Hope you are enjoying it so far. We have spent countless hours developing this free plugin for you, and we would really appreciate it if you could drop us a rating on wp.org to help us spread the word and boost our motivation.

[Rate Forminator](#) [Maybe later](#)

Are you enjoying WP Mail SMTP?

[Yes](#) [Not Really](#)

Hey, I noticed you have been using LoginWP (Formerly Peter's Login Redirect) for a while now - that's awesome! Could you please do me a BIG favor and give it a [5-star rating on WordPress?](#) This will help us spread the word and boost our motivation - thanks!

[Sure! I'd love to give a review](#) [Dismiss](#)

Hey, I noticed you have been using ProfilePress for at least 7 days now - that's awesome! Could you please do us a BIG favor and give it a [5-star rating on WordPress?](#) This will help us spread the word and boost our motivation - thanks!

wordpress:81/wp-admin/edit.php?post_type=jobs [Click here to give a review](#)

Type here to search

23°C 06:57 26-08-2022

wordpress:81/index.php/category/uncategorized/

Blog Site Customize 8 + New Edit Category WP Mail SMTP

Howdy, admin

[About](#) [Adminhome](#) [Contact](#) [Post](#) [Jobs](#) [News](#)

Uncategorized

tests

[Read More](#)

newss

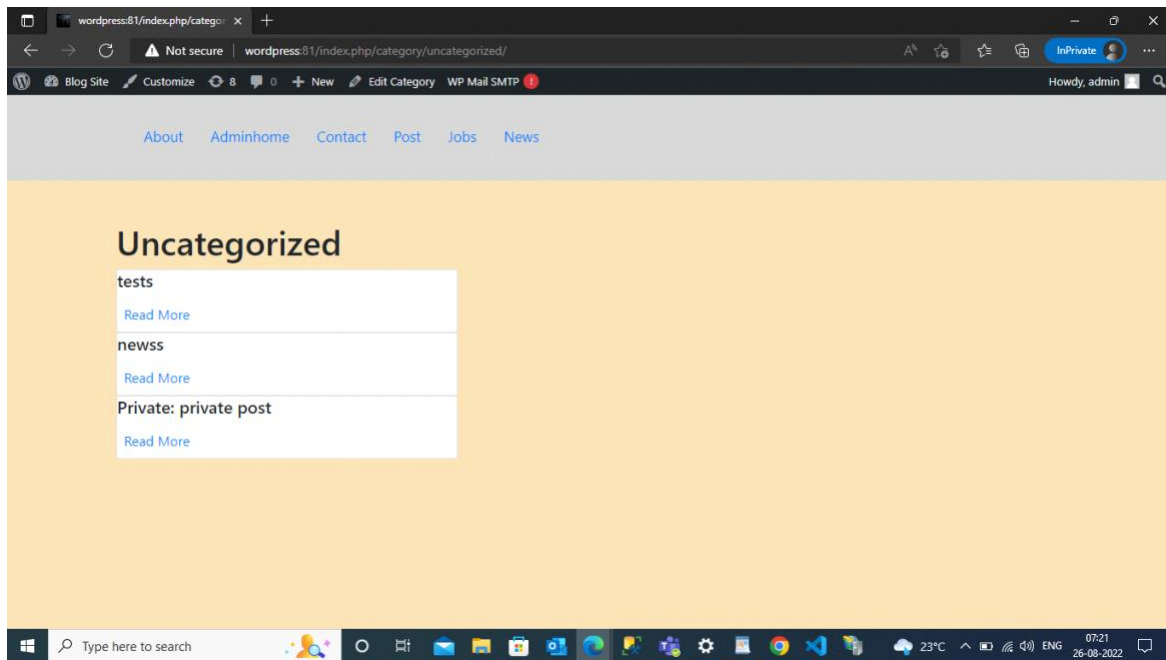
[Read More](#)

Private: private post

[Read More](#)

Type here to search

23°C 07:21 26-08-2022



10.Custom Widget

Add custom widget to page and show the post categories in the widgets. For this added the custom widget code and display the post categories in the widgetized area.

Source code

```
if ( function_exists('register_sidebar') )
register_sidebar(array(
'name' => 'CustomWidget',
'before_widget' => '<div class = "widgetizedArea">',
'after_widget' => '</div>',
'before_title' => '<h3>',
'after_title' => '</h3>',
)
);
```

single.php

```
<?php if ( !function_exists('dynamic_sidebar') || !dynamic_sidebar("CustomWidget") ) :
$categories = get_categories();
```

```

foreach($categories as $category) {
// echo "test";

echo '<div class="col-md-4"><a href="" . get_category_link($category->term_id) . "">' .
$category->name . '</a></div>';

}

endif;?>

```

archive.php

```

<?php get_header(); ?>

<div class = "container pt-5 pb-5">

    <h1><?php single_cat_title(); ?></h1>

    <?php if (have_posts()) : while(have_posts()) : the_post();?>

    <div class= "card" style="width: 25rem;">

        <h5><?php the_title(); ?></h5>

        <!-- <?php the_excerpt();?> -->

        <a href = "<?php the_permalink(); ?>" class = " m-2" style="width: 18rem;">Read More</a>

    </div>

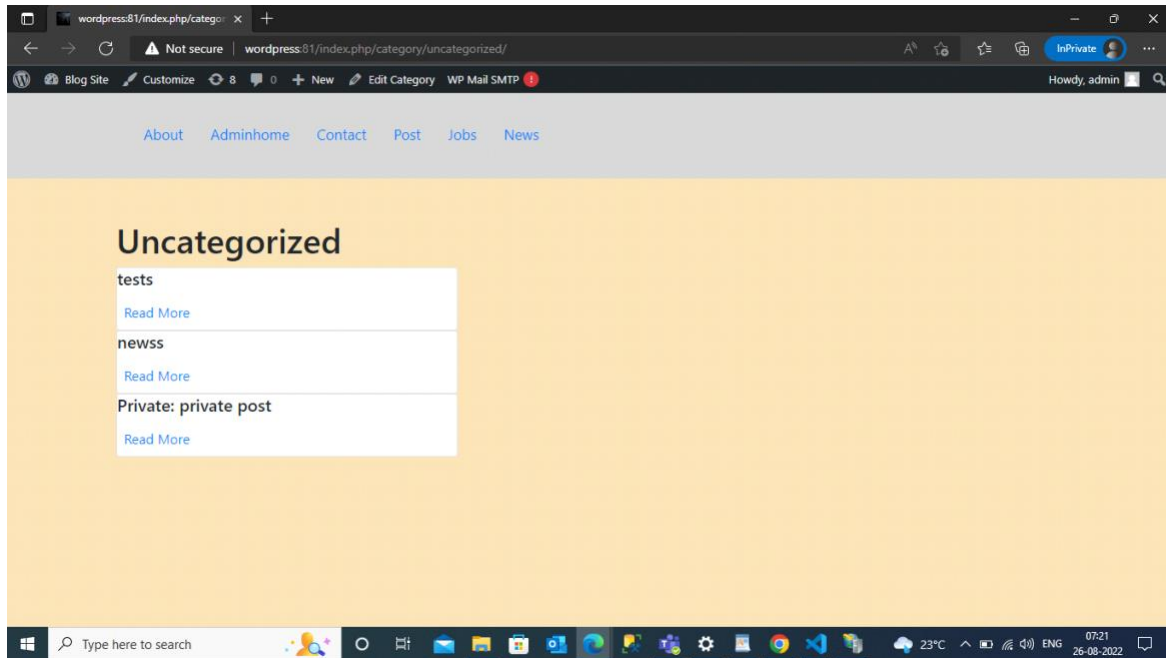
    <?php endwhile; endif;?>

</div>

<?php get_footer(); ?>

```

screenshort



11.Adding edit and delete button in custom post type

For this first remove default options from custom post type and added the column for new buttons. Then perform the edit and delete operations.

source code

Removing default options from post type

```
add_filter( 'post_row_actions', 'remove_row_actions_post', 10, 2 );
```

```
function remove_row_actions_post( $actions, $post ) {
```

```
    if( $post->post_type === 'jobs' ) {
```

```
        // global $post;
```

```
        // $postid = $post->ID;
```

```
        echo $postid;
```

```
        unset( $actions['edit'] );
```

```
        unset( $actions['trash'] );
```

```

        unset( $actions['view'] );

        unset( $actions['inline hide-if-no-js'] );

    }

    return $actions;

}

```

Adding the new buttons for operations

```

function add_column( $columns ){

    $columns['post_id_clmn'] = 'Action'; // $columns['Column ID'] = 'Column Title';

    return $columns;

}

add_filter('manage_posts_columns', 'add_column', 5);

// To get post id

function column_content( $column, $id ){

    if( $column === 'post_id_clmn')

        <?php $edit_link = get_edit_post_link( $id ); ?>

        <a href="<?=$edit_link ?>" >Edit</a>

        <?php $trash_link = get_delete_post_link( $id ); ?>

        <a href="<?=$trash_link ?>" >Delete</a>

    <?php

}

add_action('manage_posts_custom_column', 'column_content', 5, 2);

```

screenshort

jobs • Blog Site — WordPress

Not secure | wordpress:81/wp-admin/edit.php?post_type=jobs

Howdy, admin

Dashboard

Posts

jobs

Jobs

Add New

Media

Pages

Comments

news

Appearance

Plugins 7

Users

Tools

Settings

LoginWP

ProfilePress

Lazy Blocks

jobs [Add New](#)

Screen Options

ThemeGrill Demo Importer – This plugin requires [Official ThemeGrill Theme](#) to be activated to work.

Excellent! You've been using Forminator for a while now. Hope you are enjoying it so far. We have spent countless hours developing this free plugin for you, and we would really appreciate it if you could drop us a rating on wp.org to help us spread the word and boost our motivation.

[Rate Forminator](#) Maybe later

Are you enjoying WP Mail SMTP?

[Yes](#)
[Not Really](#)

Hey, I noticed you have been using LoginWP (Formerly Peter's Login Redirect) for a while now - that's awesome! Could you please do me a BIG favor and give it a [5-star rating on WordPress?](#) This will help us spread the word and boost our motivation - thanks!

[Sure! I'd love to give a review](#)
[Dismiss](#)

Hey, I noticed you have been using ProfilePress for at least 7 days now - that's awesome! Could you please do us a BIG favor and give it a [5-star rating on WordPress?](#) This will help us spread the word and boost our motivation - thanks!

wordpress:81/wp-admin/edit.php?post_type=jobs [Click to give a review](#)

Type here to search

23°C 06:57 26-08-2022