

LAMPIRAN

```

<?php
/**
 * Confirms that the activation key
 that is sent in an email after a user
 signs
 * up for a new site matches the key
 for that user and then displays
 confirmation.
 *
 * @package WordPress
 */

define( 'WP_INSTALLING', true );

/** Sets up the WordPress
Environment. */
require __DIR__ . '/wp-load.php';

require __DIR__ . '/wp-blog-
header.php';

if ( ! is_multisite() ) {
    wp_redirect(
wp_registration_url() );
    die();
}

$valid_error_codes = array(
'already_active', 'blog_taken' );

list( $activate_path ) = explode( '?',
wp_unslash(
$_SERVER['REQUEST_URI'] ) );
$activate_cookie      = 'wp-activate-'
. COOKIEHASH;

$key = "";
$result = null;

if ( isset( $_GET['key'] ) && isset(
$_POST['key'] ) && $_GET['key']
!== $_POST['key'] ) {
    wp_die( __( 'A key value
mismatch has been detected. Please
follow the link provided in your
activation email.' ), __( 'An error
occurred during the activation' ), 400
);
} elseif ( ! empty( $_GET['key'] ) ) {
    $key = $_GET['key'];
} elseif ( ! empty( $_POST['key'] ) )
{
    $key = $_POST['key'];
}

if ( $key ) {

```

```

        $redirect_url =
remove_query_arg( 'key' );

        if ( remove_query_arg( false )
!== $redirect_url ) {

            setcookie(
$activate_cookie, $key, 0,
$activate_path, COOKIE_DOMAIN,
is_ssl(), true );

            wp_safe_redirect(
$redirect_url );

            exit;

        } else {

            $result =
wpmu_activate_signup( $key );

        }

    }

if ( null === $result && isset(
$_COOKIE[ $activate_cookie ] ) ) {

    $key = $_COOKIE[
$activate_cookie ];

    $result =
wpmu_activate_signup( $key );

    setcookie( $activate_cookie, '
', time() - YEAR_IN_SECONDS,
$activate_path, COOKIE_DOMAIN,
is_ssl(), true );

}

if ( null === $result || ( is_wp_error(
$result ) && 'invalid_key' ===
$result->get_error_code() ) ) {

    status_header( 404 );

} elseif ( is_wp_error( $result ) ) {

    $error_code = $result-
>get_error_code();

    if ( ! in_array( $error_code,
$valid_error_codes, true ) ) {

        status_header( 400 );

    }

}

nocache_headers();

if ( is_object( $wp_object_cache ) ) {

    $wp_object_cache-
>cache_enabled = false;

}

// Fix for page title.

$wp_query->is_404 = false;

/**

 * Fires before the Site Activation
page is loaded.

 *

```

```

* @since 3.0.0
*/
do_action( 'activate_header' );

/**
 * Adds an action hook specific to
this page.
 *
 * Fires on { @see 'wp_head' }.
 *
 * @since MU (3.0.0)
 */
function do_activate_header() {
    /**
     * Fires before the Site
Activation page is loaded.
     *
     * Fires on the { @see
'wp_head' } action.
     *
     * @since 3.0.0
     */
    do_action(
'activate_wp_head' );
}

add_action( 'wp_head',
'do_activate_header' );

/**
 * Loads styles specific to this page.
 *
 * @since MU (3.0.0)
 */
function wpmu_activate_stylesheet()
{
    ?>
    <style type="text/css">
        form { margin-top:
2em; }
        #submit, #key {
width: 90%; font-size: 24px; }
        #language { margin-
top: .5em; }
        .error { background:
#f66; }
        span.h3 { padding: 0
8px; font-size: 1.3em; font-weight:
600; }
    </style>
    <?php
}

add_action( 'wp_head',
'wpmu_activate_stylesheet' );

add_action( 'wp_head',
'wp_strict_cross_origin_referrer' );

add_filter( 'wp_robots',
'wp_robots_sensitive_page' );

```

```
get_header( 'wp-activate' );
```

```
$blog_details = get_blog_details();
```

```
?>
```

```
<div id="signup-content"  
class="widecolumn">
```

```
    <div class="wp-activate-  
container">
```

```
        <?php if ( ! $key ) { ?>
```

```
            <h2><?php _e(  
'Activation Key Required' ); ?></h2>
```

```
            <form  
name="activateform"  
id="activateform" method="post"  
action="<?php echo  
network_site_url( $blog_details-  
>path . 'wp-activate.php' ); ?>">
```

```
                <p>
```

```
                    <label  
for="key"><?php _e( 'Activation  
Key:' ); ?></label>
```

```
                        <br  
><input type="text" name="key"  
id="key" value="" size="50" />
```

```
                    </p>
```

```
                <p  
class="submit">
```

```
                    <input  
id="submit" type="submit"
```

```
name="Submit" class="submit"  
value="<?php esc_attr_e( 'Activate'  

```

```
</p>
```

```
</form>
```

```
<?php
```

```
    } else {
```

```
        if ( is_wp_error(  
$result ) && in_array( $result-  
>get_error_code(),  
$valid_error_codes, true ) ) {
```

```
            $signup =  
$result->get_error_data();
```

```
        ?>
```

```
            <h2><?php  
_e( 'Your account is now active!' );  
?></h2>
```

```
            <?php
```

```
                echo ' <p  
class="lead-in">;
```

```
                    if ( " ===  
$signup->domain . $signup->path ) {
```

```
                        printf(  
/* translators: 1: Login URL,  
2: Username, 3: User email address,  
4: Lost password URL. */
```

```
                            __('Your account has been  
activated. You may now <a  
href="%1$s">log in</a> to the site
```

using your chosen username of `“%2$s”`. Please check your email inbox at `%3$s` for your password and login instructions. If you do not receive an email, please check your junk or spam folder. If you still do not receive an email within an hour, you can `reset your password.`),

```
network_site_url(
$blog_details->path . 'wp-login.php',
'login' ),
```

```
$signup->user_login,
```

```
$signup->user_email,
```

```
wp_lostpassword_url()
```

```
);
```

```
} else {
```

```
printf(
```

```
/* translators: 1: Site URL, 2:
Username, 3: User email address, 4:
Lost password URL. */
```

```
__( 'Your site at %1$s is
active. You may now log in to your
site using your chosen username of
&#8220;%2$s&#8221;. Please check
your email inbox at %3$s for your
password and login instructions. If
you do not receive an email, please
check your junk or spam folder. If
```

you still do not receive an email within an hour, you can `reset your password.`),

```
sprintf( '<a
href="http://%1$s%2$s">%1$s%2$s
</a>', $signup->domain,
$blog_details->path ),
```

```
$signup->user_login,
```

```
$signup->user_email,
```

```
wp_lostpassword_url()
```

```
);
```

```
}
```

```
echo '</p>';
```

```
} elseif ( null ===
```

```
$result || is_wp_error( $result ) ) {
```

```
?>
```

```
<h2><?php
```

```
_e( 'An error occurred during the
activation' ); ?></h2>
```

```
<?php if (
is_wp_error( $result ) ) : ?>
```

```
<p><?php echo $result-
>get_error_message(); ?></p>
```

```
<?php endif;
```

```
?>
```

```
<?php
```

```

    } else {
        $url = isset(
$result['blog_id'] ) ? get_home_url(
(int) $result['blog_id'] ) : ";
        $user =
get_userdata( (int) $result['user_id']
);
        ?>
        <h2><?php
_e( 'Your account is now active!' );
?></h2>
        <div
id="signup-welcome">
            <p><span
class="h3"><?php _e( 'Username:' );
?></span> <?php echo $user-
>user_login; ?></p>
            <p><span
class="h3"><?php _e( 'Password:' );
?></span> <?php echo
$result['password']; ?></p>
        </div>
        <?php
        if ( $url &&
network_home_url( " , 'http' ) !==
$url ) :
            switch_to_blog( (int)
$result['blog_id'] );
            $login_url = wp_login_url();
            restore_current_blog();
            ?>
            <p
class="view">
                <?php
                /* translators: 1: Site URL, 2:
Login URL. */
                printf( __( 'Your account is
now activated. <a
href="%1$s">View your site</a> or
<a href="%2$s">Log in</a>' ), $url,
esc_url( $login_url ) );
                ?>
            </p>
            <?php else :
                ?>
                <p
class="view">
                    <?php
                    printf(
                        /* translators: 1: Login URL,
2: Network home URL. */
                        __( 'Your account is now
activated. <a href="%1$s">Log
in</a> or go back to the <a
href="%2$s">homepage</a>.' ),

```

```

        network_site_url(
$blog_details->path . 'wp-login.php',
'login' ),

        network_home_url(
$blog_details->path )

);

?>
</p>
<?php
endif;
    }
}
?>
</div>
</div>
<script type="text/javascript">
    var key_input =
document.getElementById('key');
    key_input &&
key_input.focus();
</script>
<?php
get_footer( 'wp-activate' );
<?php
/**
* Loads the WordPress environment
and template.
*
* @package WordPress
*/
if ( ! isset( $wp_did_header ) ) {
    $wp_did_header = true;
    // Load the WordPress
library.
    require_once __DIR__ . '/wp-
load.php';
    // Set up the WordPress
query.
    wp();
    // Load the theme template.
    require_once ABSPATH .
WPINC . '/template-loader.php';
}
<?php
/**
* The base configuration for
WordPress

```



```

*
* The wp-config.php creation script
uses this file during the installation.
* You don't have to use the web site,
you can copy this file to "wp-
config.php"
* and fill in the values.
*
* This file contains the following
configurations:
*
* * Database settings
* * Secret keys
* * Database table prefix
* * ABSPATH
*
* @link
https://wordpress.org/support/article/
editing-wp-config-php/
*
* @package WordPress
*/

// ** Database settings - You can get
this info from your web host ** //

/** The name of the database for
WordPress */
define( 'DB_NAME', 'tokosab' );

/** Database username */
define( 'DB_USER', 'root' );

/** Database password */
define( 'DB_PASSWORD', '' );

/** Database hostname */
define( 'DB_HOST', '127.0.0.1' );

/** Database charset to use in
creating database tables. */
define( 'DB_CHARSET', 'utf8mb4' );

/** The database collate type. Don't
change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+
* Authentication unique keys and
salts.
*
* Change these to different unique
phrases! You can generate these
using
* the { @link
https://api.wordpress.org/secret-
key/1.1/salt/ WordPress.org secret-
key service}.
*

```

* You can change these at any point in time to invalidate all existing cookies.

* This will force all users to have to log in again.

*

* @since 2.6.0

*/

```
define( 'AUTH_KEY',
'TY9Kyyv^&H6ru254Z[ @AB)5{DB
_)VoD`8&>r)1.ESuPHF}cdECW03k
pP)vF,te#~' );
```

```
define( 'SECURE_AUTH_KEY',
';=TVn=@ldadzbq.i_U?xF+d3~#*f
M&L}OA?sX()dS-
nq*yE^CW7bcH1x?7bqv+*' );
```

```
define( 'LOGGED_IN_KEY',
'GRUW3@/9Z]_ia.AL0s?a[&Lk;[A
VIU}I|W7>@g}__&#55]>+gqyEX(
2ik)BgqI%q' );
```

```
define( 'NONCE_KEY',
'dlR_zb(i)jGRYja#/cEdzNVX$Y@2
Rt.33T|hXX@YZaS-5|m;^E-
Kyl@ARWb#fs`7' );
```

```
define( 'AUTH_SALT',
'oziw?@u$!.B:G;afIbh$$`7fpegO~?
GGn<}<qI6jn,o+ELNx5II(C-O-
}1A(xh0' );
```

```
define( 'SECURE_AUTH_SALT',
'S<&!ytT2L5=:f.pr:uW4x1g5zh,3En
U1YCqm}7:S[(A9%;DJI;~IN2U>R.
+-Qe%' );
```

```
define( 'LOGGED_IN_SALT',
'Z3p1f3Q18kd99i+&MaLwk5Z
```

```
99(m&XB#v#WAw)WA+i(ru_/e@Z
Q0iqA*VMztLd-G' );
```

```
define( 'NONCE_SALT',
'&~7.`rJ0}!zP-
@p(pOFt&~=ZCRl]#RphqgcAZNO
g>XX-kLpNX%3NFdHf[^)]b`U1' );
```

```
/**#@_*/
```

```
/**
```

* WordPress database table prefix.

*

* You can have multiple installations in one database if you give each

* a unique prefix. Only numbers, letters, and underscores please!

*/

```
$table_prefix = 'wp_';
```

```
/**
```

* For developers: WordPress debugging mode.

*

* Change this to true to enable the display of notices during development.

* It is strongly recommended that plugin and theme developers use WP_DEBUG

```
* in their development
environments.

*

* For information on other constants
that can be used for debugging,

* visit the documentation.

*

* @link
https://wordpress.org/support/article/
debugging-in-wordpress/

*/

define( 'WP_DEBUG', false );
```

```
/* Add any custom values between
this line and the "stop editing" line.
*/
```

```
/* That's all, stop editing! Happy
publishing. */
```

```
/** Absolute path to the WordPress
directory. */
```

```
if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH',
    __DIR__ . '/' );
}
```

```
/** Sets up WordPress vars and
included files. */
```

```
require_once ABSPATH . 'wp-
settings.php';
```

```
<?php
```

```
/**
```

```
* A pseudo-cron daemon for
scheduling WordPress tasks.
```

```
*
```

```
* WP-Cron is triggered when the
site receives a visit. In the scenario
```

```
* where a site may not receive
enough visits to execute scheduled
tasks
```

```
* in a timely manner, this file can be
called directly or via a server
```

```
* cron daemon for X number of
times.
```

```
*
```

```
* Defining DISABLE_WP_CRON
as true and calling this file directly
are
```

```
* mutually exclusive and the latter
does not rely on the former to work.
```

```
*
```

```
* The HTTP request to this file will
not slow down the visitor who
happens to
```

```
* visit when a scheduled cron event
runs.
```

```
*
```

```

* @package WordPress
*/

ignore_user_abort( true );

/* Don't make the request block till
we finish, if possible. */

if ( function_exists(
'fastcgi_finish_request' ) &&
version_compare( phpversion(),
'7.0.16', '>=' ) ) {

    if ( ! headers_sent() ) {

        header( 'Expires:
Wed, 11 Jan 1984 05:00:00 GMT' );

        header( 'Cache-
Control: no-cache, must-revalidate,
max-age=0' );

    }

    fastcgi_finish_request();

}

if ( ! empty( $_POST ) || defined(
'DOING_AJAX' ) || defined(
'DOING_CRON' ) ) {

    die();

}

/**

* Tell WordPress we are doing the
cron task.

*

* @var bool

*/

define( 'DOING_CRON', true );

if ( ! defined( 'ABSPATH' ) ) {

    /** Set up WordPress
environment */

    require_once __DIR__ . '/wp-
load.php';

}

/**

* Retrieves the cron lock.

*

* Returns the uncached
`doing_cron` transient.

*

* @ignore

* @since 3.3.0

*

* @global wpdb $wpdb WordPress
database abstraction object.

*

```

```

* @return string|int|false Value of
the `doing_cron` transient, 0|false
otherwise.
*/
function _get_cron_lock() {
    global $wpdb;

    $value = 0;

    if (
wp_using_ext_object_cache() ) {
        /*
         * Skip local cache
and force re-fetch of doing_cron
transient
         * in case another
process updated the cache.
        */
        $value =
wp_cache_get( 'doing_cron',
'transient', true );
    } else {
        $row = $wpdb->
get_row( $wpdb->prepare(
"SELECT option_value FROM
$wpdb->options WHERE
option_name = %s LIMIT 1",
'_transient_doing_cron' ) );

        if ( is_object( $row ) )
        {
            $value =
$row->option_value;
        }
    }

    return $value;
}

$crons = wp_get_ready_cron_jobs();
if ( empty( $crons ) ) {
    die();
}

$gmt_time = microtime( true );

// The cron lock: a unix timestamp
from when the cron was spawned.

$doing_cron_transient =
get_transient( 'doing_cron' );

// Use global $doing_wp_cron lock,
otherwise use the GET lock. If no
lock, try to grab a new lock.
if ( empty( $doing_wp_cron ) ) {
    if ( empty(
$_GET['doing_wp_cron'] ) ) {
        // Called from
external script/job. Try setting a lock.
        if (
$doing_cron_transient && (
$doing_cron_transient +

```

```

WP_CRON_LOCK_TIMEOUT >
$gmt_time ) ) {
    return;
}

$doing_wp_cron
= sprintf( '%.22F', microtime( true )
);

$doing_cron_transient
= $doing_wp_cron;

set_transient(
'doing_cron', $doing_wp_cron );
} else {
    $doing_wp_cron =
_GET['doing_wp_cron'];
}
}

/*
 * The cron lock (a unix timestamp
set when the cron was spawned),
 * must match $doing_wp_cron (the
"key").
 */

if ( $doing_cron_transient !==
$doing_wp_cron ) {
    return;
}

foreach ( $crons as $timestamp =>
$cronhooks ) {
        if ( $timestamp > $gmt_time
) {
            break;
        }

        foreach ( $cronhooks as
$hook => $keys ) {
            foreach ( $keys as $k
=> $v ) {
                $schedule =
$v['schedule'];

                if ( $schedule )
                {
                    wp_reschedule_event(
$timestamp, $schedule, $hook,
$v['args'] );
                }

                wp_unschedule_event(
$timestamp, $hook, $v['args'] );
            }
        }
    }
}

/**
 * Fires
scheduled events.
 */

```



```

header( 'Content-Type: text/xml;
charset=' . get_option( 'blog_charset'
), true );

$link_cat = "";

if ( ! empty( $_GET['link_cat'] ) ) {

    $link_cat =
$_GET['link_cat'];

    if ( ! in_array( $link_cat,
array( 'all', '0' ), true ) ) {

        $link_cat = absint(
(string) urldecode( $link_cat ) );

    }

}

echo '<?xml version="1.0"? . ">\n";
?>
<opml version="1.0">

    <head>

        <title>

            <?php

                /* translators:
%s: Site title. */

                printf( __(
'Links for %s' ), esc_attr(
get_bloginfo( 'name', 'display' ) ) );

            ?>

        </title>

        <dateCreated><?php
echo gmdate( 'D, d M Y H:i:s' ); ?>
GMT</dateCreated>

```

```

<?php

/**

 * Fires in the OPML

header.

 *

 * @since 3.0.0

 */

do_action(
'opml_head' );

?>

</head>

<body>

<?php

if ( empty( $link_cat ) ) {

    $cats = get_categories(

        array(

            'taxonomy'

=> 'link_category',

            'hierarchical'

=> 0,

        )

    );

} else {

    $cats = get_categories(

        array(

            'taxonomy'

=> 'link_category',

```



```

        'hierarchical'
=> 0,
        'include' =>
$link_cat,
    )
);
}

foreach ( (array) $cats as $cat ) :
    /** This filter is documented
    in wp-includes/bookmark-
    template.php */
    $catname = apply_filters(
'link_category', $cat->name );

    ?>
<outline type="category"
title="<?php echo esc_attr( $catname
); ?>">
    <?php
        $bookmarks =
get_bookmarks( array( 'category' =>
$cat->term_id ) );
        foreach ( (array) $bookmarks
as $bookmark ) :
            /**
            * Filters the OPML
            outline link title text.
            *
            * @since 2.2.0
    ?>
    * @param string
$title The OPML outline title text.
    */
    $title = apply_filters(
'link_title', $bookmark->link_name );
    ?>
<outline text="<?php echo esc_attr(
$title ); ?>" type="link"
xmlUrl="<?php echo esc_url(
$bookmark->link_rss ); ?>"
htmlUrl="<?php echo esc_url(
$bookmark->link_url ); ?>"
updated="
    <?php
        if ( '0000-00-00
00:00:00' !== $bookmark-
>link_updated ) {
            echo
            $bookmark->link_updated; }
        ?>
    " />
    <?php
        endforeach; // $bookmarks
    ?>
</outline>
    <?php

```

```

endforeach; // $cats

?>
</body>
</opml>
<?php
/**
 * Bootstrap file for setting the
ABSPATH constant
 * and loading the wp-config.php
file. The wp-config.php
 * file will then load the wp-
settings.php file, which
 * will then set up the WordPress
environment.
 *
 * If the wp-config.php file is not
found then an error
 * will be displayed asking the visitor
to set up the
 * wp-config.php file.
 *
 * Will also search for wp-config.php
in WordPress' parent
 * directory to allow the WordPress
directory to remain
 * untouched.
 *
 * @package WordPress
 */

/** Define ABSPATH as this file's
directory */
if ( ! defined( 'ABSPATH' ) ) {
    define( 'ABSPATH',
__DIR__ . '/' );
}

/*
 * The error_reporting() function can
be disabled in php.ini. On systems
where that is the case,
 * it's best to add a dummy function
to the wp-config.php file, but as this
call to the function
 * is run prior to wp-config.php
loading, it is wrapped in a
function_exists() check.
 */
if ( function_exists( 'error_reporting'
) ) {
    /*
 * Initialize error reporting to
a known set of levels.
 *
 * This will be adapted in
wp_debug_mode() located in wp-
includes/load.php based on
WP_DEBUG.
 * @see
http://php.net/manual/en/errorfunc.co

```

nstants.php List of known error levels.

```
*/
    error_reporting(
E_CORE_ERROR |
E_CORE_WARNING |
E_COMPILE_ERROR | E_ERROR |
E_WARNING | E_PARSE |
E_USER_ERROR |
E_USER_WARNING |
E_RECOVERABLE_ERROR );
}
```

```
/*
```

```
* If wp-config.php exists in the
WordPress root, or if it exists in the
root and wp-settings.php
```

```
* doesn't, load wp-config.php. The
secondary check for wp-settings.php
has the added benefit
```

```
* of avoiding cases where the
current directory is a nested
installation, e.g. / is WordPress(a)
```

```
* and /blog/ is WordPress(b).
```

```
*
```

```
* If neither set of conditions is true,
initiate loading the setup process.
```

```
*/
```

```
if ( file_exists( ABSPATH . 'wp-
config.php' ) ) {
```

```
/** The config file resides in
ABSPATH */
```

```
    require_once ABSPATH .
'wp-config.php';
```

```
    } elseif ( @file_exists( dirname(
ABSPATH ) . '/wp-config.php' ) &&
! @file_exists( dirname( ABSPATH
) . '/wp-settings.php' ) ) {
```

```
/** The config file resides
one level above ABSPATH but is not
part of another installation */
```

```
    require_once dirname(
ABSPATH ) . '/wp-config.php';
```

```
    } else {
```

```
        // A config file doesn't exist.
```

```
        define( 'WPINC', 'wp-
includes' );
```

```
        require_once ABSPATH .
WPINC . '/load.php';
```

```
        // Standardize $_SERVER
variables across setups.
```

```
        wp_fix_server_vars();
```

```

        require_once ABSPATH .
WPINC . '/functions.php';

        $path = wp_guess_url() .
'/wp-admin/setup-config.php';

        /*

        * We're going to redirect to
setup-config.php. While this
shouldn't result

        * in an infinite loop, that's a
silly thing to assume, don't you
think? If

        * we're traveling in circles,
our last-ditch effort is "Need more
help?"

        */

        if ( false === strpos(
$_SERVER['REQUEST_URI'],
'setup-config' ) ) {

                header( 'Location: ' .
$path );

                exit;

        }

        define(
'WP_CONTENT_DIR', ABSPATH .
'wp-content' );

        require_once ABSPATH .
WPINC . '/version.php';

```

```

        wp_check_php_mysql_versions();

        wp_load_translations_early();

        // Die with an error message.

        $die = '<p>' . sprintf(

                /* translators: %s: wp-
config.php */

                __( "There doesn't
seem to be a %s file. I need this
before we can get started." ),

                '<code>wp-
config.php</code>'

        ) . '</p>';

        $die .= '<p>' . sprintf(

                /* translators: 1:
Documentation URL, 2: wp-
config.php */

                __( 'Need more help?
<a href="%1$s">Read the support
article on %2$s</a>.' ),

                __(
'https://wordpress.org/support/article/
editing-wp-config-php/' ),

                '<code>wp-
config.php</code>'

        ) . '</p>';

        $die .= '<p>' . sprintf(

                /* translators: %s: wp-
config.php */

```

__("You can create a %s file through a web interface, but this doesn't work for all server setups. The safest way is to manually create the file."),

```
'<code>wp-  
config.php</code>' ) . '</p>';  
  
$die .= '<p><a href="' . $path  
. '" class="button button-large">' .  
__( 'Create a Configuration File' ) .  
'</a></p>';
```

```
wp_die( $die, __( 'WordPress  
&rsquo; Error' ) );
```

```
}
```

```
<?php
```

```
/**
```

```
* WordPress User Page
```

```
*
```

```
* Handles authentication,  
registering, resetting passwords,  
forgot password,
```

```
* and other user handling.
```

```
*
```

```
* @package WordPress
```

```
*/
```

```
/** Make sure that the WordPress  
bootstrap has run before continuing.  
*/
```

```
require __DIR__ . '/wp-load.php';
```

```
// Redirect to HTTPS login if forced  
to use SSL.
```

```
if ( force_ssl_admin() && ! is_ssl() )  
{
```

```
    if ( 0 === strpos(  
$_SERVER['REQUEST_URI'], 'http'  
) ) {
```

```
        wp_safe_redirect(  
set_url_scheme(  
$_SERVER['REQUEST_URI'],  
'https' ) );
```

```
        exit;
```

```
    } else {
```

```
        wp_safe_redirect(  
'https://' .  
$_SERVER['HTTP_HOST'] .  
$_SERVER['REQUEST_URI'] );
```

```
        exit;
```

```
    }
```

```
}
```

```
/**
```

```
* Output the login page header.
```

```
*
```

```
* @since 2.1.0
```

```
*
```

```
* @global string $error  
Login error message set by
```

deprecated pluggable wp_login() function

* or plugins replacing it.

* @global bool|string \$interim_login Whether interim login modal is being displayed.
String 'success'

* upon successful login.

* @global string \$action The action that brought the visitor to the login page.

*

* @param string \$title Optional. WordPress login Page title to display in the `` element.</p>
<p>* Default 'Log In'.</p>
<p>* @param string \$message Optional. Message to display in header. Default empty.</p>
<p>* @param WP_Error \$wp_error Optional. The error to pass. Default is a WP_Error instance.</p>
<p>*/</p>
<pre>function login_header(\$title = 'Log In', \$message = '', \$wp_error = null)
{
 global \$error, \$interim_login, \$action;

 // Don't index any of these forms.</pre>
</div>
<div data-bbox="545 134 859 905" data-label="Text">
<pre> add_filter('wp_robots',
'wp_robots_sensitive_page');

 add_action('login_head',
'wp_strict_cross_origin_referrer');

 add_action('login_head',
'wp_login_viewport_meta');

 if (! is_wp_error(\$wp_error
)) {

 \$wp_error = new
WP_Error();

 }

 // Shake it!

 \$shake_error_codes = array(
'empty_password', 'empty_email',
'invalid_email', 'invalidcombo',
'empty_username',
'invalid_username',
'incorrect_password',
'retrieve_password_email_failure');

 /**

 * Filters the error codes array
for shaking the login form.

 *

 * @since 3.0.0

 *

 * @param string[]
\$shake_error_codes Error codes that
shake the login form.</pre>
</div>

```

        */

        $shake_error_codes =
apply_filters( 'shake_error_codes',
$shake_error_codes );

        if ( $shake_error_codes &&
$wp_error->has_errors() &&
in_array( $wp_error-
>get_error_code(),
$shake_error_codes, true ) ) {

                add_action(
'login_footer', 'wp_shake_js', 12 );

        }

        $login_title = get_bloginfo(
'name', 'display' );

        /* translators: Login screen
title. 1: Login screen name, 2:
Network or site name. */

        $login_title = sprintf( __(
'%1$s &lsaquo; %2$s &#8212;
WordPress' ), $title, $login_title );

        if ( wp_is_recovery_mode() )
{

                /* translators: %s:
Login screen title. */

                $login_title = sprintf(
__( 'Recovery Mode &#8212; %s' ),
$login_title );

        }

        /**
         * Filters the title tag content
         for login page.
         *
         * @since 4.9.0
         *
         * @param string $login_title
         The page title, with extra context
         added.
         * @param string $title
         The original page title.
         */

        $login_title = apply_filters(
'login_title', $login_title, $title );

        ?><!DOCTYPE html>

        <html <?php
language_attributes(); ?>>

        <head>

                <meta http-equiv="Content-
Type" content="<?php bloginfo(
'html_type' ); ?>; charset=<?php
bloginfo( 'charset' ); ?>" />

                <title><?php echo
$login_title; ?></title>

        <?php

        wp_enqueue_style( 'login' );

```

```

/*
    * Remove all stored post
    data on logging out.

    * This could be added by
    add_action('login_head'...) like
    wp_shake_js(),

    * but maybe better if it's not
    removable by plugins.
*/

```

```

if ( 'loggedout' ===
$wp_error->get_error_code() ) {
    ?>

```

```

<script>if("sessionStorage"
in window){try{for(var key in
sessionStorage){if(key.indexOf("wp-
autosave-")!=-
1){sessionStorage.removeItem(key)}
}}catch(e){}};</script>

```

```

<?php

```

```

}

```

```

/**
    * Enqueue scripts and styles
    for the login page.

    *

    * @since 3.1.0

    */

```

```

do_action(
'login_enqueue_scripts' );

```

```

/**
    * Fires in the login page
    header after scripts are enqueued.

```

```

*
* @since 2.1.0

```

```

*/
do_action( 'login_head' );

```

```

$login_header_url = __(
'https://wordpress.org/' );

```

```

/**
    * Filters link URL of the
    header logo above login form.

```

```

*
* @since 2.1.0

```

```

*
* @param string
$login_header_url Login header logo
URL.

```

```

*/
$login_header_url =
apply_filters( 'login_headerurl',
$login_header_url );

```

```

$login_header_title = "";

```

```

/**

```


* Filters the title attribute of the header logo above login form.

*

* @since 2.1.0

* @deprecated 5.2.0 Use { @see 'login_headertext' } instead.

*

* @param string

\$login_header_title Login header logo title attribute.

*/

\$login_header_title = apply_filters_deprecated(

 'login_headertitle',

 array(

 \$login_header_title),

 '5.2.0',

 'login_headertext',

 __('Usage of the title attribute on the login logo is not recommended for accessibility reasons. Use the link text instead.')

);

\$login_header_text = empty(\$login_header_title) ? __('Powered by WordPress') : \$login_header_title;

/**

* Filters the link text of the header logo above the login form.

*

* @since 5.2.0

*

* @param string

\$login_header_text The login header logo link text.

*/

\$login_header_text = apply_filters('login_headertext', \$login_header_text);

\$classes = array('login-action-' . \$action, 'wp-core-ui');

if (is_rtl()) {

 \$classes[] = 'rtl';

}

if (\$interim_login) {

 \$classes[] = 'interim-login';

?>

<style type="text/css">html{background-color: transparent;}</style>

<?php

```

        if ( 'success' ===
$interim_login ) {

                $classes[] =
'interim-login-success';

        }

}

        $classes[] = ' locale-' .
sanitize_html_class( strtolower(
str_replace( '_', '-', get_locale() ) ) );

/**

        * Filters the login page body
classes.

        *

        * @since 3.5.0

        *

        * @param string[] $classes
An array of body classes.

        * @param string $action
The action that brought the visitor to
the login page.

        */

        $classes = apply_filters(
'login_body_class', $classes, $action
);

?>

</head>

        <body class="login no-js
<?php echo esc_attr( implode( ' ',
$classes ) ); ?>">

                <script
type="text/javascript">

                        document.body.className =
document.body.className.replace('n
o-js','js');

                </script>

        <?php
/**

        * Fires in the login page
header after the body tag is opened.

        *

        * @since 4.6.0

        */

do_action( 'login_header' );

?>

        <div id="login">

                <h1><a href="<?php
echo esc_url( $login_header_url );
?>"><?php echo $login_header_text;
?></a></h1>

        <?php
/**

        * Filters the message to
display above the login form.

        *

```

```

* @since 2.1.0
*
* @param string $message
Login message text.
*/
$message = apply_filters(
'login_message', $message );

if ( ! empty( $message ) ) {
    echo $message . "\n";
}

// In case a plugin uses $error
rather than the $wp_errors object.
if ( ! empty( $error ) ) {
    $wp_error->add(
'error', $error );
    unset( $error );
}

if ( $wp_error->has_errors() )
{
    $errors = "";
    $messages = "";

    foreach ( $wp_error->
get_error_codes() as $code ) {

```

```

        $severity =
$wp_error->get_error_data( $code );

        foreach (
$wp_error->get_error_messages(
$code ) as $error_message ) {
            if (
'message' === $severity ) {

                $messages .= '
'.
$error_message . "<br />\n";
            } else {

                $errors .= '
'.
$error_message . "<br />\n";
            }
        }
    }

    if ( ! empty( $errors ) )
) {
        /**
         * Filters the
error messages displayed above the
login form.
         *
         * @since
2.1.0
         *
         * @param
string $errors Login error message.
         */

```

```

        echo '<div
id="login_error">' . apply_filters(
'login_errors', $errors ) . "</div>\n";
    }

    if ( ! empty(
$messages ) ) {
        /**
         * Filters
instructional messages displayed
above the login form.
         *
         * @since
2.5.0
         *
         * @param
string $messages Login messages.
         */

        echo '<p
class="message">' . apply_filters(
'login_messages', $messages ) .
"</p>\n";
    }
}

} // End of login_header().

/**
 * Outputs the footer for the login
page.
 *
         * @since 3.1.0
         *
         * @global bool|string
$interim_login Whether interim
login modal is being displayed.
String 'success'
         *
         * upon
successful login.
         *
         * @param string $input_id Which
input to auto-focus.
         */

function login_footer( $input_id = " )
{
    global $interim_login;

    // Don't allow interim logins
to navigate away from the page.

    if ( ! $interim_login ) {
        ?>
        <p id="backtoblog">
            <?php
                $html_link =
sprintf(
                    '<a
href="%s">%s</a>',
                    esc_url( home_url( '/' ) ),
                    sprintf(

```

```

        /* translators: %s: Site title.
*/
        _x( '&larr; Go to %s', 'site' ),
        get_bloginfo( 'title', 'display' )
    );
    /**
     * Filter the
     "Go to site" link displayed in the
     login page footer.
     *
     * @since
5.7.0
     *
     * @param
string $link HTML link to the home
URL of the current site.
     */
    echo
    apply_filters( 'login_site_html_link',
$html_link );
    ?>
</p>
<?php
    the_privacy_policy_link(
        '<div class="privacy-policy-page-
link">', '</div>' );
    }
    ?>
</div><?php // End of <div
id="login">. ?>
<?php
    if (
        ! $interim_login &&
    /**
     * Filters the
     Languages select input activation on
     the login screen.
     *
     * @since 5.9.0
     *
     * @param bool
Whether to display the Languages
select input on the login screen.
     */
    apply_filters(
'login_display_language_dropdown',
true )
    ) {
        $languages =
get_available_languages();

```

```

        if ( ! empty(
$languages ) ) {
            'selected' =>
determine_locale(),

            'show_available_translations'
=> false,

            'explicit_option_en_us'
=> true,

            'languages' =>
$languages,

        );

        /**

        * Filters default arguments for the
        * Languages select input on the login
        * screen.

        * @since 5.9.0

        *

        * @param array $args Arguments for
        * the Languages select input on the
        * login screen.

        */

        <span class="dashicons
dashicons-translation" aria-
hidden="true"></span>

        <span class="screen-reader-
text"><?php _e( 'Language' );
?></span>

        </label>

        <?php

        $args = array(

            'id' =>
'language-switcher-locales',

            'name' =>
'wp_lang',

```

```
wp_dropdown_languages(
apply_filters(
'login_language_dropdown_args',
$args ) );

?>
```

```
<?php if ( $interim_login ) {
?>
```

```
<input type="hidden"
name="interim-login" value="1" />
```

```
<?php } ?>
```

```
<?php if ( isset(
$_GET['redirect_to'] ) && " !=="
$_GET['redirect_to'] ) { ?>
```

```
<input type="hidden"
name="redirect_to" value="<?php
echo esc_url_raw(
$_GET['redirect_to'] ); ?>" />
```

```
<?php } ?>
```

```
<?php if ( isset(
$_GET['action'] ) && " !=="
$_GET['action'] ) { ?>
```

```
<input type="hidden"
name="action" value="<?php echo
esc_attr( $_GET['action'] ); ?>" />
```

```
<?php } ?>
```

```
<input type="submit"
class="button" value="<?php
esc_attr_e( 'Change' ); ?>">
```

```
</form>
```

```
</div>
```

```
<?php } ?>
```

```
<?php } ?>
```

```
<?php
```

```
if ( ! empty( $input_id ) ) {
```

```
?>
```

```
<script
type="text/javascript">
```

```
try{ document.getElementById(
'<?php echo $input_id;
?>').focus();}catch(e){ }
```

```
if(typeof
wpOnload==='function')wpOnload();
```

```
</script>
```

```
<?php
```

```

    }

    /**
     * Fires in the login page
    footer.
     *
     * @since 3.1.0
     */
    do_action( 'login_footer' );

?>
<div class="clear"></div>
</body>
</html>
<?php
}

/**
 * Outputs the JavaScript to handle
the form shaking on the login page.
 *
 * @since 3.0.0
 */
function wp_shake_js() {
    ?>
    <script
type="text/javascript">

        document.querySelector('form').classList.add('shake');

    </script>
    <?php
}

/**
 * Outputs the viewport meta tag for
the login page.
 *
 * @since 3.7.0
 */
function wp_login_viewport_meta()
{
    ?>
    <meta name="viewport"
content="width=device-width" />
    <?php
}

/*
 * Main part: check the request and
redirect or display a form based on
the current action.
 */
$action = isset(
$_REQUEST['action'] ) ?
$_REQUEST['action'] : 'login';

```



```

$errors = new WP_Error();

if ( isset( $_GET['key'] ) ) {
    $action = 'resetpass';
}

if ( isset( $_GET['checkemail'] ) ) {
    $action = 'checkemail';
}

$default_actions = array(
    'confirm_admin_email',
    'postpass',
    'logout',
    'lostpassword',
    'retrievepassword',
    'resetpass',
    'rp',
    'register',
    'checkemail',
    'confirmation',
    'login',
    WP_Recovery_Mode_Link_
Service::LOGIN_ACTION_ENTER
ED,
);

// Validate action so as to default to
the login screen.
if ( ! in_array( $action,
$default_actions, true ) && false
=== has_filter( 'login_form_' .
$action ) ) {
    $action = 'login';
}

nocache_headers();

header( 'Content-Type: ' .
get_bloginfo( 'html_type' ) . ' ;
charset=' . get_bloginfo( 'charset' ) );

if ( defined( 'RELOCATE' ) &&
RELOCATE ) { // Move flag is set.
    if ( isset(
$_SERVER['PATH_INFO'] ) && (
$_SERVER['PATH_INFO'] !==
$_SERVER['PHP_SELF'] ) ) {
        $_SERVER['PHP_SELF'] =
str_replace(
$_SERVER['PATH_INFO'], "",
$_SERVER['PHP_SELF'] );
    }

    $url = dirname(
set_url_scheme( 'http://' .

```

```

$_SERVER['HTTP_HOST'] .
$_SERVER['PHP_SELF'] );

    if ( get_option( 'siteurl' ) !==
$url ) {

        update_option(
'siteurl', $url );

    }

}

// Set a cookie now to see if they are
supported by the browser.

$secure = ( 'https' === parse_url(
wp_login_url(),
PHP_URL_SCHEME ) );

setcookie( TEST_COOKIE, 'WP
Cookie check', 0, COOKIEPATH,
COOKIE_DOMAIN, $secure );

if ( SITECOOKIEPATH !==
COOKIEPATH ) {

    setcookie( TEST_COOKIE,
'WP Cookie check', 0,
SITECOOKIEPATH,
COOKIE_DOMAIN, $secure );

}

if ( isset( $_GET['wp_lang'] ) ) {

    setcookie( 'wp_lang',
sanitize_text_field(
$_GET['wp_lang'] ), 0,
COOKIEPATH,
COOKIE_DOMAIN, $secure );

}

do_action( 'login_init' );

/**
 * Fires when the login form is
initialized.
 *
 * @since 3.2.0
 */
do_action( 'login_init' );

/**
 * Fires before a specified login form
action.
 *
 * The dynamic portion of the hook
name, ` $action `, refers to the action
 * that brought the visitor to the login
form.
 *
 * Possible hook names include:
 *
 * - `login_form_checkemail`
 * -
`login_form_confirm_admin_email`
 * - `login_form_confirmation`

```

```

* -
`login_form_entered_recovery_mode`
`
* - `login_form_login`
* - `login_form_logout`
* - `login_form_lostpassword`
* - `login_form_postpass`
* - `login_form_register`
* - `login_form_resetpass`
* - `login_form_retrievepassword`
* - `login_form_rp`
*
* @since 2.8.0
*/
do_action( "login_form_{ $action}"
);

$http_post = ( 'POST' ===
$_SERVER['REQUEST_METHOD'
] );

$interim_login = isset(
$_REQUEST['interim-login'] );

/**
* Filters the separator used between
login form navigation links.
*
* @since 4.9.0
*/
*/
* @param string
$login_link_separator The separator
used between login form navigation
links.
*/
$login_link_separator =
apply_filters( 'login_link_separator', '
| ' );

switch ( $action ) {

    case 'confirm_admin_email':

        /*
        * Note that
`is_user_logged_in()` will return
false immediately after logging in
        * as the current user
is not set, see wp-
includes/pluggable.php.
        * However this action
runs on a redirect after logging in.
        */
        if ( !
is_user_logged_in() ) {

            wp_safe_redirect(
wp_login_url() );

            exit;

        }
}

```

```

        if ( ! empty(
$_REQUEST['redirect_to'] ) ) {
            $redirect_to =
$_REQUEST['redirect_to'];
        } else {
            $redirect_to =
admin_url();
        }

```

```

        if ( current_user_can(
'manage_options' ) ) {
            $admin_email
= get_option( 'admin_email' );
        } else {

```

```

        wp_safe_redirect(
$redirect_to );
        exit;
    }

```

```

/**
 * Filters the interval
for dismissing the admin email
confirmation screen.
 *
 * If `0` (zero) is
returned, the "Remind me later" link
will not be displayed.
 *
 * @since 5.3.1

```

```

 *
 * @param int
$interval Interval time (in seconds).
Default is 3 days.
 */

```

```

$remind_interval =
(int) apply_filters(
'admin_email_remind_interval', 3 *
DAY_IN_SECONDS );

```

```

        if ( ! empty(
$_GET['remind_me_later'] ) ) {
            if ( !
wp_verify_nonce(
$_GET['remind_me_later'],
'remind_me_later_nonce' ) ) {

```

```

        wp_safe_redirect(
wp_login_url() );
        exit;
    }

```

```

        if (
$remind_interval > 0 ) {
            update_option(
'admin_email_lifespan', time() +
$remind_interval );
        }

```

```

        $redirect_to =
add_query_arg(

```

```

'admin_email_remind_later', 1,
$redirect_to );

        wp_safe_redirect(
$redirect_to );

                exit;

        }

        if ( ! empty(
$_POST['correct-admin-email'] ) ) {

                if ( !
check_admin_referer(
'confirm_admin_email',
'confirm_admin_email_nonce' ) ) {

                        wp_safe_redirect(
wp_login_url() );

                                exit;

                }

                /**
                 * Filters the
                 interval for redirecting the user to the
                 admin email confirmation screen.
                 *
                 * If `0` (zero)
                 is returned, the user will not be
                 redirected.
                 *
                 * @since

```

```

*
* @param int
$interval Interval time (in seconds).
Default is 6 months.
*/

        $admin_email_check_interva
l = (int) apply_filters(
'admin_email_check_interval', 6 *
MONTH_IN_SECONDS );

        if (
$admin_email_check_interval > 0 ) {

                update_option(
'admin_email_lifespan', time() +
$admin_email_check_interval );

        }

        wp_safe_redirect(
$redirect_to );

                exit;

        }

        login_header( __(
'Confirm your administration email'
), ", $errors );

        /**

```

5.3.0

* Fires before the admin email confirm form.

*

* @since 5.3.0

*

* @param WP_Error \$errors A `WP_Error` object containing any errors generated by using invalid

*

credentials. Note that the error object may not contain any errors.

*/

do_action('admin_email_confirm', \$errors);

?>

<form class="admin-email-confirm-form" name="admin-email-confirm-form" action="<?php echo esc_url(site_url('wp-login.php?action=confirm_admin_email', 'login_post')); ?>" method="post">

<?php

/**

* Fires inside the admin-email-confirm-form form tags, before the hidden fields.

*

* @since 5.3.0

*/

do_action('admin_email_confirm_form');

wp_nonce_field('confirm_admin_email', 'confirm_admin_email_nonce');

?>

<input type="hidden" name="redirect_to" value="<?php echo esc_attr(\$redirect_to); ?>" />

<h1 class="admin-email__heading">

<?php _e('Administration email verification'); ?>

</h1>

<p class="admin-email__details">

<?php _e('Please verify that the administration email for this website is still correct.'); ?>

<?php

```
/*
translators: URL to the WordPress
help section about admin email. */
```

```
$admin_email_help_url = __(
'https://wordpress.org/support/article/
settings-general-screen/#email-
address' );
```

```
/*
translators: Accessibility text. */
```

```
$accessibility_text = sprintf(
'<span class="screen-reader-text">
%s</span>', __( '(opens in a new
tab)' ) );
```

```
printf(
'<a href="%s" rel="noopener"
target="_blank">%s%s</a>',
```

```
esc_url(
$admin_email_help_url ),
__( 'Why is this important?' ),
```

```
$accessibility_text
);
```

```
?>
</p>
```

```
<p
class="admin-email__details">
<?php
```

```
printf(
/* translators: %s: Admin
email address. */
```

```
__( 'Current administration
email: %s' ),
```

```
'<strong>' . esc_html(
$admin_email ) . '</strong>'
```

```
);
?>
```

```
</p>
<p
class="admin-email__details">
```

```
<?php
_e( 'This email may be different from
your personal email address.' ); ?>
```

```
</p>
<div
class="admin-email__actions">
<div
class="admin-email__actions-
primary">
```

```

<?php

    $change_link = admin_url(
'options-general.php' );

    $change_link =
add_query_arg( 'highlight',
'confirm_admin_email',
$change_link );

    $remind_me_link =
wp_login_url( $redirect_to );

    $remind_me_link =
add_query_arg(
        array(
            'action'
=> 'confirm_admin_email',

            'remind_me_later' =>
wp_create_nonce(
'remind_me_later_nonce' ),

        ),

        $remind_me_link
    );

    <a class="button button-
large" href="<?php echo esc_url(
$change_link ); ?>"><?php _e(
'Update' ); ?></a>

    <input type="submit"
name="correct-admin-email"
id="correct-admin-email"
class="button button-primary button-
large" value="<?php esc_attr_e( 'The
email is correct' ); ?>" />

</div>

<?php
if ( $remind_interval > 0 ) : ?>

    <div class="admin-
email__actions-secondary">

        <a href="<?php echo esc_url(
$remind_me_link ); ?>"><?php _e(
'Remind me later' ); ?></a>

    </div>

<?php

```


<pre> endif; ?> </div> </form> <?php login_footer(); break; case 'postpass': if (! array_key_exists('post_password', \$_POST)) { wp_safe_redirect(wp_get_referer()); exit; } require_once ABSPATH . WPINC . '/class- phpass.php'; \$hasher = new PasswordHash(8, true); /** * Filters the life span of the post password cookie. </pre>	<pre> * * By default, the cookie expires 10 days from creation. To turn this * into a session cookie, return 0. * * @since 3.7.0 * * @param int \$expires The expiry time, as passed to setcookie(). */ \$expire = apply_filters('post_password_expires', time() + 10 * DAY_IN_SECONDS); \$referer = wp_get_referer(); if (\$referer) { \$secure = ('https' === parse_url(\$referer, PHP_URL_SCHEME)); } else { \$secure = false; } setcookie('wp- postpass_' . COOKIEHASH, </pre>
---	---

```

$hasher->HashPassword(
    wp_unslash(
        $_POST['post_password'] ) ),
    $expire, COOKIEPATH,
    COOKIE_DOMAIN, $secure );

    'loggedout' => 'true',

    'wp_lang' =>
    get_user_locale( $user ),

),

    wp_safe_redirect(
wp_get_referer() );

    wp_login_url()

    exit;

);

    case 'logout':

        check_admin_referer(
'log-out' );

        $requested_redirect_to = "";
    }

        $user =
wp_get_current_user();

        /**
         * Filters the log out
         redirect URL.
         *
         * @since 4.2.0
         *
         * @param string
         $redirect_to The redirect
         destination URL.
         *
         * @param string
         $requested_redirect_to The requested
         redirect destination URL passed as a
         parameter.
         *
         * @param WP_User
         $user The WP_User object
         for the user that's logging out.
         */

wp_logout();

    if ( ! empty(
$_REQUEST['redirect_to'] ) ) {

        $redirect_to
= $_REQUEST['redirect_to'];

        $requested_redirect_to =
$redirect_to;

    } else {

        $redirect_to =

add_query_arg(

        array(

```

```

        $redirect_to =
apply_filters( 'logout_redirect',
$redirect_to, $requested_redirect_to,
$user );

        wp_safe_redirect(
$redirect_to );

        exit;

        case 'lostpassword':
        case 'retrievepassword':
            if ( $http_post ) {
                $errors =
retrieve_password();

                if ( !
is_wp_error( $errors ) ) {

                    $redirect_to = ! empty(
$_REQUEST['redirect_to'] ) ?
$_REQUEST['redirect_to'] : 'wp-
login.php?checkemail=confirm';

                    wp_safe_redirect(
$redirect_to );

                    exit;
                }
            }

            if ( isset(
$_GET['error'] ) ) {
                if ( 'invalidkey'
=== $_GET['error'] ) {

                    $errors->add( 'invalidkey',
__( '<strong>Error</strong>: Your
password reset link appears to be
invalid. Please request a new link
below.' ) );

                } elseif (
'expiredkey' === $_GET['error'] ) {

                    $errors->add( 'expiredkey',
__( '<strong>Error</strong>: Your
password reset link has expired.
Please request a new link below.' ) );

                }

            }

            $lostpassword_redirect = !
empty( $_REQUEST['redirect_to'] )
? $_REQUEST['redirect_to'] : "";

            /**
             * Filters the URL
            redirected to after submitting the
            lostpassword/retrievepassword form.
             *
             * @since 3.0.0
             *
             * @param string
            $lostpassword_redirect The redirect
            destination URL.
             */

```

```

        $redirect_to =
apply_filters( 'lostpassword_redirect',
$lostpassword_redirect );

        /**
        * Fires before the lost
password form.
        *
        * @since 1.5.1
        * @since 5.1.0
Added the `$errors` parameter.
        *
        * @param WP_Error
$errors A `WP_Error` object
containing any errors generated by
using invalid
        *
credentials. Note that the error object
may not contain any errors.
        */
do_action(
'lost_password', $errors );

        login_header( __(
'Lost Password' ), ' <p
class="message">' . __( 'Please enter
your username or email address. You
will receive an email message with
instructions on how to reset your
password.' ) . ' </p>', $errors );

        $user_login = ";

```

```

        if ( isset(
$_POST['user_login'] ) &&
is_string( $_POST['user_login'] ) ) {
            $user_login =
wp_unslash( $_POST['user_login'] );
        }
    }
    ?>
    <form
name="lostpasswordform"
id="lostpasswordform"
action="<?php echo esc_url(
network_site_url( 'wp-
login.php?action=lostpassword',
'login_post' ) ); ?>" method="post">
        <p>
            <label
for="user_login"><?php _e(
'Username or Email Address' );
?></label>
            <input
type="text" name="user_login"
id="user_login" class="input"
value="<?php echo esc_attr(
$user_login ); ?>" size="20"
autocapitalize="off"
autocomplete="username" />
        </p>
    <?php

```

```

        /**

```

* Fires inside
the lostpassword form tags, before
the hidden fields.

2.1.0

```
* Fires inside
the lostpassword form tags, before
the hidden fields.

*
* @since
*/
do_action(
'lostpassword_form' );

?>

<input
type="hidden" name="redirect_to"
value="<?php echo esc_attr(
$redirect_to ); ?>" />

<p
class="submit">

<input
type="submit" name="wp-submit"
id="wp-submit" class="button
button-primary button-large"
value="<?php esc_attr_e( 'Get New
Password' ); ?>" />

</p>

</form>

<p id="nav">

<a
href="<?php echo esc_url(
wp_login_url() ); ?>"><?php _e(
'Log in' ); ?></a>

<?php
```

```
if ( get_option(
'users_can_register' ) ) {

    $registration_url = sprintf(
'<a href="%s">%s</a>', esc_url(
wp_registration_url() ), __( 'Register'
) );

    echo
esc_html( $login_link_separator );

/**
This filter is documented in wp-
includes/general-template.php */

    echo
apply_filters( 'register',
$registration_url );
}

?>

</p>

<?php

    login_footer(
'user_login' );

    break;

    case 'resetpass':

    case 'rp':
```

```

        list( $rp_path ) =
explode( '?', wp_unslash(
$_SERVER['REQUEST_URI'] ) );

        $rp_cookie = 'wp-
resetpass-' . COOKIEHASH;

        if ( isset(
$_GET['key'] ) && isset(
$_GET['login'] ) ) {

                $value =
sprintf( '%s:%s', wp_unslash(
$_GET['login'] ), wp_unslash(
$_GET['key'] ) );

                setcookie(
$rp_cookie, $value, 0, $rp_path,
COOKIE_DOMAIN, is_ssl(), true );

        wp_safe_redirect(
remove_query_arg( array( 'key',
'login' ) ) );

                exit;

        }

        if ( isset( $_COOKIE[
$rp_cookie ] ) && 0 < strpos(
$_COOKIE[ $rp_cookie ], ':' ) ) {

                list( $rp_login,
$rp_key ) = explode( ':', wp_unslash(
$_COOKIE[ $rp_cookie ] ), 2 );

```

```

        $user =
check_password_reset_key( $rp_key,
$rp_login );

        if ( isset(
$_POST['pass1'] ) && !
hash_equals( $rp_key,
$_POST['rp_key'] ) ) {

                $user =
false;

        } else {

                $user = false;

        }

        if ( ! $user ||
is_wp_error( $user ) ) {

                setcookie(
$rp_cookie, '', time() -
YEAR_IN_SECONDS, $rp_path,
COOKIE_DOMAIN, is_ssl(), true );

        if ( $user &&
$user->get_error_code() ===
'expired_key' ) {

                wp_redirect( site_url( 'wp-
login.php?action=lostpassword&erro
r=expiredkey' ) );

        } else {

                wp_redirect( site_url( 'wp-

```

```
login.php?action=lostpassword&erro  
r=invalidkey' ) );
```

```
}
```

```
exit;
```

```
}
```

```
$errors = new  
WP_Error();
```

```
// Check if password  
is one or all empty spaces.
```

```
if ( ! empty(  
$_POST['pass1'] ) ) {
```

```
$_POST['pass1'] = trim(  
$_POST['pass1'] );
```

```
if ( empty(  
$_POST['pass1'] ) ) {
```

```
$errors->add(  
'password_reset_empty_space', __(  
'The password cannot be a space or  
all spaces.' ) );
```

```
}
```

```
}
```

```
// Check if password  
fields do not match.
```

```
if ( ! empty(  
$_POST['pass1'] ) && trim(  
$_POST['pass2'] ) !==  
$_POST['pass1'] ) {
```

```
$errors->add(  
'password_reset_mismatch', __(  
'<strong>Error</strong>: The  
passwords do not match.' ) );
```

```
}
```

```
/**
```

```
* Fires before the  
password reset procedure is  
validated.
```

```
*
```

```
* @since 3.5.0
```

```
*
```

```
* @param WP_Error  
$errors WP Error object.
```

```
* @param  
WP_User|WP_Error $user  
WP_User object if the login and reset  
key match. WP_Error object  
otherwise.
```

```
*/
```

```
do_action(  
'validate_password_reset', $errors,  
$user );
```

```
if ( ( ! $errors-  
>has_errors() ) && isset(  
$_POST['pass1'] ) && ! empty(  
$_POST['pass1'] ) ) {
```

```

        reset_password( $user,
$_POST['pass1'] );

        setcookie(
$rp_cookie, ' ', time() -
YEAR_IN_SECONDS, $rp_path,
COOKIE_DOMAIN, is_ssl(), true );

        login_header(
__( 'Password Reset' ), '<p
class="message reset-pass">'. __(
'Your password has been reset.' ) . '
<a href="" . esc_url( wp_login_url() )
. "">'. __( 'Log in' ) . '</a></p>');

        login_footer();

        exit;
    }

    wp_enqueue_script(
'utils' );

    wp_enqueue_script(
'user-profile' );

        login_header( __(
'Reset Password' ), '<p
class="message reset-pass">'. __(
'Enter your new password below or
generate one.' ) . '</p>', $errors );

    ?>

    <form
name="resetpassform"
id="resetpassform" action="<?php
echo esc_url( network_site_url( 'wp-

login.php?action=resetpass',
'login_post' ) ); ?>" method="post"
autocomplete="off">

        <input
type="hidden" id="user_login"
value="<?php echo esc_attr(
$rp_login ); ?>" autocomplete="off"
/>

        <div
class="user-pass1-wrap">

            <p>

                <label for="pass1"><?php
_e( 'New password' ); ?></label>

            </p>

            <div
class="wp-pwd">

                <input type="password" data-
reveal="1" data-pw="<?php echo
esc_attr( wp_generate_password( 16
) ); ?>" name="pass1" id="pass1"
class="input password-input"
size="24" value=""
autocomplete="new-password" aria-
describedby="pass-strength-result"
/>

                <button type="button"
class="button button-secondary wp-
hide-pw hide-if-no-js" data-

```



```
toggle="0" aria-label="<?php
esc_attr_e( 'Hide password' ); ?>">
```

```
<span class="dashicons
dashicons-hidden" aria-
hidden="true"></span>
```

```
</button>
```

```
<div id="pass-strength-
result" class="hide-if-no-js" aria-
live="polite"><?php _e( 'Strength
indicator' ); ?></div>
```

```
</div>
```

```
<div
```

```
class="pw-weak">
```

```
<input type="checkbox"
name="pw_weak" id="pw-weak"
class="pw-checkbox" />
```

```
<label for="pw-weak"><?php
_e( 'Confirm use of weak password'
); ?></label>
```

```
</div>
```

```
</div>
```

```
<p
class="user-pass2-wrap">
```

```
<label
for="pass2"><?php _e( 'Confirm
new password' ); ?></label>
```

```
<input
type="password" name="pass2"
id="pass2" class="input" size="20"
value="" autocomplete="new-
password" />
```

```
</p>
```

```
<p
class="description indicator-
hint"><?php echo
wp_get_password_hint(); ?></p>
```

```
<br
class="clear" />
```

```
<?php
```

```
/**
 * Fires
 following the 'Strength indicator'
 meter in the user password reset
 form.
```

```
*
 * @since
3.9.0
```

```
*
 * @param
WP_User $user User object of the
user whose password is being reset.
```

```
*/
do_action(
'resetpass_form', $user );
```

<pre> ?> <input type="hidden" name="rp_key" value="<?php echo esc_attr(\$rp_key); ?>" /> <p class="submit reset-pass-submit"> <button type="button" class="button wp-generate-pw hide- if-no-js skip-aria-expanded"><?php _e('Generate Password'); ?></button> <input type="submit" name="wp-submit" id="wp-submit" class="button button-primary button-large" value="<?php esc_attr_e('Save Password'); ?>" /> </p> </form> <p id="nav"> <a href="<?php echo esc_url(wp_login_url()); ?>"><?php _e('Log in'); ?> <?php if (get_option('users_can_register')) { </pre>	<pre> \$registration_url = sprintf('%s', esc_url(wp_registration_url()), __('Register')); echo esc_html(\$login_link_separator); /** This filter is documented in wp- includes/general-template.php */ echo apply_filters('register', \$registration_url); } ?> </p> <?php login_footer('pass1'); break; case 'register': if (is_multisite()) { /** * Filters the Multisite sign up URL. * </pre>
---	--

```

3.0.0
    * @since
    *
    * @param
string $sign_up_url The sign up
URL.
    */
    wp_redirect(
apply_filters( 'wp_signup_location',
network_site_url( 'wp-signup.php' ) )
);
    exit;
}

    if ( ! get_option(
'users_can_register' ) ) {
        wp_redirect(
site_url( 'wp-
login.php?registration=disabled' ) );
        exit;
    }

    $user_login = "";
    $user_email = "";

    if ( $http_post ) {
        if ( isset(
$_POST['user_login'] ) &&
is_string( $_POST['user_login'] ) ) {
            $user_login = wp_unslash(
$_POST['user_login'] );
        }

        if ( isset(
$_POST['user_email'] ) &&
is_string( $_POST['user_email'] ) ) {
            $user_email = wp_unslash(
$_POST['user_email'] );
        }

        $errors =
register_new_user( $user_login,
$user_email );

        if ( !
is_wp_error( $errors ) ) {
            $redirect_to = ! empty(
$_POST['redirect_to'] ) ?
$_POST['redirect_to'] : 'wp-
login.php?checkemail=registered';

            wp_safe_redirect(
$redirect_to );

            exit;
        }
    }
}

```

```

        $registration_redirect
= ! empty(
$_REQUEST['redirect_to'] ) ?
$_REQUEST['redirect_to'] : "";

/**
 * Filters the
registration redirect URL.
 *
 * @since 3.0.0
 * @since 5.9.0
Added the `$errors` parameter.
 *
 * @param string
$registration_redirect The redirect
destination URL.
 * @param
int|WP_Error $errors User
id if registration was successful,
 *
WP_Error object otherwise.
 */

$redirect_to =
apply_filters( 'registration_redirect',
$registration_redirect, $errors );

login_header( __(
'Registration Form' ), '<p
class="message register">' . __(
'Register For This Site' ) . '</p>',
$errors );

        ?>
        <form
name="registerform"
id="registerform" action="<?php
echo esc_url( site_url( 'wp-
login.php?action=register',
'login_post' ) ); ?>" method="post"
novalidate="novalidate">
        <p>
                <label
for="user_login"><?php _e(
'Username' ); ?></label>
                <input
type="text" name="user_login"
id="user_login" class="input"
value="<?php echo esc_attr(
wp_unslash( $user_login ) ); ?>"
size="20" autocapitalize="off"
autocomplete="username" />
        </p>
        <p>
                <label
for="user_email"><?php _e( 'Email'
); ?></label>
                <input
type="email" name="user_email"
id="user_email" class="input"
value="<?php echo esc_attr(
wp_unslash( $user_email ) ); ?>"
size="25" autocomplete="email" />
        </p>
        <?php
/**

```

	* Fires	</form>
following the 'Email' field in the user registration form.		
	*	<p id="nav">
	* @since	<a
2.1.0		href="<?php echo esc_url(wp_login_url()); ?>"><?php _e('Log in'); ?>
	*/	
	do_action(<?php
'register_form');		echo esc_html(\$login_link_separator); ?>
	?>	<a
	<p	href="<?php echo esc_url(wp_lostpassword_url()); ?>"><?php _e('Lost your password?'); ?>
id="reg_passmail">		
	<?php	</p>
_e('Registration confirmation will be emailed to you.'); ?>		<?php
	</p>	
	<br	login_footer('user_login');
class="clear" />		break;
	<input	
type="hidden" name="redirect_to" value="<?php echo esc_attr(\$redirect_to); ?>" />		case 'checkemail':
	<p	\$redirect_to =
class="submit">		admin_url();
	<input	\$errors = new
type="submit" name="wp-submit" id="wp-submit" class="button button-primary button-large" value="<?php esc_attr_e('Register'); ?>" />		WP_Error();
		if ('confirm' === \$_GET['checkemail']) {
	</p>	\$errors->add(

```

        'confirm',
                                sprintf(
                                );
                                }

        /* translators: %s: Link to the
login page. */

        __( 'Check your email for the
confirmation link, then visit the <a
href="%s">login page</a>.' ),

        wp_login_url()
                                ),

        'message'
                                );

        } elseif ( 'registered'
=== $_GET['checkemail'] ) {

        $errors->add(

        'registered',

        sprintf(

        /* translators: %s: Link to the
login page. */

        __( 'Registration complete.
Please check your email, then visit
the <a href="%s">login page</a>.' ),

        wp_login_url()

        ),

        'message'
                                );

        /** This action is
documented in wp-login.php */

        $errors =
apply_filters( 'wp_login_errors',
$errors, $redirect_to );

        login_header( __(
'Check your email' ), ", $errors );

        login_footer();

        break;

        case 'confirmation':

        if ( ! isset(
$_GET['request_id'] ) ) {

        wp_die( __(
'Missing request ID.' ) );

        }

        if ( ! isset(
$_GET['confirm_key'] ) ) {

        wp_die( __(
'Missing confirm key.' ) );

        }

```

```

        $request_id = (int)
$_GET['request_id'];

        $key =
sanitize_text_field( wp_unslash(
$_GET['confirm_key'] ) );

        $result =
wp_validate_user_request_key(
$request_id, $key );

        if ( is_wp_error(
$result ) ) {

                wp_die(

                        $message =
_wp_privacy_account_request_confirmed_message( $request_id );

                        login_header( __(
'User action confirmed.' ), $message
);

                        login_footer();

                        exit;

                        case 'login':

                        default:

                                $secure_cookie = "";

                                $customize_login =
isset( $_REQUEST['customize-
login'] );

                                if ( $customize_login
) {

```

*

* @since 4.9.6

*

* @param int
 \$request_id Request ID.

*/

do_action(
 'user_request_action_confirmed',
 \$request_id);

\$message =
 _wp_privacy_account_request_confirmed_message(\$request_id);

login_header(__(
 'User action confirmed.'), \$message
);

login_footer();

exit;

case 'login':

default:

\$secure_cookie = "";

\$customize_login =
 isset(\$_REQUEST['customize-
 login']);

if (\$customize_login
) {

* Fires an action
 hook when the account action has
 been confirmed by the user.

*

* Using this you can
 assume the user has agreed to
 perform the action by

* clicking on the link
 in the confirmation email.

*

* After firing this
 action hook the page will redirect to
 wp-login a callback

* redirects or exits
 first.

```

    }
    wp_enqueue_script(
'customize-base' );
    }
}

// If the user wants
SSL but the session is not SSL, force
a secure cookie.

if ( ! empty(
$_POST['log'] ) && !
force_ssl_admin() ) {

    $user_name =
sanitize_user( wp_unslash(
$_POST['log'] ) );

    $user =
get_user_by( 'login', $user_name );

    if ( ! $user &&
strpos( $user_name, '@' ) ) {

        $user =
get_user_by( 'email', $user_name );

    }

    if ( $user ) {

        if (
get_user_option( 'use_ssl', $user->ID
) ) {

            $secure_cookie = true;

            force_ssl_admin( true );

        }

        if ( isset(
$_REQUEST['redirect_to'] ) ) {

            $redirect_to =
$_REQUEST['redirect_to'];

            // Redirect to
HTTPS if user wants SSL.

            if (
$secure_cookie && false !== strpos(
$redirect_to, 'wp-admin' ) ) {

                $redirect_to = preg_replace(
'^http://', 'https://', $redirect_to );

            }

        } else {

            $redirect_to =
admin_url();

        }

        $reauth = empty(
$_REQUEST['reauth'] ) ? false : true;

        $user = wp_signon(
array(), $secure_cookie );

```



```

        if ( empty(
$_COOKIE[
LOGGED_IN_COOKIE ] ) ) {
            if (
headers_sent() ) {
                $user =
new WP_Error(
                    'test_cookie',
                    sprintf(
                        /* translators: 1: Browser
cookie documentation URL, 2:
Support forums URL. */
                        __( '<strong>Error</strong>:
Cookies are blocked due to
unexpected output. For help, please
see <a href="%1$s">this
documentation</a> or try the <a
href="%2$s">support forums</a>.' ),
                        __(
'https://wordpress.org/support/article/
cookies/' ),
                        __(
'https://wordpress.org/support/forums
/' )
                    )
                );
            }
        } elseif ( isset(
$_POST['testcookie'] ) && empty(
$_COOKIE[ TEST_COOKIE ] ) ) {
            // If
cookies are disabled, we can't log in
even with a valid user and password.
            $user =
new WP_Error(
                'test_cookie',
                sprintf(
                    /* translators: %s: Browser
cookie documentation URL. */
                    __( '<strong>Error</strong>:
Cookies are blocked or not supported
by your browser. You must <a
href="%s">enable cookies</a> to
use WordPress.' ),
                    __(
'https://wordpress.org/support/article/
cookies/#enable-cookies-in-your-
browser' )
                )
            );
        }
    }
);

```

```

        $requested_redirect_to =
isset( $_REQUEST['redirect_to'] ) ?
$_REQUEST['redirect_to'] : '';

```

```

/**

```

```

    * Filters the login
redirect URL.

```

```

*

```

```

* @since 3.0.0

```

```

*

```

```

    * @param string
$redirect_to      The redirect
destination URL.

```

```

    * @param string
$requested_redirect_to The requested
redirect destination URL passed as a
parameter.

```

```

    * @param
WP_User|WP_Error $user
WP_User object if login was
successful, WP_Error object
otherwise.

```

```

*/

```

```

        $redirect_to =
apply_filters( 'login_redirect',
$redirect_to, $requested_redirect_to,
$user );

```

```

        if ( ! is_wp_error(
$user ) && ! $reauth ) {

```

```

            if (
$interim_login ) {

```

```

        $message = '<p
class="message">' . __( 'You have
logged in successfully.' ) . '</p>';

```

```

        $interim_login = 'success';

```

```

        login_header( "", $message );

```

```

?>

```

```

</div>

```

```

<?php

```

```

/**

```

```

This action is documented in wp-
login.php */

```

```

do_action( 'login_footer' );

```

```

if (

```

```

$customize_login ) {

```

```

?>

```

```

        <script
type="text/javascript">setTimeout(
function(){ new
wp.customize.Messenger({ url:
'<?php echo wp_customize_url();
?>', channel: 'login' }).send('login' )},
1000 );</script>

```

```

<?php

```

```

    }
    /**
    This filter is documented in wp-
    login.php */
?>
    $admin_email_check_interva
l = (int) apply_filters(
'admin_email_check_interval', 6 *
MONTH_IN_SECONDS );

    exit;
    if (
$admin_email_check_interval > 0
&& time() > $admin_email_lifespan
) {

        // Check if it is
time to add a redirect to the admin
email confirmation screen.

        $redirect_to =
add_query_arg(
        array(
            'action' =>
'confirm_admin_email',
            'wp_lang' =>
get_user_locale( $user ),
        ),
        wp_login_url( $redirect_to )
    );
    * to
the admin email confirmation screen.
    */
}
}

```

```

        if ( ( empty(
$redirect_to ) || 'wp-admin/' ===
$redirect_to || admin_url() ===
$redirect_to ) ) {

                                wp_safe_redirect(
$redirect_to );

                                exit;

                                }

                                $errors = $user;

                                // Clear errors if
loggedout is set.

                                if ( ! empty(
$_GET['loggedout'] ) || $reauth ) {

                                $errors = new
WP_Error();

                                }

                                if ( empty( $_POST )
&& $errors->get_error_codes() ===
array( 'empty_username',
'empty_password' ) ) {

                                $errors = new
WP_Error( "", "" );

                                }

                                if ( $interim_login ) {

                                if ( ! $errors-
>has_errors() ) {

                                $errors->add( 'expired', __(
'Your session has expired. Please log

```

```

in to continue where you left off.' ),
'message' );

    }

    } else {

        // Some parts
of this script use the main login form
to display a message.

        if ( isset(
$_GET['loggedout'] ) &&
$_GET['loggedout'] ) {

            $errors->add( 'loggedout', __(
'You are now logged out.' ),
'message' );

        } elseif ( isset(
$_GET['registration'] ) && 'disabled'
=== $_GET['registration'] ) {

            $errors->add(
'registerdisabled', __(
'<strong>Error</strong>: User
registration is currently not allowed.'
) );

        } elseif (
strpos( $redirect_to,
'about.php?updated' ) ) {

            $errors->add( 'updated', __(
'<strong>You have successfully
updated WordPress!</strong> Please
log back in to see what&#8217;s
new.' ), 'message' );

        } elseif (
WP_Recovery_Mode_Link_Service:

```

```

:LOGIN_ACTION_ENTERED ===
$action ) {

    $errors->add(
'enter_recovery_mode', __(
'Recovery Mode Initialized. Please
log in to continue.' ), 'message' );

    } elseif ( isset(
$_GET['redirect_to'] ) && false !==
strpos( $_GET['redirect_to'], 'wp-
admin/authorize-application.php' ) )
    {

        $query_component =
wp_parse_url( $_GET['redirect_to'],
PHP_URL_QUERY );

        $query
= array();

        if (
$query_component ) {

            parse_str(
$query_component, $query );

        }

        if ( !
empty( $query['app_name'] ) ) {

            /* translators: 1: Website
name, 2: Application name. */

            $message = sprintf( 'Please
log in to %1$s to authorize %2$s to
connect to your account.',
get_bloginfo( 'name', 'display' ),

```



```

        $aria_describedby_error = '
aria-describedby="login_error";

        } else {

        $aria_describedby_error = "";

        }

        wp_enqueue_script(
'user-profile' );

        ?>

        <form
name="loginform" id="loginform"
action="<?php echo esc_url(
site_url( 'wp-login.php', 'login_post' )
); ?>" method="post">

        <p>

                <label

for="user_login"><?php _e(
'Username or Email Address' );
?></label>

                <input

type="text" name="log"
id="user_login"<?php echo
$aria_describedby_error; ?>
class="input" value="<?php echo
esc_attr( $user_login ); ?>"
size="20" autocapitalize="off"
autocomplete="username" />

        </p>

```

```

        <div
class="user-pass-wrap">

                <label

for="user_pass"><?php _e(
'Password' ); ?></label>

                <div
class="wp-pwd">

                        <input type="password"
name="pwd" id="user_pass"<?php
echo $aria_describedby_error; ?>
class="input password-input"
value="" size="20"
autocomplete="current-password" />

                        <button type="button"
class="button button-secondary wp-
hide-pw hide-if-no-js" data-
toggle="0" aria-label="<?php
esc_attr_e( 'Show password' ); ?>">

                                <span class="dashicons
dashicons-visibility" aria-
hidden="true"></span>

                                </button>

                </div>

        </div>

        <?php

        /**

```

* Fires following the 'Password' field in the login form.

```

2.1.1.0
*
* @since
*/
do_action(
'login_form' );
?>
?>
<p
class="forgetmenot"><input
name="rememberme"
type="checkbox" id="rememberme"
value="forever" <?php checked(
$rememberme ); ?> /> <label
for="rememberme"><?php
esc_html_e( 'Remember Me' );
?></label></p>
<p
class="submit">
<input
type="submit" name="wp-submit"
id="wp-submit" class="button
button-primary button-large"
value="<?php esc_attr_e( 'Log In' );
?>" />
<?php
if (
$interim_login ) {
?>
?>
<input
type="hidden" name="testcookie"
value="1" />
<input type="hidden"
name="interim-login" value="1" />
<?php
} else {
?>
<input type="hidden"
name="redirect_to" value="<?php
echo esc_attr( $redirect_to ); ?>" />
<?php
}
if (
$customize_login ) {
?>
<input type="hidden"
name="customize-login" value="1"
/>
<?php
}
}
?>
<input
type="hidden" name="testcookie"
value="1" />

```



```

        </p>
    </form>
    <?php
        if ( ! $interim_login )
        {
            <p id="nav">
                <?php
                    $login_script =
                    'function wp_attempt_focus() {';
                    $login_script .=
                    'setTimeout( function() {';
                    $login_script .= 'try
                    {';
                    if ( $user_login ) {
                        $login_script
                        .= 'd = document.getElementById(
                        "user_pass" ); d.value = ""';
                    } else {
                        $login_script
                        .= 'd = document.getElementById(
                        "user_login" );';
                    }
                    if ( $errors-
                    >get_error_code() ===
                    'invalid_username' ) {
                }
            }
        }
    }

```

```

}
$login_script .= 'd.value =
"";';
}
}
$login_script .=
'd.focus(); d.select();';
$login_script .= '}'
catch( er ) { }';
$login_script .= '}',
200);';
$login_script .= "}\n";
// End of wp_attempt_focus().

/**
 * Filters whether to
print the call to `wp_attempt_focus()`
on the login screen.
 *
 * @since 4.8.0
 *
 * @param bool
$print Whether to print the function
call. Default true.
 */
if ( apply_filters(
'enable_login_autofocus', true ) && !
$error ) {
$login_script
.= "wp_attempt_focus();\n";
}
}
// Run `wpOnload()`
if defined.
$login_script .= "if (
typeof wpOnload === 'function' ) {
wpOnload() }";
?>
<script
type="text/javascript">
<?php echo
$login_script; ?>
</script>
<?php
if ( $interim_login ) {
?>
<script
type="text/javascript">
( function() {
try {
var i, links =
document.getElementsByTagName(
'a' );
for ( i in links ) {
if ( links[i].href ) {

```

```

links[i].target =
'_blank';

links[i].rel =
'noopener';

    }

    }

        }
catch( er ) {

        }());

</script>

<?php

    }

        login_footer();

        break;

} // End action switch.

<?php

/**

 * Used to set up and fix common
variables and include

 * the WordPress procedural and
class library.

 *

 * Allows for some configuration in
wp-config.php (see default-
constants.php)

 *

 * @package WordPress

 */

/**

 * Stores the location of the
WordPress directory of functions,
classes, and core content.

 *

 * @since 1.0.0

 */

define( 'WPINC', 'wp-includes' );

/**

 * Version information for the
current WordPress release.

 *

 * These can't be directly globalized
in version.php. When updating,

 * we're including version.php from
another installation and don't want

 * these values to be overridden if
already set.

 *

 * @global string $wp_version
The WordPress version string.

 * @global int $wp_db_version
WordPress database version.

```

```

* @global string $tinymce_version
TinyMCE version.

* @global string
$required_php_version The
required PHP version string.

* @global string
$required_mysql_version The
required MySQL version string.

* @global string
$wp_local_package Locale code
of the package.

*/

global $wp_version,
$wp_db_version, $tinymce_version,
$required_php_version,
$required_mysql_version,
$wp_local_package;

require ABSPATH . WPINC .
'/version.php';

require ABSPATH . WPINC .
'/load.php';

// Check for the required PHP
version and for the MySQL
extension or a database drop-in.

wp_check_php_mysql_versions();

// Include files required for
initialization.

require ABSPATH . WPINC .
'/class-wp-paused-extensions-
storage.php';

require ABSPATH . WPINC .
'/class-wp-fatal-error-handler.php';

require ABSPATH . WPINC .
'/class-wp-recovery-mode-cookie-
service.php';

require ABSPATH . WPINC .
'/class-wp-recovery-mode-key-
service.php';

require ABSPATH . WPINC .
'/class-wp-recovery-mode-link-
service.php';

require ABSPATH . WPINC .
'/class-wp-recovery-mode-email-
service.php';

require ABSPATH . WPINC .
'/class-wp-recovery-mode.php';

require ABSPATH . WPINC .
'/error-protection.php';

require ABSPATH . WPINC .
'/default-constants.php';

require_once ABSPATH . WPINC .
'/plugin.php';

/**
 * If not already configured,
 * ` $blog_id ` will default to 1 in a
 * single site
 *
 * configuration. In multisite, it will
 * be overridden by default in ms-
 * settings.php.
 *
 * @global int $blog_id

```

```

* @since 2.0.0
*/

global $blog_id;

// Set initial default constants
including WP_MEMORY_LIMIT,
WP_MAX_MEMORY_LIMIT,
WP_DEBUG, SCRIPT_DEBUG,
WP_CONTENT_DIR and
WP_CACHE.

wp_initial_constants();

// Make sure we register the
shutdown handler for fatal errors as
soon as possible.

wp_register_fatal_error_handler();

// WordPress calculates offsets from
UTC.

// phpcs:ignore
WordPress.DateTime.RestrictedFunc
tions.timezone_change_date_default
_timezone_set

date_default_timezone_set( 'UTC' );

// Standardize $_SERVER variables
across setups.

wp_fix_server_vars();

// Check if we're in maintenance
mode.

wp_maintenance();

// Start loading timer.

timer_start();

// Check if we're in WP_DEBUG
mode.

wp_debug_mode();

/**
 * Filters whether to enable loading
of the advanced-cache.php drop-in.
 *
 * This filter runs before it can be
used by plugins. It is designed for
non-web
 * run-times. If false is returned,
advanced-cache.php will never be
loaded.
 *
 * @since 4.6.0
 *
 * @param bool
$enable_advanced_cache Whether to
enable loading advanced-cache.php
(if present).
 *
 * Default true.
 */
if ( WP_CACHE && apply_filters(
'enable_loading_advanced_cache_dr

```

```

opin', true ) && file_exists(
WP_CONTENT_DIR . '/advanced-
cache.php' ) ) {

    // For an advanced caching
plugin to use. Uses a static drop-in
because you would only want one.

    include
WP_CONTENT_DIR . '/advanced-
cache.php';

    // Re-initialize any hooks
added manually by advanced-
cache.php.

    if ( $wp_filter ) {

        $wp_filter =
WP_Hook::build_preinitialized_hoo
ks( $wp_filter );

    }

}

// Define WP_LANG_DIR if not set.
wp_set_lang_dir();

// Load early WordPress files.
require ABSPATH . WPINC .
'/compat.php';

require ABSPATH . WPINC .
'/class-wp-list-util.php';

require ABSPATH . WPINC .
'/formatting.php';

require ABSPATH . WPINC .
'/meta.php';

require ABSPATH . WPINC .
'/functions.php';

require ABSPATH . WPINC .
'/class-wp-meta-query.php';

require ABSPATH . WPINC .
'/class-wp-matchesmapregex.php';

require ABSPATH . WPINC .
'/class-wp.php';

require ABSPATH . WPINC .
'/class-wp-error.php';

require ABSPATH . WPINC .
'/pomo/mo.php';

/**
 * @global wpdb $wpdb WordPress
database abstraction object.
 * @since 0.71
 */
global $wpdb;

// Include the wpdb class and, if
present, a db.php database drop-in.
require_wp_db();

// Set the database table prefix and
the format specifiers for database
table columns.
$GLOBALS['table_prefix'] =
$table_prefix;

```

```

wp_set_wpdb_vars();

// Start the WordPress object cache,
or an external object cache if the
drop-in is present.

wp_start_object_cache();

// Attach the default filters.

require ABSPATH . WPINC .
'default-filters.php';

// Initialize multisite if enabled.

if ( is_multisite() ) {

    require ABSPATH . WPINC
    . '/class-wp-site-query.php';

    require ABSPATH . WPINC
    . '/class-wp-network-query.php';

    require ABSPATH . WPINC
    . '/ms-blogs.php';

    require ABSPATH . WPINC
    . '/ms-settings.php';
} elseif ( ! defined( 'MULTISITE' ) )
{
    define( 'MULTISITE', false );
}

register_shutdown_function(
'shutdown_action_hook' );

// Stop most of WordPress from
being loaded if we just want the
basics.

if ( SHORTINIT ) {
    return false;
}

// Load the L10n library.

require_once ABSPATH . WPINC .
'/l10n.php';

require_once ABSPATH . WPINC .
'/class-wp-locale.php';

require_once ABSPATH . WPINC .
'/class-wp-locale-switcher.php';

// Run the installer if WordPress is
not installed.

wp_not_installed();

// Load most of WordPress.

require ABSPATH . WPINC .
'/class-wp-walker.php';

require ABSPATH . WPINC .
'/class-wp-ajax-response.php';

require ABSPATH . WPINC .
'/capabilities.php';

require ABSPATH . WPINC .
'/class-wp-roles.php';

require ABSPATH . WPINC .
'/class-wp-role.php';

```

```
require ABSPATH . WPINC .
/class-wp-user.php';

require ABSPATH . WPINC .
/class-wp-query.php';

require ABSPATH . WPINC .
/query.php';

require ABSPATH . WPINC .
/class-wp-date-query.php';

require ABSPATH . WPINC .
/theme.php';

require ABSPATH . WPINC .
/class-wp-theme.php';

require ABSPATH . WPINC .
/class-wp-theme-json-schema.php';

require ABSPATH . WPINC .
/class-wp-theme-json.php';

require ABSPATH . WPINC .
/class-wp-theme-json-resolver.php';

require ABSPATH . WPINC .
/global-styles-and-settings.php';

require ABSPATH . WPINC .
/class-wp-block-template.php';

require ABSPATH . WPINC .
/block-template-utils.php';

require ABSPATH . WPINC .
/block-template.php';

require ABSPATH . WPINC .
/theme-templates.php';

require ABSPATH . WPINC .
/template.php';

require ABSPATH . WPINC .
/https-detection.php';

require ABSPATH . WPINC .
/https-migration.php';

require ABSPATH . WPINC .
/class-wp-user-request.php';

require ABSPATH . WPINC .
/user.php';

require ABSPATH . WPINC .
/class-wp-user-query.php';

require ABSPATH . WPINC .
/class-wp-session-tokens.php';

require ABSPATH . WPINC .
/class-wp-user-meta-session-
tokens.php';

require ABSPATH . WPINC .
/class-wp-metadata-lazyloader.php';

require ABSPATH . WPINC .
/general-template.php';

require ABSPATH . WPINC . /link-
template.php';

require ABSPATH . WPINC .
/author-template.php';

require ABSPATH . WPINC .
/robots-template.php';

require ABSPATH . WPINC .
/post.php';

require ABSPATH . WPINC .
/class-walker-page.php';

require ABSPATH . WPINC .
/class-walker-page-dropdown.php';

require ABSPATH . WPINC .
/class-wp-post-type.php';
```


require ABSPATH . WPINC .
'/class-wp-post.php';

require ABSPATH . WPINC . '/post-
template.php';

require ABSPATH . WPINC .
'/revision.php';

require ABSPATH . WPINC . '/post-
formats.php';

require ABSPATH . WPINC . '/post-
thumbnail-template.php';

require ABSPATH . WPINC .
'/category.php';

require ABSPATH . WPINC .
'/class-walker-category.php';

require ABSPATH . WPINC .
'/class-walker-category-
dropdown.php';

require ABSPATH . WPINC .
'/category-template.php';

require ABSPATH . WPINC .
'/comment.php';

require ABSPATH . WPINC .
'/class-wp-comment.php';

require ABSPATH . WPINC .
'/class-wp-comment-query.php';

require ABSPATH . WPINC .
'/class-walker-comment.php';

require ABSPATH . WPINC .
'/comment-template.php';

require ABSPATH . WPINC .
'/rewrite.php';

require ABSPATH . WPINC .
'/class-wp-rewrite.php';

require ABSPATH . WPINC .
'/feed.php';

require ABSPATH . WPINC .
'/bookmark.php';

require ABSPATH . WPINC .
'/bookmark-template.php';

require ABSPATH . WPINC .
'/kses.php';

require ABSPATH . WPINC .
'/cron.php';

require ABSPATH . WPINC .
'/deprecated.php';

require ABSPATH . WPINC .
'/script-loader.php';

require ABSPATH . WPINC .
'/taxonomy.php';

require ABSPATH . WPINC .
'/class-wp-taxonomy.php';

require ABSPATH . WPINC .
'/class-wp-term.php';

require ABSPATH . WPINC .
'/class-wp-term-query.php';

require ABSPATH . WPINC .
'/class-wp-tax-query.php';

require ABSPATH . WPINC .
'/update.php';

require ABSPATH . WPINC .
'/canonical.php';

require ABSPATH . WPINC .
'/shortcodes.php';

```
require ABSPATH . WPINC .
'/embed.php';

require ABSPATH . WPINC .
'/class-wp-embed.php';

require ABSPATH . WPINC .
'/class-wp-oembed.php';

require ABSPATH . WPINC .
'/class-wp-oembed-controller.php';

require ABSPATH . WPINC .
'/media.php';

require ABSPATH . WPINC .
'/http.php';

require ABSPATH . WPINC .
'/class-wp-http.php';

require ABSPATH . WPINC .
'/class-wp-http-streams.php';

require ABSPATH . WPINC .
'/class-wp-http-curl.php';

require ABSPATH . WPINC .
'/class-wp-http-proxy.php';

require ABSPATH . WPINC .
'/class-wp-http-cookie.php';

require ABSPATH . WPINC .
'/class-wp-http-encoding.php';

require ABSPATH . WPINC .
'/class-wp-http-response.php';

require ABSPATH . WPINC .
'/class-wp-http-requests-
response.php';

require ABSPATH . WPINC .
'/class-wp-http-requests-hooks.php';
```

```
require ABSPATH . WPINC .
'/widgets.php';

require ABSPATH . WPINC .
'/class-wp-widget.php';

require ABSPATH . WPINC .
'/class-wp-widget-factory.php';

require ABSPATH . WPINC . '/nav-
menu-template.php';

require ABSPATH . WPINC . '/nav-
menu.php';

require ABSPATH . WPINC .
'/admin-bar.php';

require ABSPATH . WPINC .
'/class-wp-application-
passwords.php';

require ABSPATH . WPINC . '/rest-
api.php';

require ABSPATH . WPINC . '/rest-
api/class-wp-rest-server.php';

require ABSPATH . WPINC . '/rest-
api/class-wp-rest-response.php';

require ABSPATH . WPINC . '/rest-
api/class-wp-rest-request.php';

require ABSPATH . WPINC . '/rest-
api/endpoints/class-wp-rest-
controller.php';

require ABSPATH . WPINC . '/rest-
api/endpoints/class-wp-rest-posts-
controller.php';

require ABSPATH . WPINC . '/rest-
api/endpoints/class-wp-rest-
attachments-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-global-styles-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-post-types-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-post-statuses-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-revisions-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-autosaves-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-taxonomies-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-terms-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-menu-items-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-menus-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-menu-locations-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-users-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-comments-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-search-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-blocks-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-block-types-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-block-renderer-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-settings-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-themes-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-plugins-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-block-directory-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-edit-site-export-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-pattern-directory-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-block-patterns-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-block-pattern-categories-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-application-passwords-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-site-health-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-sidebars-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-widget-types-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-widgets-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-templates-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/endpoints/class-wp-rest-url-details-controller.php';
```

```
require ABSPATH . WPINC . '/rest-api/fields/class-wp-rest-meta-fields.php';
```

```
require ABSPATH . WPINC . '/rest-api/fields/class-wp-rest-comment-meta-fields.php';
```

```
require ABSPATH . WPINC . '/rest-api/fields/class-wp-rest-post-meta-fields.php';
```

```
require ABSPATH . WPINC . '/rest-api/fields/class-wp-rest-term-meta-fields.php';
```

```
require ABSPATH . WPINC . '/rest-api/fields/class-wp-rest-user-meta-fields.php';
```

```
require ABSPATH . WPINC . '/rest-api/search/class-wp-rest-search-handler.php';
```

```
require ABSPATH . WPINC . '/rest-api/search/class-wp-rest-post-search-handler.php';
```

```
require ABSPATH . WPINC . '/rest-api/search/class-wp-rest-term-search-handler.php';
```

```
require ABSPATH . WPINC . '/rest-api/search/class-wp-rest-post-format-search-handler.php';
```

```
require ABSPATH . WPINC . '/sitemaps.php';
```

```
require ABSPATH . WPINC . '/sitemaps/class-wp-sitemaps.php';
```

```
require ABSPATH . WPINC . '/sitemaps/class-wp-sitemaps-index.php';
```

```
require ABSPATH . WPINC . '/sitemaps/class-wp-sitemaps-provider.php';
```

```
require ABSPATH . WPINC . '/sitemaps/class-wp-sitemaps-registry.php';
```

```
require ABSPATH . WPINC .
'/sitemaps/class-wp-sitemaps-
renderer.php';

require ABSPATH . WPINC .
'/sitemaps/class-wp-sitemaps-
stylesheet.php';

require ABSPATH . WPINC .
'/sitemaps/providers/class-wp-
sitemaps-posts.php';

require ABSPATH . WPINC .
'/sitemaps/providers/class-wp-
sitemaps-taxonomies.php';

require ABSPATH . WPINC .
'/sitemaps/providers/class-wp-
sitemaps-users.php';

require ABSPATH . WPINC .
'/class-wp-block-editor-context.php';

require ABSPATH . WPINC .
'/class-wp-block-type.php';

require ABSPATH . WPINC .
'/class-wp-block-pattern-categories-
registry.php';

require ABSPATH . WPINC .
'/class-wp-block-patterns-
registry.php';

require ABSPATH . WPINC .
'/class-wp-block-styles-registry.php';

require ABSPATH . WPINC .
'/class-wp-block-type-registry.php';

require ABSPATH . WPINC .
'/class-wp-block.php';

require ABSPATH . WPINC .
'/class-wp-block-list.php';
```

```
require ABSPATH . WPINC .
'/class-wp-block-parser.php';

require ABSPATH . WPINC .
'/blocks.php';

require ABSPATH . WPINC .
'/blocks/index.php';

require ABSPATH . WPINC .
'/block-editor.php';

require ABSPATH . WPINC .
'/block-patterns.php';

require ABSPATH . WPINC .
'/class-wp-block-supports.php';

require ABSPATH . WPINC .
'/block-supports/utils.php';

require ABSPATH . WPINC .
'/block-supports/align.php';

require ABSPATH . WPINC .
'/block-supports/border.php';

require ABSPATH . WPINC .
'/block-supports/colors.php';

require ABSPATH . WPINC .
'/block-supports/custom-
classname.php';

require ABSPATH . WPINC .
'/block-supports/dimensions.php';

require ABSPATH . WPINC .
'/block-supports/duotone.php';

require ABSPATH . WPINC .
'/block-supports/elements.php';

require ABSPATH . WPINC .
'/block-supports/generated-
classname.php';
```

```

require ABSPATH . WPINC .
'/block-supports/layout.php';

require ABSPATH . WPINC .
'/block-supports/spacing.php';

require ABSPATH . WPINC .
'/block-supports/typography.php';

$GLOBALS['wp_embed'] = new
WP_Embed();

// Load multisite-specific files.

if ( is_multisite() ) {

    require ABSPATH . WPINC
    . '/ms-functions.php';

    require ABSPATH . WPINC
    . '/ms-default-filters.php';

    require ABSPATH . WPINC
    . '/ms-deprecated.php';
}

// Define constants that rely on the
API to obtain the default value.

// Define must-use plugin directory
constants, which may be overridden
in the sunrise.php drop-in.

wp_plugin_directory_constants();

$GLOBALS['wp_plugin_paths'] =
array();

// Load must-use plugins.

foreach ( wp_get_mu_plugins() as
$mu_plugin ) {

    $_wp_plugin_file =
$mu_plugin;

    include_once $mu_plugin;

    $mu_plugin =
$_wp_plugin_file; // Avoid stomping
of the $mu_plugin variable in a
plugin.

/**
 * Fires once a single must-
use plugin has loaded.
 *
 * @since 5.1.0
 *
 * @param string $mu_plugin
Full path to the plugin's main file.
 */

    do_action(
'mu_plugin_loaded', $mu_plugin );
}

unset( $mu_plugin,
$_wp_plugin_file );

// Load network activated plugins.

if ( is_multisite() ) {

```

```

        foreach (
wp_get_active_network_plugins() as
$network_plugin ) {

        wp_register_plugin_realpath(
$network_plugin );

        $_wp_plugin_file =
$network_plugin;

        include_once
$network_plugin;

        $network_plugin =
$_wp_plugin_file; // Avoid stomping
of the $network_plugin variable in a
plugin.

        /**
        * Fires once a single
network-activated plugin has loaded.
        *
        * @since 5.1.0
        *
        * @param string
$network_plugin Full path to the
plugin's main file.
        */

        do_action(
'network_plugin_loaded',
$network_plugin );
    }

        unset( $network_plugin,
$_wp_plugin_file );
    }

    /**
    * Fires once all must-use and
network-activated plugins have
loaded.
    *
    * @since 2.8.0
    */
do_action( 'muplugins_loaded' );

if ( is_multisite() ) {
    ms_cookie_constants();
}

// Define constants after multisite is
loaded.

wp_cookie_constants();

// Define and enforce our SSL
constants.

wp_ssl_constants();

// Create common globals.

require ABSPATH . WPINC .
'vars.php';

```

```

// Make taxonomies and posts
available to plugins and themes.

// @plugin authors: warning: these
get registered again on the init hook.
create_initial_taxonomies();

create_initial_post_types();

wp_start_scraping_edited_file_errors
();

// Register the default theme
directory root.

register_theme_directory(
get_theme_root() );

if ( ! is_multisite() ) {
    // Handle users requesting a
recovery mode link and initiating
recovery mode.

    wp_recovery_mode()-
>initialize();
}

// Load active plugins.

foreach (
wp_get_active_and_valid_plugins()
as $plugin ) {

    wp_register_plugin_realpath(
$plugin );

    $_wp_plugin_file = $plugin;

    include_once $plugin;

    $plugin = $_wp_plugin_file;
// Avoid stomping of the $plugin
variable in a plugin.

/**
 * Fires once a single
activated plugin has loaded.
 *
 * @since 5.1.0
 *
 * @param string $plugin
Full path to the plugin's main file.
 */
do_action( 'plugin_loaded',
$plugin );
}

unset( $plugin, $_wp_plugin_file );

// Load pluggable functions.

require ABSPATH . WPINC .
'pluggable.php';

require ABSPATH . WPINC .
'pluggable-deprecated.php';

// Set internal encoding.

```



```

wp_set_internal_encoding();

// Run wp_cache_postload() if object
cache is enabled and the function
exists.

if ( WP_CACHE &&
function_exists( 'wp_cache_postload'
) ) {

    wp_cache_postload();

}

/**

* Fires once activated plugins have
loaded.

*

* Pluggable functions are also
available at this point in the loading
order.

*

* @since 1.5.0

*/

do_action( 'plugins_loaded' );

// Define constants which affect
functionality if not already defined.

wp_functionality_constants();

// Add magic quotes and set up
$_REQUEST ( $_GET + $_POST ).

```

```

wp_magic_quotes();

/**

* Fires when comment cookies are
sanitized.

*

* @since 2.0.11

*/

do_action(
'sanitize_comment_cookies' );

/**

* WordPress Query object

*

* @global WP_Query
$wp_the_query WordPress Query
object.

* @since 2.0.0

*/

$GLOBALS['wp_the_query'] = new
WP_Query();

/**

* Holds the reference to @see
$wp_the_query

* Use this global for WordPress
queries

*

```

```
* @global WP_Query $wp_query
WordPress Query object.
```

```
* @since 1.5.0
```

```
*/
```

```
$GLOBALS['wp_query'] =
$GLOBALS['wp_the_query'];
```

```
/**
```

```
* Holds the WordPress Rewrite
object for creating pretty URLs
```

```
*
```

```
* @global WP_Rewrite
$wp_rewrite WordPress rewrite
component.
```

```
* @since 1.5.0
```

```
*/
```

```
$GLOBALS['wp_rewrite'] = new
WP_Rewrite();
```

```
/**
```

```
* WordPress Object
```

```
*
```

```
* @global WP $wp Current
WordPress environment instance.
```

```
* @since 2.0.0
```

```
*/
```

```
$GLOBALS['wp'] = new WP();
```

```
/**
```

```
* WordPress Widget Factory Object
```

```
*
```

```
* @global WP_Widget_Factory
$wp_widget_factory
```

```
* @since 2.8.0
```

```
*/
```

```
$GLOBALS['wp_widget_factory'] =
new WP_Widget_Factory();
```

```
/**
```

```
* WordPress User Roles
```

```
*
```

```
* @global WP_Roles $wp_roles
WordPress role management object.
```

```
* @since 2.0.0
```

```
*/
```

```
$GLOBALS['wp_roles'] = new
WP_Roles();
```

```
/**
```

```
* Fires before the theme is loaded.
```

```
*
```

```
* @since 2.6.0
```

```
*/
```

```
do_action( 'setup_theme' );
```

```

// Define the template related
constants.

wp_templating_constants();

// Load the default text localization
domain.

load_default_textdomain();

$locale = get_locale();

$locale_file = WP_LANG_DIR .
"$locale.php";

if ( ( 0 === validate_file( $locale ) )
&& is_readable( $locale_file ) ) {
    require $locale_file;
}

unset( $locale_file );

/**
 * WordPress Locale object for
loading locale domain date and
various strings.
 *
 * @global WP_Locale $wp_locale
WordPress date and time locale
object.
 * @since 2.1.0
 */
$GLOBALS['wp_locale'] = new
WP_Locale();

/**
 * WordPress Locale Switcher object
for switching locales.
 *
 * @since 4.7.0
 *
 * @global WP_Locale_Switcher
$wp_locale_switcher WordPress
locale switcher object.
 */
$GLOBALS['wp_locale_switcher'] =
new WP_Locale_Switcher();
$GLOBALS['wp_locale_switcher']->
init();

// Load the functions for the active
theme, for both parent and child
theme if applicable.

foreach (
wp_get_active_and_valid_themes()
as $theme ) {
    if ( file_exists( $theme .
'/functions.php' ) ) {
        include $theme .
'/functions.php';
    }
}

unset( $theme );

```

```

/**
 * Fires after the theme is loaded.
 *
 * @since 3.0.0
 */
do_action( 'after_setup_theme' );

// Create an instance of
WP_Site_Health so that Cron events
may fire.
if ( ! class_exists( 'WP_Site_Health' )
) {
    require_once ABSPATH .
'wp-admin/includes/class-wp-site-
health.php';
}
WP_Site_Health::get_instance();

// Set up current user.
$GLOBALS['wp']->init();

/**
 * Fires after WordPress has finished
loading but before any headers are
sent.
 *
 * Most of WP is loaded at this stage,
and the user is authenticated. WP
continues
 * to load on the {@see 'init'} hook
that follows (e.g. widgets), and many
plugins instantiate
 * themselves on it for all sorts of
reasons (e.g. they need a user, a
taxonomy, etc.).
 *
 * If you wish to plug an action once
WP is loaded, use the {@see
'wp_loaded'} hook below.
 *
 * @since 1.5.0
 */
do_action( 'init' );

// Check site status.
if ( is_multisite() ) {
    $file = ms_site_check();
    if ( true !== $file ) {
        require $file;
        die();
    }
    unset( $file );
}

/**
 * This hook is fired once WP, all
plugins, and the theme are fully
loaded and instantiated.

```

*

* Ajax requests should use wp-admin/admin-ajax.php. admin-ajax.php can handle requests for

* users not logged in.

*

* @link

https://codex.wordpress.org/AJAX_in_Plugins

*

* @since 3.0.0

*/

do_action('wp_loaded');



KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN TEKNIK KOMPUTER
Jln. Srijaya Negara Bukit Besar Palembang 30139
Telp. (0711) 353414, Fax (0711) 355918



SURAT KESEPAKATAN BIMBINGAN LAPORAN AKHIR

Saya yang bertanda tangan dibawah ini :

Pihak Pertama

Nama : Damian Chatur Prasasti
NIM : 061930701611
Jurusan/Prodi : Teknik Komputer/D3 Teknik Komputer

Pihak Kedua

Nama : Ahyar Supani, S.T., M.T.
NIP : 196802111992031002
Jurusan/Prodi : Teknik Komputer

Pada hari ini tanggal Telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir.

Isi kesepakatan :

1. Konsultasi bimbingan sekurang-kurangnya 1(kali) dalam 1(satu) minggu.
2. Pelaksanaan bimbingan pada setiap hari pukul tempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini di buat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Pihak Pertama,

(Damian Chatur Prasasti)
NIM. 061930701611

Palembang, Februari 2022
Pihak Kedua,

(Ahyar Supani, S.T., M.T.)
NIP. 196802111992031002

Mengetahui,

Ketua Jurusan Teknik Komputer,

Azwardi, S.T., M.T.
NIP. 197005232005011004



KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
JURUSAN TEKNIK KOMPUTER
Jln. Srijaya Negara Bukit Besar Palembang 30139
Telp. (0711) 353414, Fax (0711) 355918



SURAT KESEPAKATAN BIMBINGAN LAPORAN AKHIR

Saya yang bertanda tangan dibawah ini :

Pihak Pertama

Nama : Damian Chatur Prasasti
NIM : 061930701611
Jurusan/Prodi : Teknik Komputer/D3 Teknik Komputer

Pihak Kedua

Nama : Indarto, S.T., M.Cs.
NIP : 197307062005011003
Jurusan/Prodi : Teknik Komputer

Pada hari ini tanggal Telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir.

Isi kesepakatan :

1. Konsultasi bimbingan sekurang-kurangnya 1(kali) dalam 1(satu) minggu.
2. Pelaksanaan bimbingan pada setiap hari pukul tempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini di buat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Pihak Pertama,

(Damian Chatur Prasasti)

NIM. 061930701611

Palembang, Februari 2022

Pihak Kedua,

(Indarto, S.T., M.Cs.)

NIP. 197307062005011003

Mengetahui,

Ketua Jurusan Teknik Komputer,

Azwardi, S.T., M.T.

NIP. 197005232005011004



**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA**

Jalan Srijaya Negara Bukit Besar, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918

Website : www.polsri.ac.id E-mail : info@polsri.ac.id



LEMBAR KONSULTASI LAPORAN AKHIR (LA)

Nama Mahasiswa : Damian Chatur Prasasti
NIM : 061930701611
Jurusan/Program Studi : Teknik Komputer / D-III Teknik Komputer
Judul Laporan Akhir : Aplikasi Pengelolaan Penjualan Besi
Bangunan Berbasis Web Pada Tb. Sukses
Abadi Bersama
Dosen Pembimbing 2 : Ahyar Supani, S.T., M.T.

No	Tanggal Konsultasi	Uraian Konsultasi	Paraf Pembimbing
1	24/5/2022	Revisi Judul dan Bab I	
2	31/5/2022	ACC Bab I Revisi Bab II	
3	14/6/2022	ACC Bab II Revisi Bab III Revisi Bab I	
4	24/6/2022	ACC Bab I Revisi Bab III	

5	28/2022 /6	ACC Bab <u>III</u> Revisi Bab <u>IV</u>	✓
6.	5/2022 /7	ACC Bab <u>IV</u> Revisi Bab <u>V</u>	✓
7.	12/2022 /7	ACC Bab <u>IV</u> Revisi Bab <u>V</u> Demo Aplikasi Web	✓
8.	19/2022 /7		✓

Mengetahui,

Ketua Jurusan Teknik Komputer



Azwardi, S.T., M. T

NIP. 197005232005011004



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara Bukit Besar, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918

Website : www.polsri.ac.id E-mail : info@polsri.ac.id



LEMBAR KONSULTASI LAPORAN AKHIR (LA)

Nama Mahasiswa : Damian Chatur Prasasti
NIM : 061930701611
Jurusan/Program Studi : Teknik Komputer / D-III Teknik Komputer
Judul Laporan Akhir : Aplikasi Pengelolaan Penjualan Besi
Bangunan Berbasis Web Pada Tb. Sukses
Abadi Bersama
Dosen Pembimbing 2 : Indarto, S.T., M.Cs.

No	Tanggal Konsultasi	Uraian Konsultasi	Paraf Pembimbing
	27-Juni-22	Bab I, Revisi BAB I	
	28/6 2022	Acc Bab I	
	28/6 2022	ACC Bab II	
	30/6 2022	ACC Bab III	
	30/6 2022	Revisi Web (Hosting web)	

4/7 2022	Revisi Bab <u>IV</u> dan <u>V</u>	↓
5/7 2022	Revisi Bab <u>V</u>	↓
11/7 2022	Acc Bab <u>IV</u> dan <u>V</u>	↓

Mengetahui,

Ketua Jurusan Teknik Komputer

Azwardi, S.T., M. T

NIP. 197005232005011004

	<p>KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI POLITEKNIK NEGERI SRIWIJAYA Jalan Sriwijaya Negara, Palembang 30139 Telp. 0711-353414 fax. 0711-355918 Website : www.polsri.ac.id E-mail : info@polsri.ac.id</p>	 
REKOMENDASI SEMINAR LAPORAN AKHIR (LA)		

Pembimbing Laporan Akhir memberikan rekomendasi kepada,

Nama : Damian Chatur Prasasti
NIM : 061930701611
Jurusan/Program Studi : Teknik Komputer / D3 Teknik Komputer
Judul Laporan : Aplikasi Pengelolaan Penjualan Besi
Bangunan Berbasis Web Pada Tb. Sukses
Abadi Bersama

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Seminar Laporan Akhir pada Tahun Akademik 2021/2022.

Palembang, Juli 2022

Pembimbing I,



Ahyar Supani, S.T., M.T.

NIP.196802111992031002



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI

POLITEKNIK NEGERI SRIWIJAYA

Jalan Srijaya Negara, Palembang 30139

Telp. 0711-353414 fax. 0711-355918

Website : www.polsri.ac.id E-mail : info@polsri.ac.id



REKOMENDASI SEMINAR LAPORAN AKHIR (LA)

Pembimbing Laporan Akhir memberikan rekomendasi kepada,

Nama : Damian Chatur Prasasti
NIM : 061930701611
Jurusan/Program Studi : Teknik Komputer / D3 Teknik Komputer
Judul Laporan : Aplikasi Pengelolaan Penjualan Besi
Bangunan Berbasis Web Pada Tb. Sukses
Abadi Bersama

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Seminar Laporan Akhir pada Tahun Akademik 2021/2022.

Palembang, Juli 2022

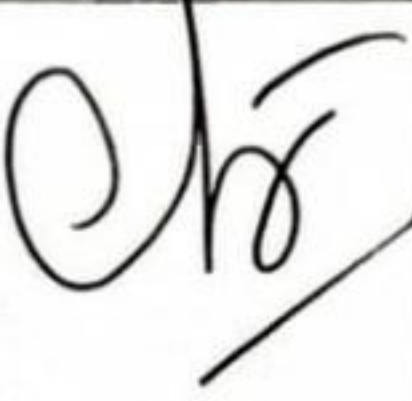
Pembimbing II,

Indarto, S.T., M.Cs.

NIP.197307062005011003

No. Dok. : 	Tgl. Berlaku : KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 Website : www.polsri.ac.id E-mail : info@polsri.ac.id	No. Rev. :  
REVISI TUGAS AKHIR (TA)		

Dosen Penguji : Ema Laila, S.Kom, M.Kom
 Nama Mahasiswa : Damian Chatur Prasasti
 NIM : 061930701611
 Jurusan /Program Studi : Teknik Komputer/D3 Teknik Komputer
 Judul LA/ Skripsi : Aplikasi Pengelolaan Penjualan Besi Bangunan Berbasis Web
 Pada TB. Sukses Abadi Bersama

No	Uraian Revisi	Paraf
1. 2.	Pisahkan akun admin & user. Perbaiki flowchart.	

Palembang, Juli 2022


Dosen Penguji,



Ema Laila, S.Kom, M.Kom
 NIP:197703292001122002

No. Dok. : 	Tgl. Berlaku : KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 Website : www.polsri.ac.id E-mail : info@polsri.ac.id	No. Rev. : 
REVISI TUGAS AKHIR (TA)		

Dosen Penguji : Adi Sutrisman, S.Kom, M.Kom
 Nama Mahasiswa : Damian Chatur Prasasti
 NIM : 061930701611
 Jurusan /Program Studi : Teknik Komputer/D3 Teknik Komputer
 Judul LA/ Skripsi : Aplikasi Pengelolaan Penjualan Besi Bangunan Berbasis Web
 Pada TB. Sukses Abadi Bersama

No	Uraian Revisi	Paraf
-	Bas III Tambahkan Rancangan Database	

Palembang, 28 Juli 2022
 Dosen Penguji,



(Adi Sutrisman, S.Kom, M.Kom)
 NIP:197503052001121005

No. Dok. :	Tgl. Berlaku :	No. Rev. :
	KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 Website : www.polsri.ac.id E-mail : info@polsri.ac.id	 
REVISI TUGAS AKHIR (TA)		

Dosen Penguji : Ikhtison Mekongga, ST., M.Kom
 Nama Mahasiswa : Damian Chatur Prasasti
 NIM : 061930701611
 Jurusan /Program Studi : Teknik Komputer/D3 Teknik Komputer
 Judul LA/ Skripsi : Aplikasi Pengelolaan Penjualan Besi Bangunan Berbasis Web
 : Pada TB. Sukses Abadi Bersama

No	Uraian Revisi	Paraf
	Format penulisan . Daftar pustaka . Susunan perancangan dan hasil Harus sesuai	


Palembang, Juli 2022
 Dosen Penguji,



Ikhtison Mekongga, ST., M.Kom
 NIP:197705242000091002

No. Dok. :	Tgl. Berlaku :	No. Rev. :
	KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 Website : www.polsri.ac.id E-mail : info@polsri.ac.id	 
REVISI TUGAS AKHIR (TA)		

Dosen Penguji : Slamet Widodo, S.Kom., M.Kom
 Nama Mahasiswa : Damian Chatur Prasasti
 NIM : 061930701611
 Jurusan /Program Studi : Teknik Komputer/D3 Teknik Komputer
 Judul LA/ Skripsi : Aplikasi Pengelolaan Penjualan Besi Bangunan Berbasis Web
 Pada TB. Sukses Abadi Bersama

No	Uraian Revisi	Paraf
	<p>harga produk semen. dan pasir / besi / rebar.</p> <p style="text-align: center;">✓</p>	


Palembang, Juli 2022
 Dosen Penguji,




Slamet Widodo, S.Kom., M.Kom
 NIP:197305162002121001

No. Dok. :	Tgl. Berlaku :	No. Rev. :
	KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN, RISET DAN TEKNOLOGI POLITEKNIK NEGERI SRIWIJAYA Jalan Srijaya Negara, Palembang 30139. Telp. 0711-353414 Website : www.polsri.ac.id E-mail : info@polsri.ac.id	
REVISI TUGAS AKHIR (TA)		

Dosen Penguji : Isnainy Azro, S.Kom, M.Kom
 Nama Mahasiswa : Damian Chatur Prasasti
 NIM : 061930701611
 Jurusan /Program Studi : Teknik Komputer/D3 Teknik Komputer
 Judul LA/ Skripsi : Aplikasi Pengelolaan Penjualan Besi Bangunan Berbasis Web
 Pada TB. Sukses Abadi Bersama

No	Uraian Revisi	Paraf
1	Revisi Sama dengan dosen lain	

Palembang, Juli 2022
 Dosen Penguji,


 Isnainy Azro, S.Kom, M.Kom
 NIP.197310012002122007



KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN,
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polsri.ac.id E-mail : info@polsri.ac.id



LEMBAR PELAKSANAAN REVISI LAPORAN AKHIR (LA)

Mahasiswa berikut,

Nama : Damian Chatur Prasasti
NIM : 061930701611
Jurusan/Program Studi : Teknik Komputer / DIII-Teknik Komputer
Judul Laporan Akhir : Aplikasi Pengelolaan Penjualan Besi Bangunan Berbasis Web Pada TB.
Sukses Abadi Bersama

Telah melaksanakan revisi terhadap Laporan Akhir (LA) yang diseminarkan pada hari Kamis tanggal 28 bulan Juli tahun 2022. Pelaksanaan revisi terhadap Laporan Akhir tersebut telah disetujui oleh Dosen Penilai yang memberikan revisi:

No.	Komentar	Nama Dosen Penilai	Tanggal	Tanda Tangan
1.	OK	Ema Laila, S.Kom., M.Kom	01 Agustus 2022	
2.	OK	Adi Sutrisman, S.Kom., M.Kom	01 Agustus 2022	
3.	OK	Ikhthison Mekongga, S.T., M.Kom	01 Agustus 2022	
4.	OK	Isnainy Azro, S.Kom., M.Kom	01 Agustus 2022	
5.	OK	Slamet Widodo, S.Kom., M.Kom	01 Agustus 2022	

Palembang, 01 Agustus 2022

Ketua Penilai **)

Ema Laila, S.Kom., M.Kom.

NIP. 197703292001122002

Catatan:

*) Dosen penilai yang memberikan revisi saat seminar Laporan Akhir.

***) Dosen penilai yang ditugaskan sebagai Ketua Penilai saat seminar Laporan Akhir.

Lembaran pelaksanaan revisi ini harus dilampirkan dalam Laporan Akhir.