

TEDxNIITUniversity Software Design Specification

1. Introduction

1.1 Purpose of this document

The aim of this software design document is to provide specifications of the website created for an event. This document will help the reader in understanding the working of the above mentioned website.

Additionally, the purpose of this document is to elucidate the details of our software and to provide a high-level design framework around which to build our online reusable items inventory.

The Software Design Specification (SDS) document will have two major releases:

1. Version 1 focuses on specifying a high-level view of the architecture of our system, and on the interaction between the user and the system.
2. Version 2 focuses on detailing a low-level view of each component of the software and how the components interact with each other.

1.2 Scope of the development project

The website will help the users in obtaining information about the event that is going to be held in the university premises. The users will vary from students, teachers to business professionals. Live web-cast will be available to the users. The users can also register and buy the tickets for the event on the website linked through a payment gateway. The helpline number as well as a feedback form would be available to the users.

The organizers require the website to publicize the event as well as to provide details. The theme for this project will be custom-ed and be made on WordPress for non-developers to make changes to. The main objective of the product is its maintainability and easy to understand functionalities which make the product user-friendly.

1.3 Definitions, acronyms, and abbreviations

- Tedx: - A Tedx event is a local gathering where live TED-like talks and videos previously recorded at TED conferences are shared with the community. TEDx events are fully planned and coordinated independently, on a community-by-community basis.
- PC - Personal Computer (desktop or laptop)
- HTML-Hypertext Markup Language
- CSS-Cascading Stylesheets
- PHP – Hypertext Pre Processing

1.4 References

IEEE Software Engineering Standards Committee, “IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specifications, October 20, 1998.

- Official Guidelines:

<https://www.ted.com/participate/organize-a-local-tedx-event/tedx-organizer-guide/branding-promotions/your-event-website>

We have tried to make use of the above site to make a benchmark and to build our product better. Please note that no content has been copied from their product either in the form of code or whatever content is there on their website.

1.5 Overview of document

This document is made for individuals directly involved in the development of the website. The document is supposed to be read sequentially but the users can jump to any section they find helpful.

The main focus of this document is to provide the description of system architecture and also the overview of system’s major components.

The Detailed description of components section will be the main focus of version 2 of this document. It will describe lower-level classes, components, and functions, as well as the interaction between these internal components.

In section 3 various diagrams are available for easy understanding of the product as well as the description of logical architecture. After that execution architecture is written which contains the runtime environment, processes and the deployment view. The trade-offs that were done are present in section 5 followed by the code.

2. Conceptual Architecture/Architecture Diagram

2.1 Overview of modules / components

The system has been designed keeping scalability and extensibility in mind. We are taking great care in designing a framework which can be updated easily. Many of the anticipated changes to our system in future phases will only require adding new types of data and changing the user presentation code to make use of these new data. The framework we have designed will only require "plugging in" these new types of data without refactoring the logic that passes the data over the network, retrieves and updates the database, etc. The components/modules have been carefully prepared giving details at whatever depth possible so as to ensure proper documentation methods and guidelines are followed.

2.2 Structure and relationships

Database Engine

- Existing open source software: Mysql
- Hosts the backend database which is used for central data storage.

Server Application

- Implemented in PHP
- Provides methods and procedures that can be invoked remotely by a client application via the server procedure proxy.
 - Store client's Information.
- Central process which can make all decisions that arise due to the distributed nature of this application.
 - For instance, when a client wishes to update a project, there may be conflicts that need to be resolved if another client has updated the same project.
 - The server can coordinate conflict resolution with the client application (which can ask the user questions, if necessary).

Client Applications

- Implemented in HTML,CSS,JAVASCRIPT,JQuery,Bootstrap, fancybox etc.
- Contains all presentation logic.
- Interacts exclusively with the user.
- Communicates with the server application

3. Logical Architecture (Class Diagram, Sequence Diagram, State Diagram)

Lines and boxes are a software architect's best friend. UML can help with that, if used appropriately. LA includes "high-level design", "detailed design" and for some components, may extend even to the code.

3.1 Logical Architecture Description

Discuss some details(generic) of Logical Architecture

3.2 Component Overview

The following bulleted outline provides a basic overview of the purpose and architecture of our system's major components. The tables in the remainder of section 3 will provide more details on each component:

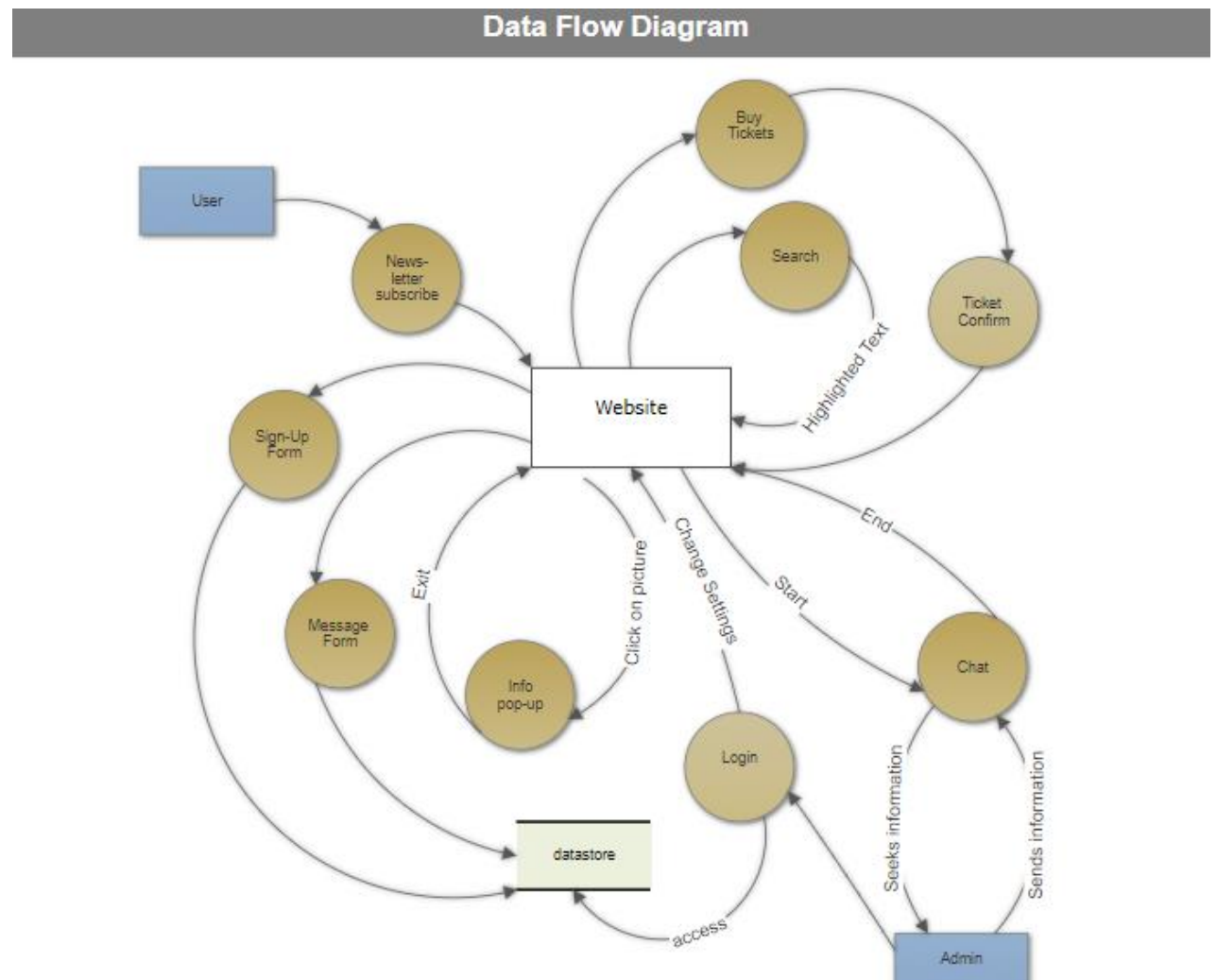
3.2.1 Database - 1(1 sign-up form & 1 Contact Us): We are using database which contain tables to represent various item's data.

3.2.2 Database - 2(Wordpress – 12 tables): We are using database which contain tables to represent various item's data.

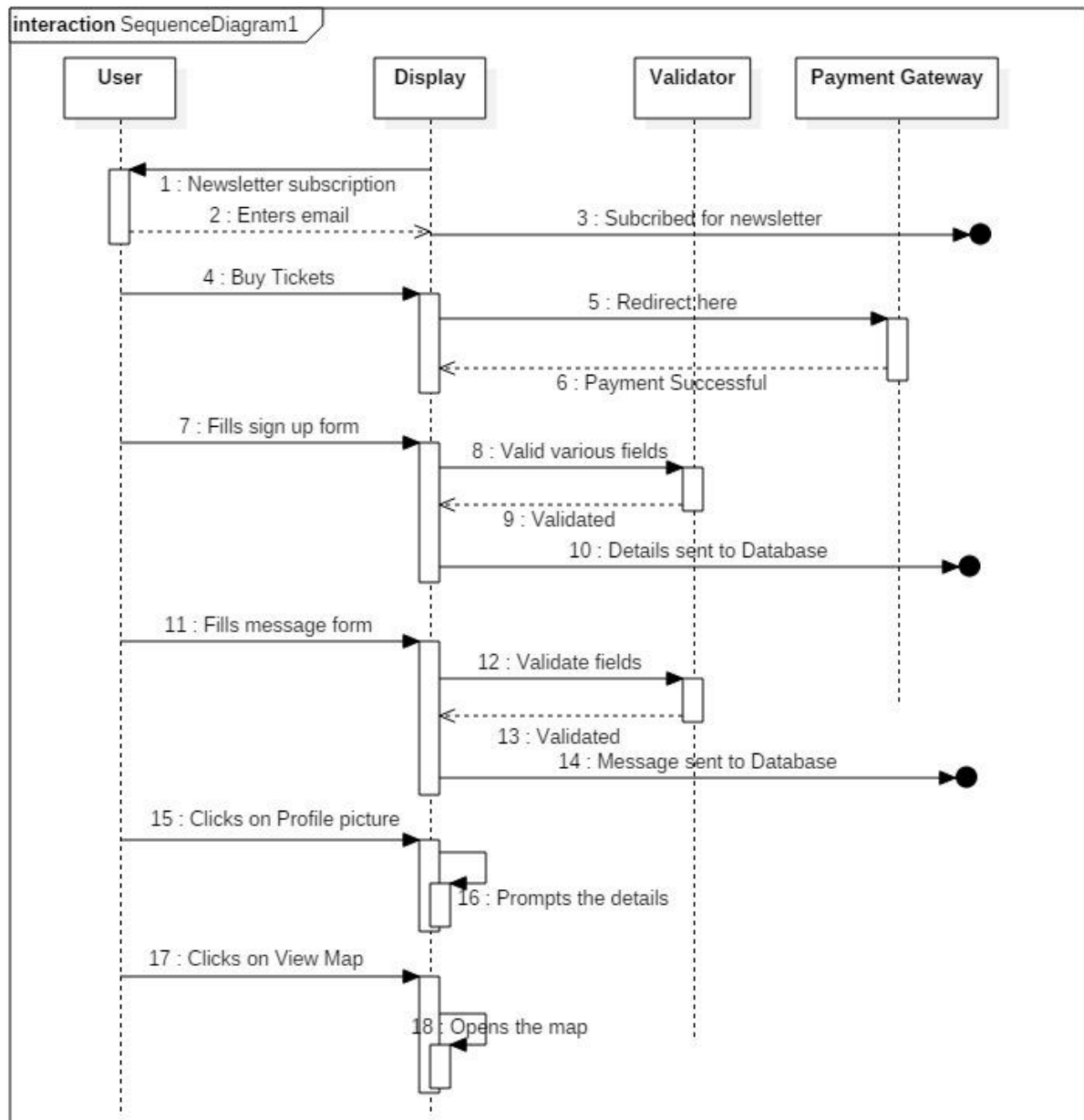
3.2.3 Server Application (Wordpress – Admin Login): Server application will the application that will be on the same system as the CMS. The client software will connect to the server software to send and retrieve data.

3.2.4 Code Functionality (Special functions – JS & PHP): Special functionality of making connection between HTML form and Databases. Also, functionality like parallax effect and countdown etc.

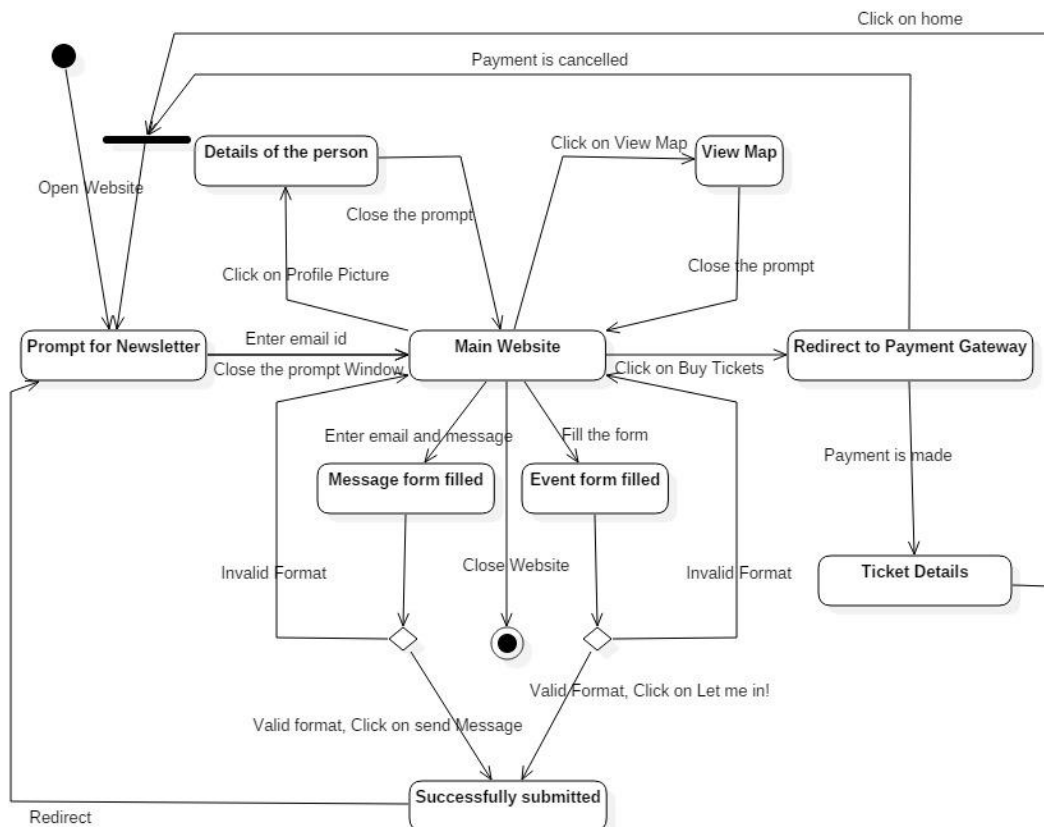
3.3 Data Flow Diagram



3.4 Sequence Diagram



3.5 State Diagram



4.0 Reuse and relationships to other products

4.1 PHP and PHP tools

From the beginning of over project development we set a goal to make use of any existing code so that we can avoid wasting of time duplicating other's works. We also decide to use freeware solutions and open source wherever possible.

Because we are creating web application, it is possible to use open source technologies in each area (we are considering our university as a base as for now). First we decide to implement our code in PHP because it is free and allows us run on a wide range of browser and operating system. We have a server from ECO Web Hosting (servers in UK).

4.2 Database and Reporting

With these decisions in place, we needed a database on which to store our records. We use MYSQL in PHPmyAdmin.

5.0 Design decisions and tradeoffs

5.1 PHP/Ruby on Rails Debate

The first issue we encountered when we started working as a group was whether to use PHP or Ruby on Rails for developing the back-end of our web portal. Two people among three were in favor of PHP as it is more widely used and is being used since a long time and one member of the group is highly experienced with working in PHP. So we ended up deciding PHP as our back-end language.

5.2 Three Tier Design

Deciding how to judiciously divide the project between all team members was another design issue. We finally decided on a 3 tier design, which is an application program organized into three major parts, each of which is distributed to different places in a network.

The three parts are:

1. The client application
2. The admin application
3. The database and programming related to managing it

A 3-tier application uses the client/server computing model. With three tiers or parts, each part can be developed concurrently by a different team of programmers. Because the programming for a tier can be changed or relocated without affecting the other tiers, the 3-tier model makes it easier for an enterprise or software packager to continually evolve an application as new needs and opportunities arise.

5.3 Schedule

As a team being in the same college at the time of the development of this project it was easy for us to manage our schedules and synchronize our schedule and work at the same time so that everything is done consistently.

6.0 Pseudocode for components

- Database – 1 (main domain tedxniituniversity.com)
 - Connection with phpmyAdmin

```
<?php

    $con = mysqli_connect('xxxxxxxxxxxxxxxxxxxx', 'data-from-site-ui-3230d1a7', 'a1b2c3$2011');
    header( "refresh:2;url=index.php" );
    if(!$con)
    {
        echo 'NOT Connected to database server!';
    }

    if(!mysqli_select_db($con, 'data-from-site-ui-3230d1a7'))
    {
        echo 'Database NOT Selected';
    }

    $Name = $_POST['Name'];
    $email = $_POST['email'];
    $phone_no = $_POST['phone_no'];
    $yes_no = $_POST['yes_no'];

    $sql = "INSERT INTO formdata (Name,email,phone_no,yes_no)
VALUES ('$Name', '$email', '$phone_no', '$yes_no')";
    if(!mysqli_query($con,$sql))
    {
        echo 'NOT INSERTED';
    }
    else
    {
        echo 'Data INSERTED Successfully!';
    }

?>
```

- Database – 2 (sub-domain blog.tedxniituniversity.com)
 - Connection with Mysql

```
<?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:
*
* * MySQL settings
* * Secret keys
* * Database table prefix
* * ABSPATH
*
* @link https://codex.wordpress.org/Editing_wp-config.php
*
* @package WordPress
*/

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress-3231ee29');

/** MySQL database username */
define('DB_USER', 'wordpress-3231ee29');

/** MySQL database password */
define('DB_PASSWORD', '27ba60cd58f2');

/** MySQL hostname */
define('DB_HOST', 'shareddb1d.hosting.stackcp.net');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');

/** 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',
'Fvuk4UuwtfId5oUKLxT811cTB8XnZ+Lz');
define('SECURE_AUTH_KEY',
'Ql7pOET1Od1vqZAVBP3DseXhQKQOdyMU');

```

```

define('LOGGED_IN_KEY',
'owDWrtb+n1r9zJ0sTKvvPiuVZI42BYyA');
define('NONCE_KEY',
'V9k7AI7tWZAyg3ouMNs4L+l4g2S1sgXZ');
define('AUTH_SALT',
'gfCfCBas84CsHXnVM0X5DiD8+A1QEAYi');
define('SECURE_AUTH_SALT',
'f7t3wnpmit1pGb643M4+SvLV4z5RnOBM');
define('LOGGED_IN_SALT',
'pF8qGchC0omQOWqBkEM+0mUekLFWYc5/');
define('NONCE_SALT',
'hnj1/r40FdIhpjM03syzXePODuTrMn7w');
/**#@-*/

/**
 * 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 Codex.
 *
 * @link https://codex.wordpress.org/Debugging_in_WordPress
 */
define('WP_DEBUG', false);

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

/** Absolute path to the WordPress directory. */
if ( !defined('ABSPATH') )
    define('ABSPATH', dirname(__FILE__) . '/');

/** Sets up WordPress vars and included files. */
require_once(ABSPATH . 'wp-settings.php');

```

- **Server Application (Wordpress – Admin Login):**

<?php

```

/**
 * WordPress Administration Bootstrap
 *
 * @package WordPress
 * @subpackage Administration
 */

/**
 * In WordPress Administration Screens
 *
 * @since 2.3.2
 */
if ( ! defined( 'WP_ADMIN' ) ) {
    define( 'WP_ADMIN', true );
}

if ( ! defined( 'WP_NETWORK_ADMIN' ) )
    define( 'WP_NETWORK_ADMIN', false );

if ( ! defined( 'WP_USER_ADMIN' ) )
    define( 'WP_USER_ADMIN', false );

if ( ! WP_NETWORK_ADMIN && ! WP_USER_ADMIN ) {
    define( 'WP_BLOG_ADMIN', true );
}

if ( isset( $_GET['import'] ) && ! defined( 'WP_LOAD_IMPORTERS' ) )
    define( 'WP_LOAD_IMPORTERS', true );

require_once( dirname( dirname( __FILE__ ) ) . '/wp-load.php' );

nocache_headers();

if ( get_option( 'db_upgraded' ) ) {
    flush_rewrite_rules();
    update_option( 'db_upgraded', false );

    /**
     * Fires on the next page load after a successful DB upgrade.
     *
     * @since 2.8.0
     */
    do_action( 'after_db_upgrade' );
} elseif ( get_option( 'db_version' ) != $wp_db_version && empty( $_POST ) ) {
    if ( ! is_multisite() ) {
        wp_redirect( admin_url( 'upgrade.php?_wp_http_referer='
        . urlencode( wp_unslash( $_SERVER['REQUEST_URI'] ) ) ) );
        exit;
    }
}

```

```

/**
 * Filters whether to attempt to perform the multisite DB upgrade
routine.
 *
 * In single site, the user would be redirected to wp-
admin/upgrade.php.
 * In multisite, the DB upgrade routine is automatically fired, but
only
 * when this filter returns true.
 *
 * If the network is 50 sites or less, it will run every time. Otherwise,
 * it will throttle itself to reduce load.
 *
 * @since 3.0.0
 *
 * @param bool $do_mu_upgrade Whether to perform the
Multisite upgrade routine. Default true.
 */
} elseif ( apply_filters( 'do_mu_upgrade', true ) ) {
    $c = get_blog_count();

    /**
     * If there are 50 or fewer sites, run every time. Otherwise,
throttle to reduce load:
     * attempt to do no more than threshold value, with some +/-
allowed.
     */
    if ( $c <= 50 || ( $c > 50 && mt_rand( 0, (int)( $c / 50 ) ) == 1
) ) {
        require_once( ABSPATH . WPINC . '/http.php' );
        $response = wp_remote_get( admin_url(
'upgrade.php?step=1' ), array( 'timeout' => 120, 'httpversion' => '1.1' ) );
        /** This action is documented in wp-
admin/network/upgrade.php */
        do_action( 'after_mu_upgrade', $response );
        unset($response);
    }
    unset($c);
}

}

require_once(ABSPATH . 'wp-admin/includes/admin.php');

auth_redirect();

// Schedule trash collection
if ( ! wp_next_scheduled( 'wp_scheduled_delete' ) && ! wp_installing() )
    wp_schedule_event(time(), 'daily', 'wp_scheduled_delete');

set_screen_options();

```

```
$date_format = __( 'F j, Y' );  
$time_format = __( 'g:i a' );
```

```
wp_enqueue_script( 'common' );
```

```
/**  
 * $pagenow is set in vars.php  
 * $wp_importers is sometimes set in wp-admin/includes/import.php  
 * The remaining variables are imported as globals elsewhere, declared as  
globals here  
 *  
 * @global string $pagenow  
 * @global array $wp_importers  
 * @global string $hook_suffix  
 * @global string $plugin_page  
 * @global string $typenow  
 * @global string $taxnow  
 */  
global $pagenow, $wp_importers, $hook_suffix, $plugin_page, $typenow,  
$taxnow;
```

```
$page_hook = null;
```

```
$editing = false;
```

```
if ( isset($_GET['page']) ) {  
    $plugin_page = wp_unslash( $_GET['page'] );  
    $plugin_page = plugin_basename($plugin_page);  
}
```

```
if ( ( isset( $_REQUEST['post_type'] ) ) && post_type_exists(  
$_REQUEST['post_type'] ) )  
    $typenow = $_REQUEST['post_type'];  
else  
    $typenow = '';
```

```
if ( ( isset( $_REQUEST['taxonomy'] ) ) && taxonomy_exists(  
$_REQUEST['taxonomy'] ) )  
    $taxnow = $_REQUEST['taxonomy'];  
else  
    $taxnow = '';
```

```
if ( WP_NETWORK_ADMIN )  
    require(ABSPATH . 'wp-admin/network/menu.php');  
elseif ( WP_USER_ADMIN )  
    require(ABSPATH . 'wp-admin/user/menu.php');
```

```

else
    require(ABSPATH . 'wp-admin/menu.php');

if ( current_user_can( 'manage_options' ) ) {
    wp_raise_memory_limit( 'admin' );
}

/**
 * Fires as an admin screen or script is being initialized.
 *
 * Note, this does not just run on user-facing admin screens.
 * It runs on admin-ajax.php and admin-post.php as well.
 *
 * This is roughly analogous to the more general {@see 'init'} hook, which
 * fires earlier.
 *
 * @since 2.5.0
 */
do_action( 'admin_init' );

if ( isset($plugin_page) ) {
    if ( !empty($typenow) )
        $the_parent = $pagenow . '?post_type=' . $typenow;
    else
        $the_parent = $pagenow;
    if ( ! $page_hook = get_plugin_page_hook($plugin_page,
$the_parent) ) {
        $page_hook = get_plugin_page_hook($plugin_page,
$plugin_page);

        // Back-compat for plugins using add_management_page().
        if ( empty( $page_hook ) && 'edit.php' == $pagenow && "
!= get_plugin_page_hook($plugin_page, 'tools.php') ) {
            // There could be plugin specific params on the URL,
            so we need the whole query string
            if ( !empty($_SERVER[ 'QUERY_STRING' ]) )
                $query_string = $_SERVER[
'QUERY_STRING' ];
            else
                $query_string = 'page=' . $plugin_page;
            wp_redirect( admin_url('tools.php?' . $query_string)
);
            exit;
        }
    }
    unset($the_parent);
}

$hook_suffix = '';
if ( isset( $page_hook ) ) {

```

```

        $hook_suffix = $page_hook;
    } elseif ( isset( $plugin_page ) ) {
        $hook_suffix = $plugin_page;
    } elseif ( isset( $pagenow ) ) {
        $hook_suffix = $pagenow;
    }

    set_current_screen();

    // Handle plugin admin pages.
    if ( isset($plugin_page) ) {
        if ( $page_hook ) {
            /**
             * Fires before a particular screen is loaded.
             *
             * The load-* hook fires in a number of contexts. This hook
            is for plugin screens
             * where a callback is provided when the screen is registered.
             *
             * The dynamic portion of the hook name, `$page_hook`,
            refers to a mixture of plugin
             * page information including:
             * 1. The page type. If the plugin page is registered as a
            submenu page, such as for
             * Settings, the page type would be 'settings'. Otherwise the
            type is 'oplevel'.
             * 2. A separator of '_page_'.
             * 3. The plugin basename minus the file extension.
             *
             * Together, the three parts form the `$page_hook`. Citing
            the example above,
             * the hook name used would be 'load-
            settings_page_pluginbasename'.
             *
             * @see get_plugin_page_hook()
             *
             * @since 2.1.0
             */
            do_action( "load-{$page_hook}" );
            if ( ! isset($_GET['noheader']) )
                require_once(ABSPATH . 'wp-admin/admin-
            header.php');

            /**
             * Used to call the registered callback for a plugin screen.
             *
             * @ignore
             * @since 1.5.0
             */
            do_action( $page_hook );

```



```

    } else {
        if ( validate_file( $plugin_page ) ) {
            wp_die( __( 'Invalid plugin page.' ) );
        }

        if ( !( file_exists(WP_PLUGIN_DIR . "/" . $plugin_page) ) &&
            is_file(WP_PLUGIN_DIR . "/" . $plugin_page) ) &&
            !( file_exists(WPMU_PLUGIN_DIR . "/" . $plugin_page) ) &&
            is_file(WPMU_PLUGIN_DIR . "/" . $plugin_page) ) )
            wp_die(sprintf(__( 'Cannot load %s.' ),
                htmlentities($plugin_page)));

        /**
         * Fires before a particular screen is loaded.
         *
         * The load-* hook fires in a number of contexts. This hook
         * is for plugin screens
         * where the file to load is directly included, rather than the
         * use of a function.
         *
         * The dynamic portion of the hook name, `$plugin_page`,
         * refers to the plugin basename.
         *
         * @see plugin_basename()
         *
         * @since 1.5.0
         */
        do_action( 'load-' . $plugin_page );

        if ( !isset($_GET['noheader']) )
            require_once(ABSPATH . 'wp-admin/admin-
header.php');

        if ( file_exists(WPMU_PLUGIN_DIR . "/" . $plugin_page) )
            include(WPMU_PLUGIN_DIR . "/" . $plugin_page);
        else
            include(WP_PLUGIN_DIR . "/" . $plugin_page);
    }

    include(ABSPATH . 'wp-admin/admin-footer.php');

    exit();
} elseif ( isset( $_GET['import'] ) ) {

    $importer = $_GET['import'];

    if ( ! current_user_can( 'import' ) ) {
        wp_die( __( 'Sorry, you are not allowed to import content.' )
    );
    }
}

```

```

        if ( validate_file($importer) ) {
            wp_redirect( admin_url( 'import.php?invalid=' . $importer
        ));
            exit;
        }

        if ( ( ! isset($wp_importers[$importer]) || !
is_callable($wp_importers[$importer][2]) ) {
            wp_redirect( admin_url( 'import.php?invalid=' . $importer
        ));
            exit;
        }

/**
 * Fires before an importer screen is loaded.
 *
 * The dynamic portion of the hook name, `{$importer}`, refers to the
importer slug.
 *
 * @since 3.5.0
 */
do_action( "load-importer-{$importer}" );

$parent_file = 'tools.php';
$submenu_file = 'import.php';
$title = __('Import');

if ( ! isset($_GET['noheader']) )
    require_once(ABSPATH . 'wp-admin/admin-header.php');

require_once(ABSPATH . 'wp-admin/includes/upgrade.php');

define('WP_IMPORTING', true);

/**
 * Whether to filter imported data through kses on import.
 *
 * Multisite uses this hook to filter all data through kses by default,
 * as a super administrator may be assisting an untrusted user.
 *
 * @since 3.1.0
 *
 * @param bool $force Whether to force data to be filtered through
kses. Default false.
 */
if ( apply_filters( 'force_filtered_html_on_import', false ) ) {
    kses_init_filters(); // Always filter imported data with kses
on multisite.
}

```

```

call_user_func($wp_importers[$importer][2]);

include(ABSPATH . 'wp-admin/admin-footer.php');

// Make sure rules are flushed
flush_rewrite_rules(false);

exit();
} else {
/**
 * Fires before a particular screen is loaded.
 *
 * The load-* hook fires in a number of contexts. This hook is for
core screens.
 *
 * The dynamic portion of the hook name, `$pagenow`, is a global
variable
 * referring to the filename of the current page, such as 'admin.php',
 * 'post-new.php' etc. A complete hook for the latter would be
 * 'load-post-new.php'.
 *
 * @since 2.1.0
 */
do_action( 'load-{$pagenow}' );

/**
 * The following hooks are fired to ensure backward compatibility.
 * In all other cases, 'load-' . $pagenow should be used instead.
 */
if ( $typenow == 'page' ) {
    if ( $pagenow == 'post-new.php' )
        do_action( 'load-page-new.php' );
    elseif ( $pagenow == 'post.php' )
        do_action( 'load-page.php' );
} elseif ( $pagenow == 'edit-tags.php' ) {
    if ( $taxnow == 'category' )
        do_action( 'load-categories.php' );
    elseif ( $taxnow == 'link_category' )
        do_action( 'load-edit-link-categories.php' );
} elseif ( 'term.php' === $pagenow ) {
    do_action( 'load-edit-tags.php' );
}
}

if ( ! empty( $_REQUEST['action'] ) ) {
/**
 * Fires when an 'action' request variable is sent.
 *
 * The dynamic portion of the hook name, `$_REQUEST['action']`,

```

```

    * refers to the action derived from the `GET` or `POST` request.
    *
    * @since 2.6.0
    */
    do_action( 'admin_action_' . $_REQUEST['action'] );
}

```

- Other functionality (JS and Fancybox)
 - Parallax Effect:

```

var parallax = function(id, val) {
    if ($(window).scrollTop() > id.offset().top -
    $(window).height() && $(window).scrollTop() < id.offset().top +
    id.outerHeight()) {
        var px = parseInt($(window).scrollTop() -
        (id.offset().top - $(window).height()))
        px *= -val;
        id.css({
            'background-position': 'center ' + px +
            'px'
        })
    }
}

```

```

$(window).load(function() {
    if($('.parallax').length) {
        $('.parallax').each(function() {
            parallax($(this), 0.1);
        })
    }
    $(window).scroll(function() {
        if($('.parallax').length) {
            $('.parallax').each(function() {
                parallax($(this), 0.1);
            })
        }
    })
    $('.map-view').on('click',function(){
        $('.map-overlay').fadeOut();
        $('.back-btn').fadeIn();

        return false;
    })

    $('.back-btn').on('click',function(){
        $('.back-btn').fadeOut();
        $('.map-overlay').fadeIn();
    })
}

```

```

        return false;
    })

```

- Custom Google Maps:

```

if ($('#map-canvas').length) {
    var map = new GMaps({
        div : '#map-canvas',
        lat : 27.962866,
        lng : 76.403188,
        disableDefaultUI : true,
        zoom : 17,
        scrollwheel : true,
        draggable: true
    });

    map.drawOverlay({
        lat : map.getCenter().lat(),
        lng : map.getCenter().lng(),
        content : '<a href="#"'
class="mapmarker"><i class="fa fa-map-marker"></i></a>',
        verticalAlign : 'top',
        horizontalAlign : 'center'
    });

    if ($(window).width() >= 1200) {
        map.setOptions({
            styles : Site.styles,
            center : new
google.maps.LatLng(41.401836, -74.329801),
        });
    } else if ($(window).width() >= 992) {
        map.setOptions({
            styles : Site.styles,
            center : new
google.maps.LatLng(41.401836, -74.331801),
        });
    } else if ($(window).width() >= 768) {
        map.setOptions({
            styles : Site.styles,
            center : new
google.maps.LatLng(41.401836, -74.329801),
        });
    } else {
        map.setOptions({
            styles : Site.styles,

```

```

                                center      :      new
google.maps.LatLng(41.400136, -74.332562),
                                });
                                }
                                }

```

○ **Countdown Effect**

```

$("#defaultCountdown").countdown({
    //Time set = Year, Month,Date; Starts from 0-jan
    month
    until: new Date(2018, 0, 14)
});

```

-----Thank You-----