

Final exam development of dynamic website

Q1IN WORDPRESS WHAT ARE COMMENTS?

Comments are **a way for visitors to add feedback and reply to your posts and pages**. If you choose to enable comments on your site, then a comment form will appear at the bottom of the posts and people can respond to what you have written. Table of Contents. Enable/Disable Comments. In Settings.

Q2.IN WORDPRESS,WHERE IS USUALLY A COMMENT LIST LOCATED?

All comments on a WordPress website are displayed on **the Comments page in your WordPress admin area**. By default, it displays all comments.

Q3.IN WORDPRESS,WHERE IS IT POSSIBLE TO DEFINE SOME COMMENTS PARAMETER?

Navigate to the **Posts/Pages screen**. In the list of pages or posts, find the one you want and hover your cursor over the title of the post. You will see several links appear underneath the title. Click “Quick Edit” and check “Allow Comments.” Click “Update” to turn comments on for that post.

Q4.IN WORDPRESS,IN ORDER TO MODERATE COMMENTS,WHERE IN THE DASHBOARD WOULD YOU HAVE TO GO AND WHAT BASIC ACTIONS CAN THERE BE TAKEN?

Simply **go to Posts** » All Posts and click on the Edit link below the post you want to open. On the post editor screen scroll down to the bottom, and you will see all comments made for the post. You can edit, delete, reply and moderate comments directly from here.

Q5.EXPLAIN WHAT DOES THE FOLLOWING FUNCTION WHEN PLACED IN THE WORDPRESS LOOP;<?PHP COMMENTS_TEMPLATE();?>

```
function comments_template( $file = '/comments.php',  
$separate_comments = false ) {  
    global $wp_query, $withcomments, $post, $wpdb, $id,  
$comment, $user_login, $user_identity,  
$overridden_cpage;
```

```
    if ( ! ( is_single() || is_page() || $withcomments ) ||  
empty( $post ) ) {  
        return;  
    }
```

```
    if ( empty( $file ) ) {  
        $file = '/comments.php';  
    }
```

```
$req = get_option( 'require_name_email' );
```

```

/*
 * Comment author information fetched from the
comment cookies.
 */
$commenter = wp_get_current_commenter();

/*
 * The name of the current comment author escaped for
use in attributes.
 * Escaped by sanitize_comment_cookies().
 */
$comment_author = $commenter['comment_author'];

/*
 * The email address of the current comment author
escaped for use in attributes.
 * Escaped by sanitize_comment_cookies().
 */
$comment_author_email =
$commenter['comment_author_email'];

/*
 * The URL of the current comment author escaped for
use in attributes.
 */
$comment_author_url = esc_url(
$commenter['comment_author_url'] );

$comment_args = array(
    'orderby'          => 'comment_date_gmt',

```

```
'order'           => 'ASC',
'status'          => 'approve',
'post_id'         => $post->ID,
'no_found_rows'   => false,
'update_comment_meta_cache' => false, // We lazy-
load comment meta for performance.
);
```

```
if ( get_option( 'thread_comments' ) ) {
    $comment_args['hierarchical'] = 'threaded';
} else {
    $comment_args['hierarchical'] = false;
}
```

```
if ( is_user_logged_in() ) {
    $comment_args['include_unapproved'] = array(
get_current_user_id() );
} else {
    $unapproved_email =
wp_get_unapproved_comment_author_email();
```

```
    if ( $unapproved_email ) {
        $comment_args['include_unapproved'] = array(
$unapproved_email );
    }
}
```

```
$per_page = 0;
if ( get_option( 'page_comments' ) ) {
    $per_page = (int) get_query_var(
'comments_per_page' );
```

```

        if ( 0 === $per_page ) {
            $per_page = (int) get_option(
'comments_per_page' );
        }

        $comment_args['number'] = $per_page;
        $page
            = (int) get_query_var( 'cpage' );

        if ( $page ) {
            $comment_args['offset'] = ( $page - 1 ) *
$per_page;
        } elseif ( 'oldest' === get_option(
'default_comments_page' ) ) {
            $comment_args['offset'] = 0;
        } else {
            // If fetching the first page of 'newest', we need a
top-level comment count.
            $top_level_query = new WP_Comment_Query();
            $top_level_args = array(
                'count' => true,
                'orderby' => false,
                'post_id' => $post->ID,
                'status' => 'approve',
            );

            if ( $comment_args['hierarchical'] ) {
                $top_level_args['parent'] = 0;
            }

            if ( isset( $comment_args['include_unapproved'] ) )
{

```

```

        $top_level_args['include_unapproved'] =
$comment_args['include_unapproved'];
    }

    /**
     * Filters the arguments used in the top level
comments query.
     *
     * @since 5.6.0
     *
     * @see WP_Comment_Query::__construct()
     *
     * @param array $top_level_args {
     *     The top level query arguments for the
comments template.
     *
     *     @type bool      $count Whether to return a
comment count.
     *     @type string|array $orderby The field(s) to
order by.
     *     @type int        $post_id The post ID.
     *     @type string|array $status The comment
status to limit results by.
     * }
    */
    $top_level_args = apply_filters(
'comments_template_top_level_query_args',
$top_level_args );

    $top_level_count = $top_level_query->query(
$top_level_args );

```

```

        $comment_args['offset'] = ( ceil( $top_level_count /
$per_page ) - 1 ) * $per_page;
    }
}

```

```

/**
 * Filters the arguments used to query comments in
comments_template().
 *
 * @since 4.5.0
 *
 * @see WP_Comment_Query::__construct()
 *
 * @param array $comment_args {
 *     Array of WP_Comment_Query arguments.
 *
 *     @type string|array $orderby           Field(s) to
order by.
 *     @type string      $order             Order of
results. Accepts 'ASC' or 'DESC'.
 *     @type string      $status            Comment
status.
 *     @type array        $include_unapproved Array
of IDs or email addresses whose unapproved comments
 *                                     will be included in
results.
 *     @type int          $post_id          ID of the post.
 *     @type bool         $no_found_rows    Whether
to refrain from querying for found rows.

```

```

*    @type bool        $update_comment_meta_cache
Whether to prime cache for comment meta.
*    @type bool|string $hierarchical        Whether to
query for comments hierarchically.
*    @type int         $offset                Comment
offset.
*    @type int         $number                Number of
comments to fetch.
* }
*/
$comment_args = apply_filters(
'comments_template_query_args', $comment_args );

$comment_query = new WP_Comment_Query(
$comment_args );
$_comments     = $comment_query->comments;

// Trees must be flattened before they're passed to the
walker.
if ( $comment_args['hierarchical'] ) {
    $comments_flat = array();
    foreach ( $_comments as $_comment ) {
        $comments_flat[] = $_comment;
        $comment_children = $_comment->get_children(
            array(
                'format' => 'flat',
                'status' => $comment_args['status'],
                'orderby' => $comment_args['orderby'],
            )
        );
    }
};

```



```

        foreach ( $comment_children as $comment_child )
        {
            $comments_flat[] = $comment_child;
        }
    }
} else {
    $comments_flat = $_comments;
}

/**
 * Filters the comments array.
 *
 * @since 2.1.0
 *
 * @param array $comments Array of comments
supplied to the comments template.
 * @param int $post_ID Post ID.
 */
$wp_query->comments = apply_filters(
'comments_array', $comments_flat, $post->ID );

$comments = &$wp_query->comments;
$wp_query->comment_count = count( $wp_query-
>comments );
$wp_query->max_num_comment_pages =
$comment_query->max_num_pages;

if ( $separate_comments ) {
    $wp_query->comments_by_type =
separate_comments( $comments );

```

```
        $comments_by_type      = &$wp_query-  
>comments_by_type;  
    } else {  
        $wp_query->comments_by_type = array();  
    }
```

```
$overridden_cpage = false;
```

```
    if ( " == get_query_var( 'cpage' ) && $wp_query-  
>max_num_comment_pages > 1 ) {  
        set_query_var( 'cpage', 'newest' === get_option(  
'default_comments_page' ) ?  
get_comment_pages_count() : 1 );  
        $overridden_cpage = true;  
    }
```

```
    if ( ! defined( 'COMMENTS_TEMPLATE' ) ) {  
        define( 'COMMENTS_TEMPLATE', true );  
    }
```

```
$theme_template = STYLESPATH . $file;
```

```
/**  
 * Filters the path to the theme template file used for the  
comments template.  
 *  
 * @since 1.5.1  
 *  
 * @param string $theme_template The path to the  
theme template file.  
 */
```

```

    $include = apply_filters( 'comments_template',
    $theme_template );

    if ( file_exists( $include ) ) {
        require $include;
    } elseif ( file_exists( TEMPLATEPATH . $file ) ) {
        require TEMPLATEPATH . $file;
    } else { // Backward compat code will be removed in a
future release.
        require ABSPATH . WPINC . '/theme-
compat/comments.php';
    }
}

```

Q6.BRIEFLY EXPLAIN WHAT IS THE PLUGIN AKISMET USED FOR IN WORDPRESS?

Akismet is a **spam filtering service that filters spam from comments, trackbacks, and contact form messages**. The filter works by combining information about spam captured on all participating sites, and then using those spam rules to block future spam. Akismet is offered by Automattic, the company behind WordPress.com

Q7.BRIEFLY EXPLAIN WHAT IS CUSTOM TYPE IN WORDPRESS?

A custom post type is **nothing more than a regular post with a different post_type value in the database**. The post type of regular posts is post , pages use page , attachments use attachment and so on. You can now create your own to indicate the type of content created.

Q8.IN A WORDPRESS TEMPLATE,BREIFLY EXPLAIN WHAT IS THE PURPOSE OF THE FOLLOWING CODELOINE:

```
<?PHP IF(HAVE_POSTS() ): WHILE(HAVE_POST() ):THE_POSTS();?>
```

<?PHP ENDWHILE:ENDIF;?>

Using the Loop

The Loop should be placed in the [Theme](#)'s `index.php` and in any other Templates used to display post information.

Be sure to include the call for the header template at the top of your [Theme](#)'s templates. If you are using The Loop inside your own design (and your own design is not a template), set `WP_USE_THEMES` to `false`:

```
<?php define( 'WP_USE_THEMES', false ); get_header(); ?>
```

The loop starts here:

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

and ends here:

```
<?php endwhile; else : ?>

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

<?php endif; ?>
```

This is using PHP's alternative syntax for control structures, and could also be expressed as:

```
<?php

if ( have_posts() ) {

    while ( have_posts() ) {

        the_post();

        //

        // Post Content here
```

```

//

} // end while

} // end if

?>

```

Q9.BREIFLY EXPLAIN WHAT THE FOLLOWING CODELINE IS USED FOR?

```

<?PHP IF(IS_SINGULURE('MYPAGE')) {

function is_singular( $post_types = '' ) {
    global $wp_query;

    if ( ! isset( $wp_query ) ) {
        _doing_it_wrong( __FUNCTION__, __( 'Conditional query tags do not
work before the query is run. Before then, they always return false.' ),
'3.1.0' );
        return false;
    }

    return $wp_query->is_singular( $post_types );
}

```

Q10 BREIFLY EXPLAIN WHAT IS SIDEBAR AND WHAT IS IT USED FOR?

In computing, a sidebar is **a user interface element that displays a list of choices**. It typically appears as a column to the left of the main content, though it can appear on the right side as well. Both Windows and Mac OS X desktop windows include a sidebar by default.

Q11.BRIEFLY EXPLIAN WHAT IS WIDGET AND GIVE ONE EXAMPLE?

A widget is **an element of a graphical user interface (GUI) that displays information or provides a specific way for a user to interact with the operating system or an application**. ... Using Microsoft's Visual Basic, a widget can be implemented as or part of an ActiveX control.

Q12.WORDPRESS ALLOWS THE USE OF VARIOUS PLUGINS.BREIFLY EXPLAIN WHAT IS A PLUGIN AND GIVE AN EXAMPLE.

WordPress plugins are small software apps that integrate and run on top of the WordPress software. This allows you **to create almost any kind of website with WordPress** (not just blogs). For example, you can: Start an online store with WordPress using the WooCommerce plugin.

Q13.BREIFLY EXPLAIN WHAT IS THE PURPOSE OF THE GULTENBURG PLUG-IN.

The Gutenberg plugin is usually several version releases ahead of core WordPress, so **you can preview the new features of the WordPress block editor before they are introduced into WordPress core**. Running the Gutenberg plugin is a great way to get the latest new features of the WordPress block editor.

Q14.WHAT KIND OF PLU-IN IS <<SUCURI>>?
IT IS A SECURITY KIND OF PLU-IN .