Smart WP Chat Pro

Multi-Agent WhatsApp Chat Plugin for WordPress

1. Introduction

Smart WP Chat Pro is a multi-agent WhatsApp chat plugin for WordPress that makes it easy for website owners to connect visitors with the most relevant support agent.

With this plugin, you can create and manage multiple WhatsApp agents and display them using a floating widget or chat button. You also have full control over where and when the chat appears on your website. Each agent can be configured with details such as department, availability schedule, greeting message, and an offline message.

Smart WP Chat Pro is well suited for businesses, service providers, and online stores that want a simple and reliable way to communicate with customers through WhatsApp, improve response times, and offer better support without depending on external platforms.

The plugin is compatible with modern versions of WordPress, follows WordPress coding standards, and works smoothly with most themes and commonly used plugins, including WooCommerce.

2. Key Features

2.1. Floating WhatsApp Chat Widget

Display a floating WhatsApp chat widget that remains visible while visitors browse your website. The widget can be positioned on the left or right side of the screen and provides quick access to WhatsApp support without interrupting the user experience.

2.2. Multiple WhatsApp Agents

Create and manage multiple WhatsApp agents from the WordPress admin panel. Each agent can be displayed within the widget, allowing visitors to choose the most relevant contact.

2.3. Agent Availability & Offline Handling

Set availability schedules for each agent. When an agent is offline, an optional offline message can be shown to inform visitors and manage expectations.

2.4. Agent-Specific Information

Assign individual details to each agent, including department name, greeting message, profile image, and WhatsApp number. This helps visitors understand who they are contacting before starting a chat.

2.5. Customizable Widget Design

Customize the appearance of the floating widget and chat buttons, including widget text, button label, icon, shape, and colors, to match your website design.

2.6. Page-Level Visibility Control

Control where the chat widget appears by enabling or disabling it on specific pages, posts, or custom post types. This allows you to show the chat only where it is relevant.

2.7. Mobile and Desktop Visibility Settings

Choose whether the floating widget should appear on desktop devices, mobile devices, or both, ensuring better control over the user experience.

2.8. WooCommerce Integration

Integrate the chat widget with WooCommerce pages such as shop, product, cart, and checkout pages to assist customers during the purchasing process.

2.9. Analytics & Event Tracking

Track user interactions with the chat widget using Google Analytics and Facebook Pixel events, helping you understand user engagement and improve support performance.

2.10. Gutenberg Block & Shortcode Support

Add WhatsApp chat buttons using a Gutenberg block or shortcode, making it easy to place chat access anywhere within your content.

2.11. Developer-Friendly Hooks & Filters

The plugin provides action hooks and filters that allow developers to extend or customize functionality without modifying core plugin files.

2.12. Lightweight & Standards Compliant

Built following WordPress coding standards, the plugin is lightweight, secure, and compatible with modern WordPress themes and plugins.

2.13. Translation Ready

The plugin is fully translated into multiple languages, including English, Spanish, French, German, Italian, Portuguese, and Russian. This makes it easy to use in international markets and reach a wider audience.

3. Requirements

- WordPress 5.8+
- PHP 7.4+
- MySQL 5.6+ or MariaDB 10.1+
- HTTPS recommended
- WooCommerce (optional)

4. Getting Started

4.1 Install Plugin

- Go to Plugins → Add New
- Click Upload Plugin
- Upload ZIP file
- Install & Activate

4.2 Update Plugin

- Backup your site
- Deactivate old version
- Upload new ZIP
- Activate plugin

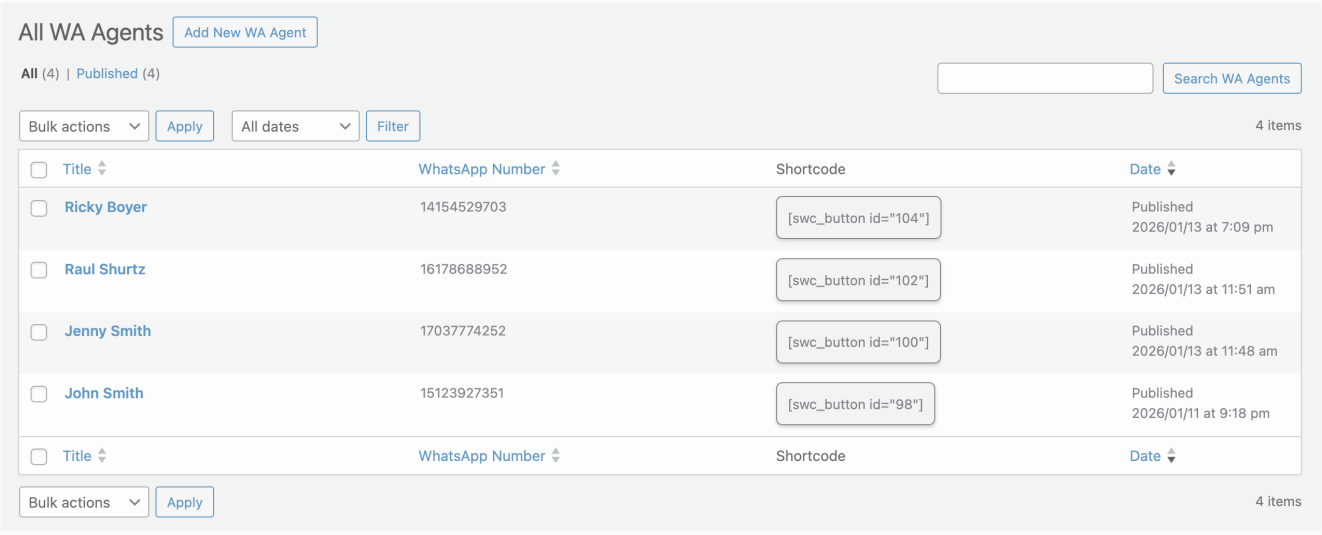
> Plugin settings are preserved during updates.

5. How It Works

WA Agents

5.1. All WA Agents

Manage all WhatsApp agents in one place, where you can easily view the agent name, WhatsApp number or chat URL, and the associated agent button shortcode.



Screenshot: Smart WP Chat > All WA Agents

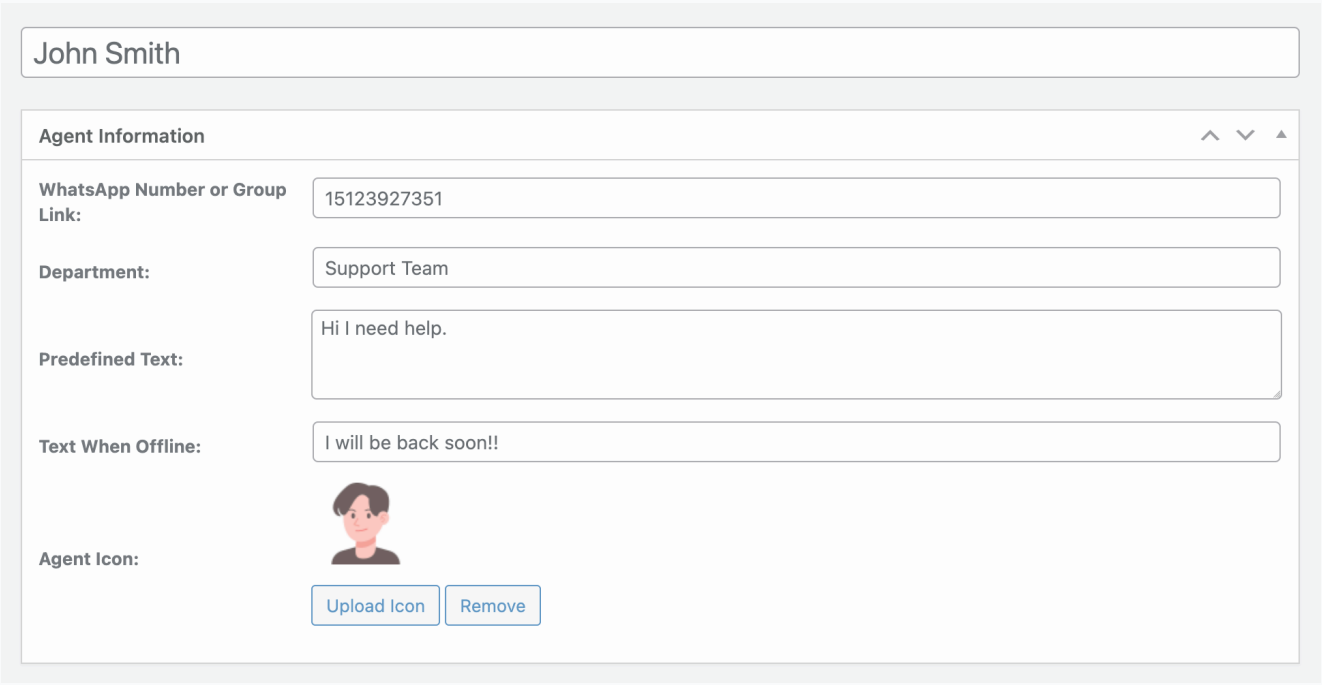
5.2 Add WA Agent

The Add New Agent section allows you to create a new WhatsApp agent that can be used in the floating widget or as an individual chat button.

Each agent can be configured with contact details, availability, messages, and button styling.

To add a new agent, navigate to **All WA Agents** → **Add New WA Agent** and configure the fields described below.

Agent Information



Screenshot: Add New WA Agent (Agent Information)

Agent Name : Add the display name for agent.

WhatsApp Number or Group Link : Add the WhatsApp number in international format or a WhatsApp group link. i.e. 1234567890 or https://chat.whatsapp.com/GroupLink

Department : Add the department name associated with the agent. i.e. Support Team, Technical Team, etc.

Predefined Text : Add a predefined text message to be sent when the agent is selected.

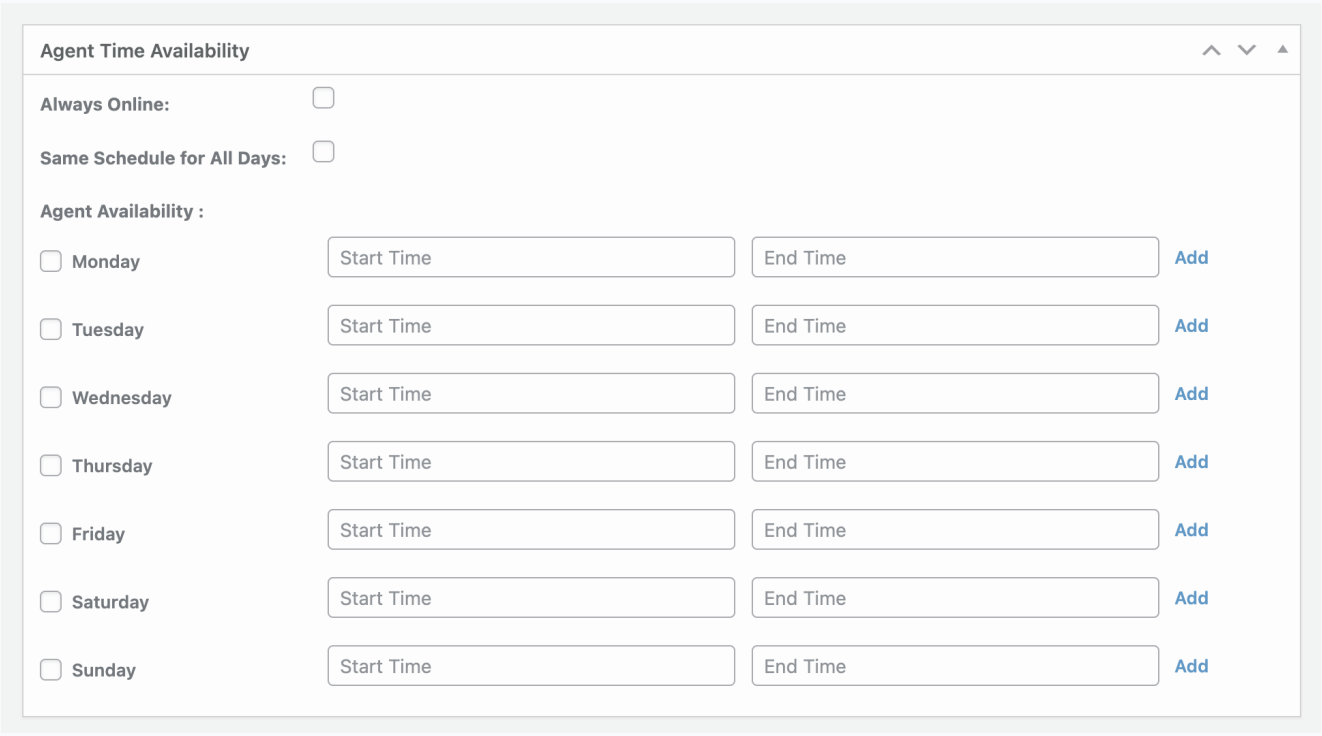
Text When Offline : Add a message to be displayed when the agent is offline.

Avatar Icon : Upload an avatar icon for the agent.

Agent Time Availability

The Agent Time Availability settings allow you to define when a WhatsApp agent is available to receive messages.

Based on this configuration, the plugin automatically determines whether the agent is shown as online or offline in the floating widget.This time availability works on Wordpress set timezone **Settings > General > Timezone**



Screenshot: Add New WA Agent (Agent Time Availability)

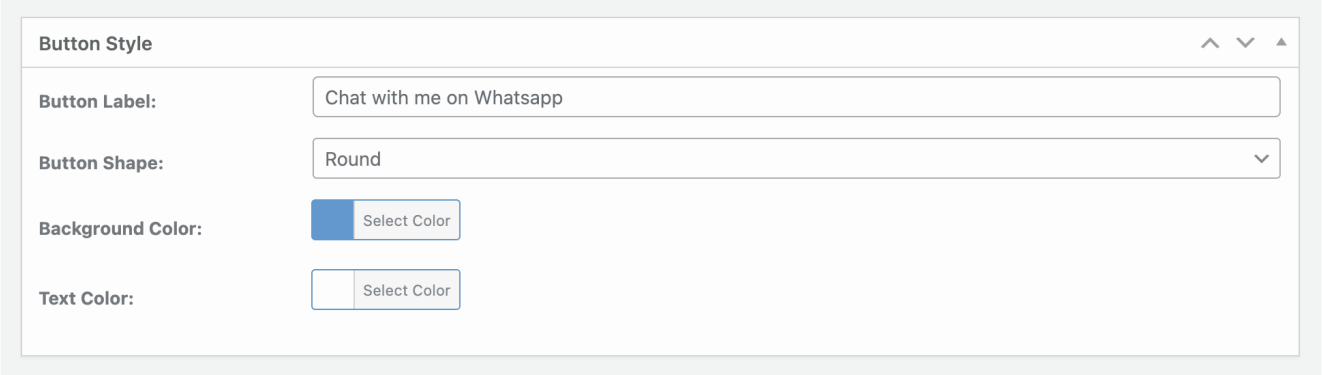
Always Online : Set the agent to always appear online, regardless of actual availability.

Same Schedule for All Days : Set the same schedule (start and end time) for all days of the week. You can add multiple time slots by clicking the Add button.

Agent Availability : Set different start and end times for each day of the week. You can also add multiple time slots for a single day by clicking the Add button.

Button Style

The Button Style settings allow you to customize the appearance of the chat button for a WhatsApp agent. These styles are applied when the agent's WhatsApp chat button shortcode is displayed on the front end.



Screenshot: Add New WA Agent (Button Style)

Button Label : Add a custom label for the text displayed on the chat button.

Button Shape : Choose the shape of the chat button (e.g., square, circle).

Background Color : Choose the background color of the chat button.

Text Color : Choose the text color of the chat button.

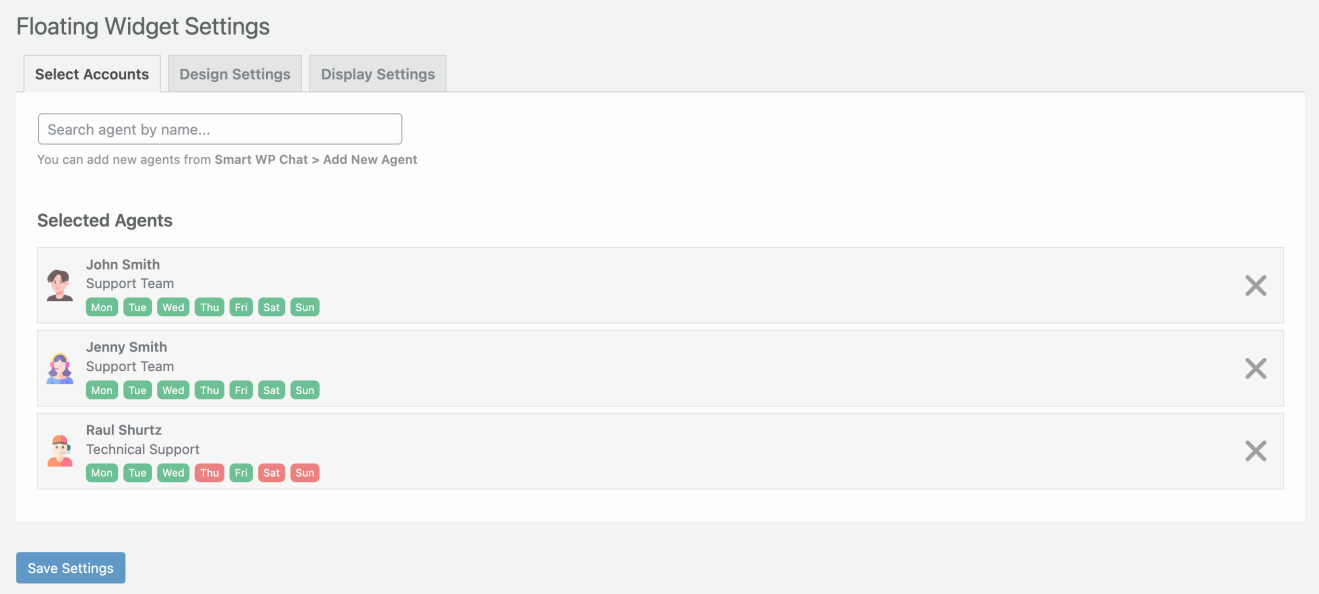
5.2 Floating Widget

The floating widget is the primary interface of Smart WP Chat Pro that allows visitors to start a WhatsApp conversation directly from any page on your website.

Once the Floating Widget configuration is completed, the widget appears as a floating button on the front end of the website. The button remains visible while users scroll the page, allowing quick and consistent access to WhatsApp support without interrupting the browsing experience.

To configure the floating widget, navigate to **Smart WP Chat → Floating Widget** and configure the fields described below.

5.2.1 Select Accounts



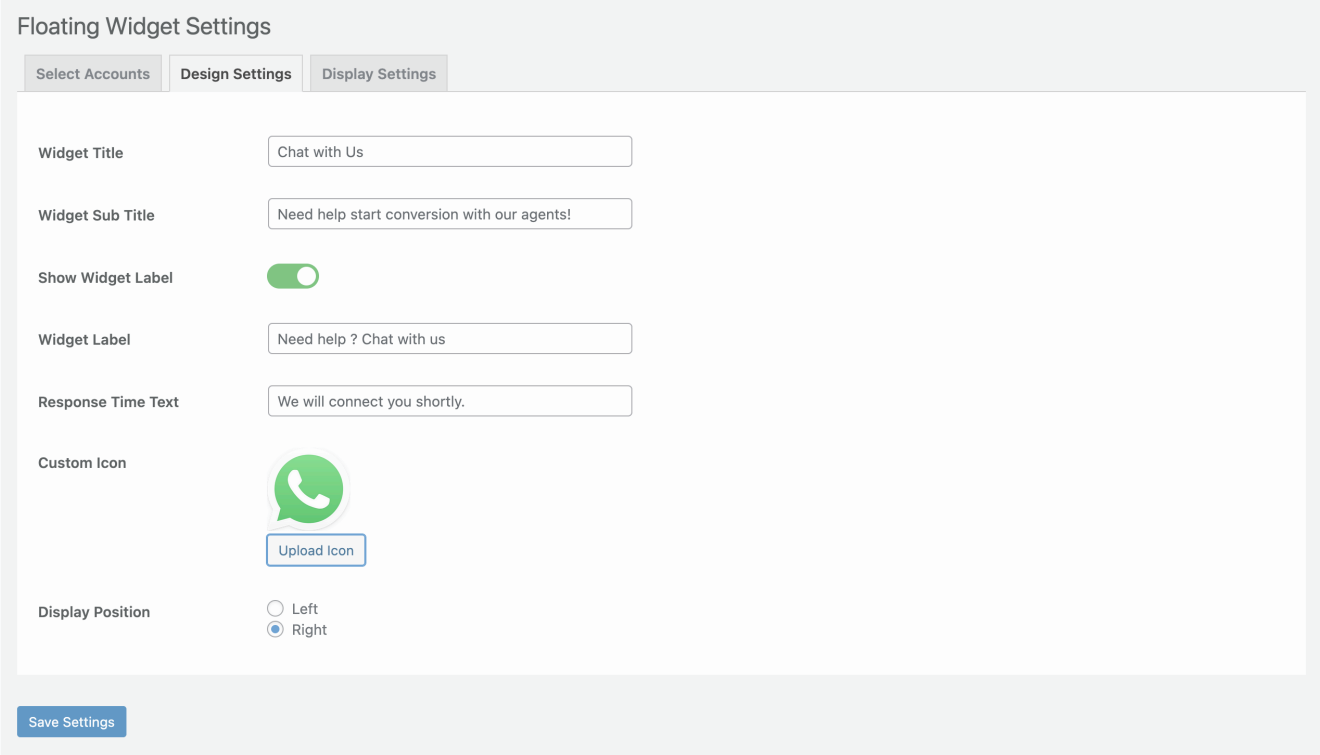
Screenshot: Floating Widget Settings (Select Agents)

Select one or more WhatsApp agents to be displayed in the floating widget. Search for an agent by name using the **Search agent by name** field, then click on the agent box from the dropdown list.**Note:** The search works after entering at least 2 letters.

You can add new agent from **Smart WP Chat → Add New Agent**

5.2.2 Design Settings

The settings in this tab are used to customize the appearance of the floating widget. Each field is explained below.



Screenshot: Floating Widget Settings (Design Settings)

Widget Title : This widget title will be displayed on the floating widget box header.

Widget Sub Title : This widget sub title will be displayed on the floating widget box header after widget title.

Show Widget Label : This setting allows you to show or hide the label (Tooltip) for the floating widget icon.

Widget Label : This setting allows you to change widget label. This field only works when the "Show Widget Label" is enabled.

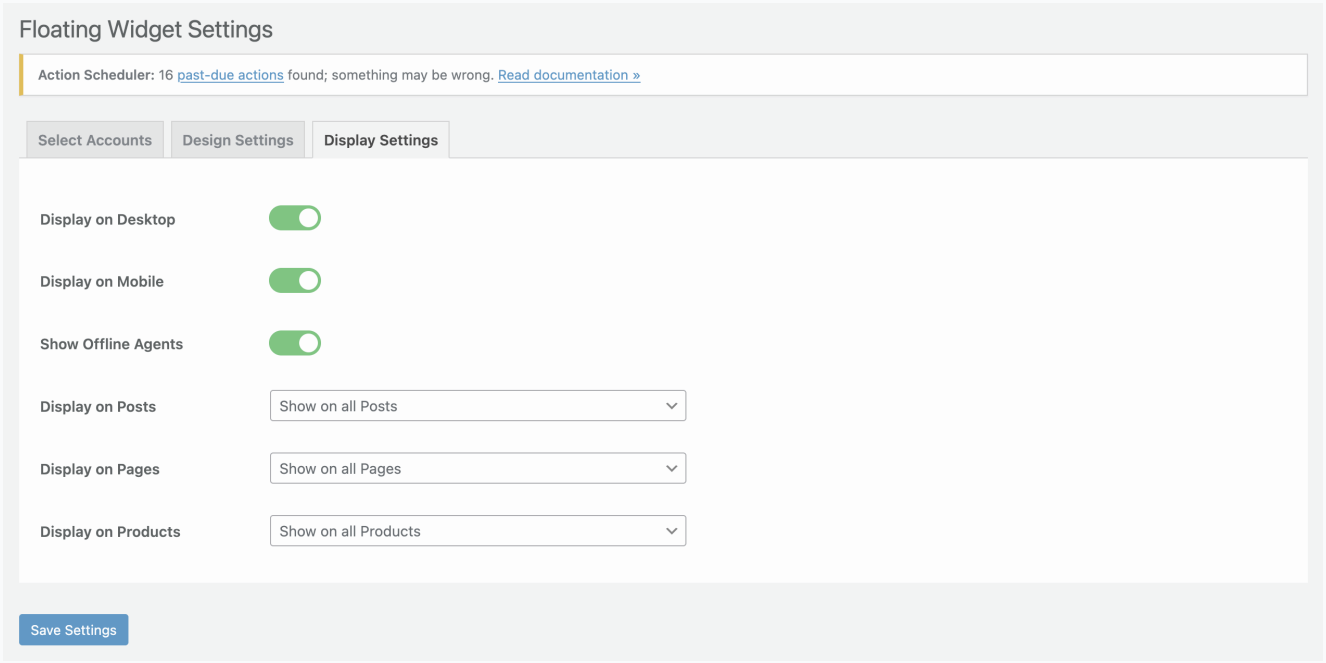
Response Time Text : This text will be visible above agent list in floating widget box.

Custom Icon : This icon will be displayed in floating widget header and in floating icon. You can use different icons for the widget header and the floating icon by using filter in functions.php. Check developer section for this.

Display Position : Choose the position of the floating widget on the screen. i.e. left or right.

5.2.3 Display Settings

The settings in this tab allow you to control where and when the floating widget appears on your website. Each field is explained below.



Screenshot: Floating Widget Settings (Display Settings)

Enable on Desktop : Enable or disable the floating widget on desktop devices.

Enable on Mobile : Enable or disable the floating widget on mobile devices.

Display on Posts : Enable or disable the floating widget on blog posts. This setting provides three options: **Show on all posts**, **Show on these posts**, and **Hide on these posts**. **Show on all posts** is the default option and displays the floating widget on all blog posts. If you want to display the floating widget only on specific posts, select **Show on these posts**. Once selected, an additional field named **Selected Posts** will appear, allowing you to choose the desired posts. Similarly, selecting **Hide on these posts** will hide the floating widget on the selected posts while keeping it visible on all others.

Display on Pages : Enable or disable the floating widget on pages. This setting provides three options: **Show on all pages**, **Show on these pages**, and **Hide on these pages**. **Show on all pages** is the default option and displays the floating widget on all pages. If you want to display the floating widget only on specific pages, select **Show on these pages**. Once selected, an additional field named **Selected Pages** will appear, allowing you to choose the desired pages. Similarly, selecting **Hide on these pages** will hide the floating widget on the selected pages while keeping it visible on all others.

Display on Products : Enable or disable the floating widget on products. This setting provides three options: **Show on all products**, **Show on these products**, and **Hide on these products**. **Show on all products** is the default option and displays the floating widget on all products. If you want to display the floating widget only on specific products, select **Show on these products**. Once selected, an additional field named **Selected Products** will appear, allowing you to choose the desired products. Similarly, selecting **Hide on these products** will hide the floating widget on the selected products while keeping it visible on all others.

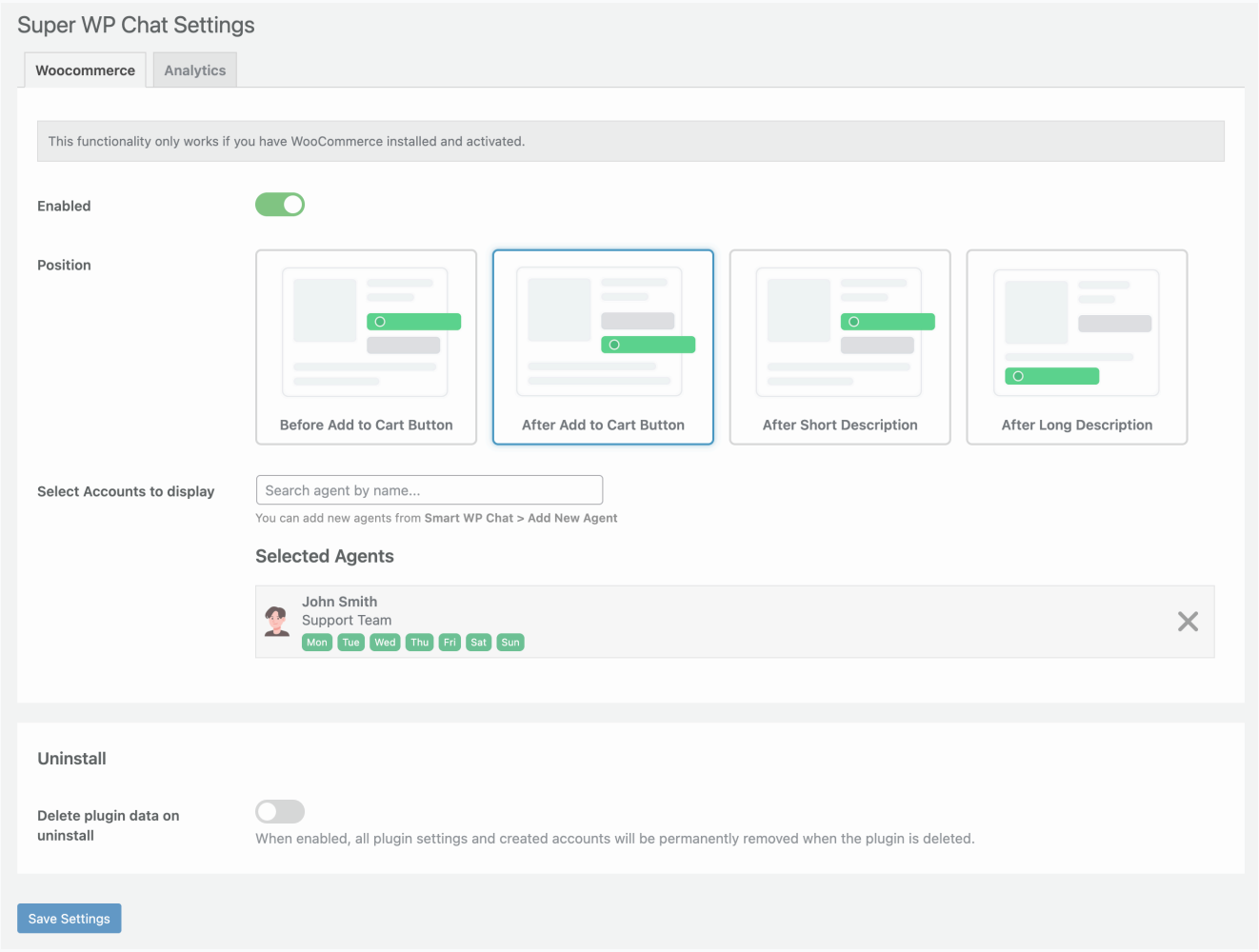
Notes: By default, this tab displays three post type options: **Post**, **Page**, and **Product**. The **Product** display option is available only when WooCommerce is installed and activated. If your website includes other custom post types, the floating widget will be displayed on those post types by default. However, if you want to apply the same display configuration to additional custom post types, you can do so by using a filter in the `functions.php` file. Refer to the Developer section for more details.

5.3 Settings

5.3.1 WooCommerce

Enable contact buttons on WooCommerce product pages. To enable this feature, go to **Smart WP Chat > Settings > WooCommerce Tab.**

Note : This feature is only available when WooCommerce is installed and activated.

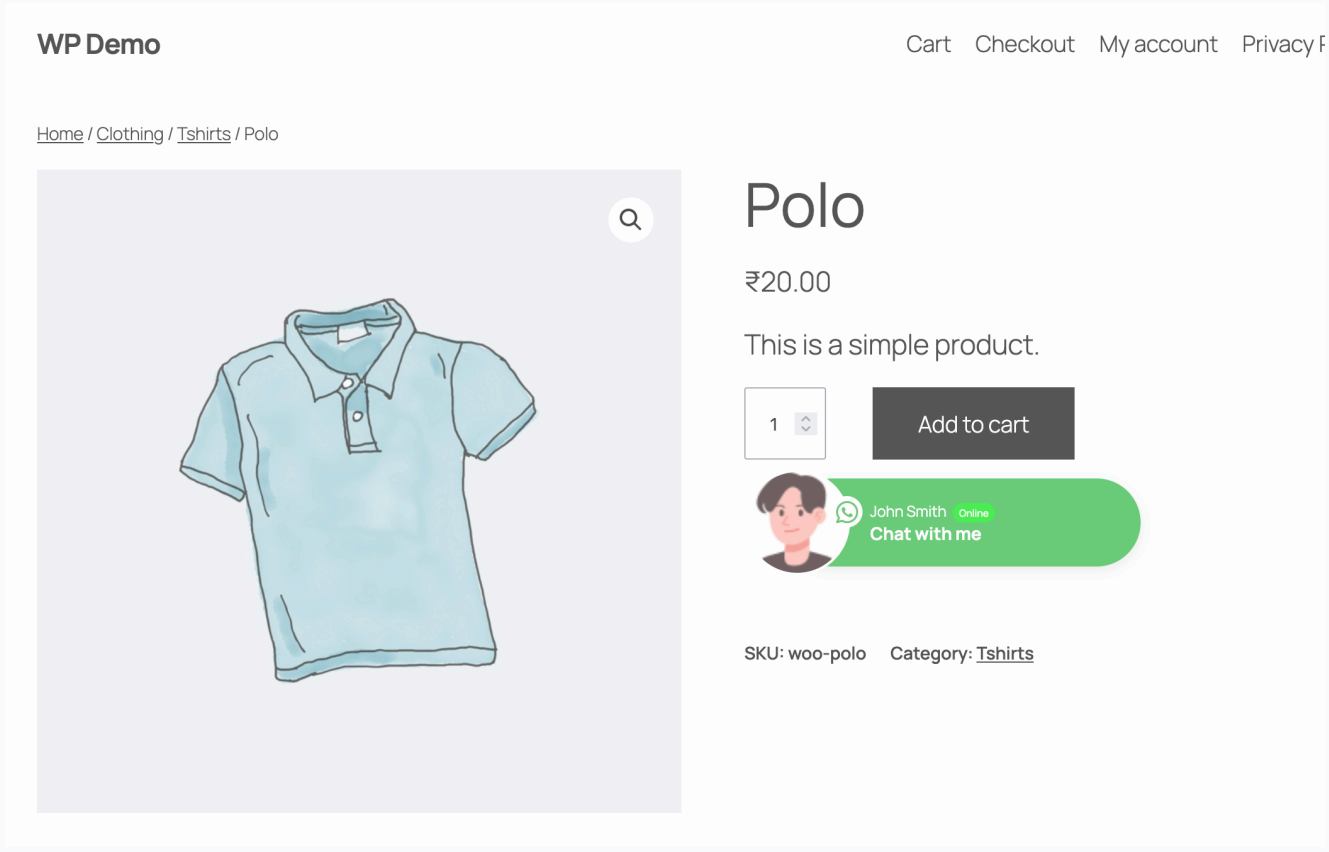


Screenshot: WooCommerce Integration Settings

Enabled: Enable or disable the WooCommerce integration.

Position: Select the position of the contact buttons on WooCommerce product pages. You can select one of the following options: **Before Add to Cart Button**, **After Add to Cart Button**, **After Short Description** or **After Long Description**.

Select Accounts to Display: Select one or more WhatsApp agents to be displayed in product pages as Contact Button(s). Search for an agent by name using the **Search agent by name** field, then click on the agent box from the dropdown list. **Note:** The search works after entering at least 2 letters.

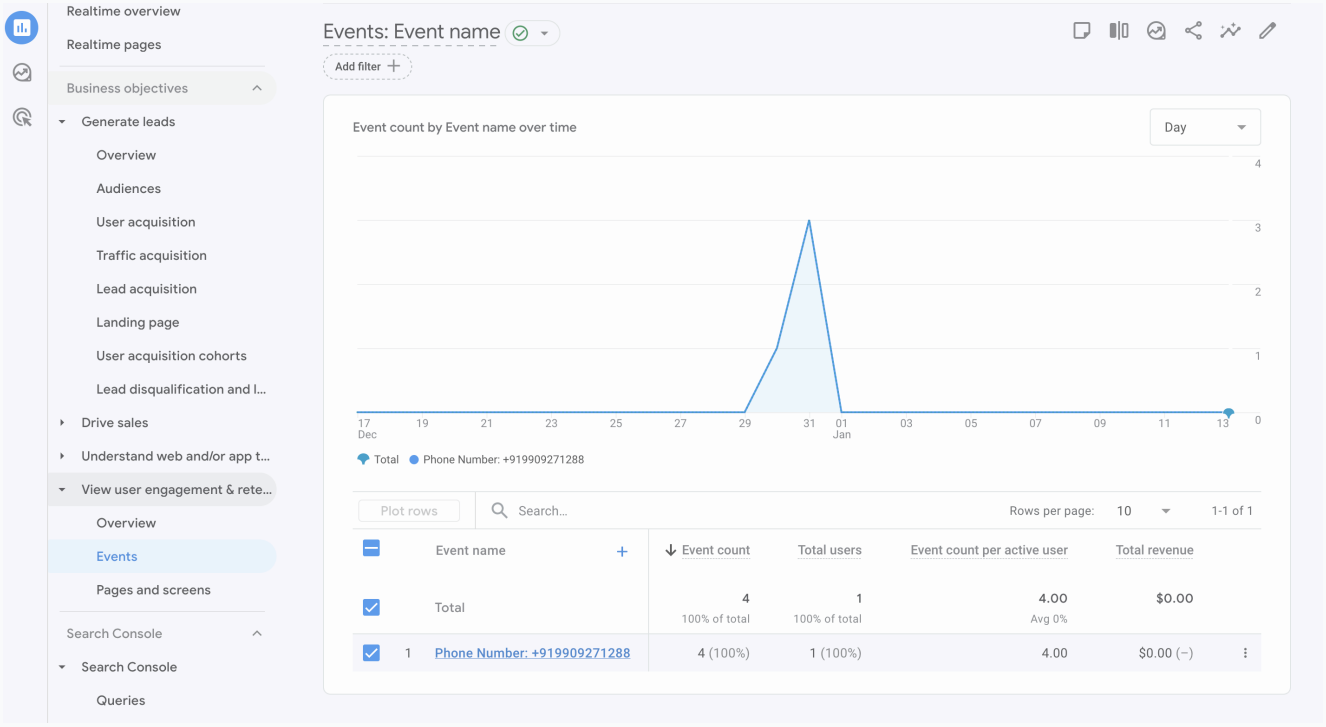


Screenshot: WooCommerce Integration Settings

5.3.2 Analytics Settings

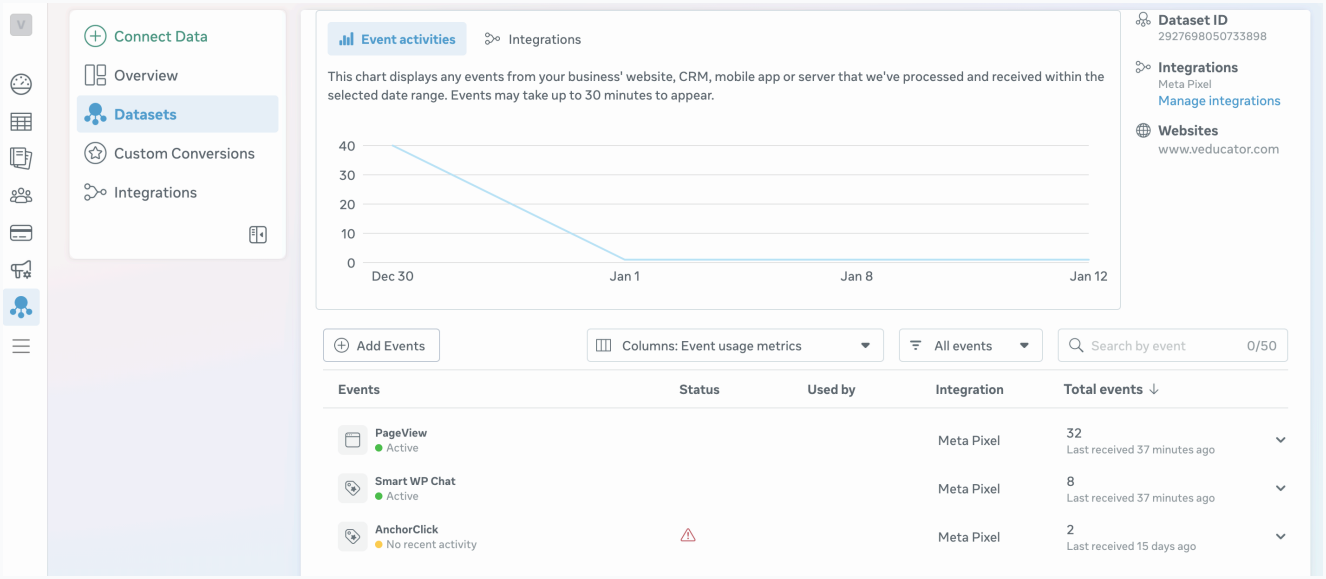
Track WhatsApp Button clicks using GA, GA4 and Facebook Pixel. This setting works when google analytics and pixel are setup for your website

Enabled Google Analytics : Enable or disable Google Analytics tracking.



Screenshot: Google Analytics & Event Tracking

Enabled Facebook Pixel : Enable or disable Facebook Pixel tracking.



Screenshot: Pixel Analytics & Event Tracking

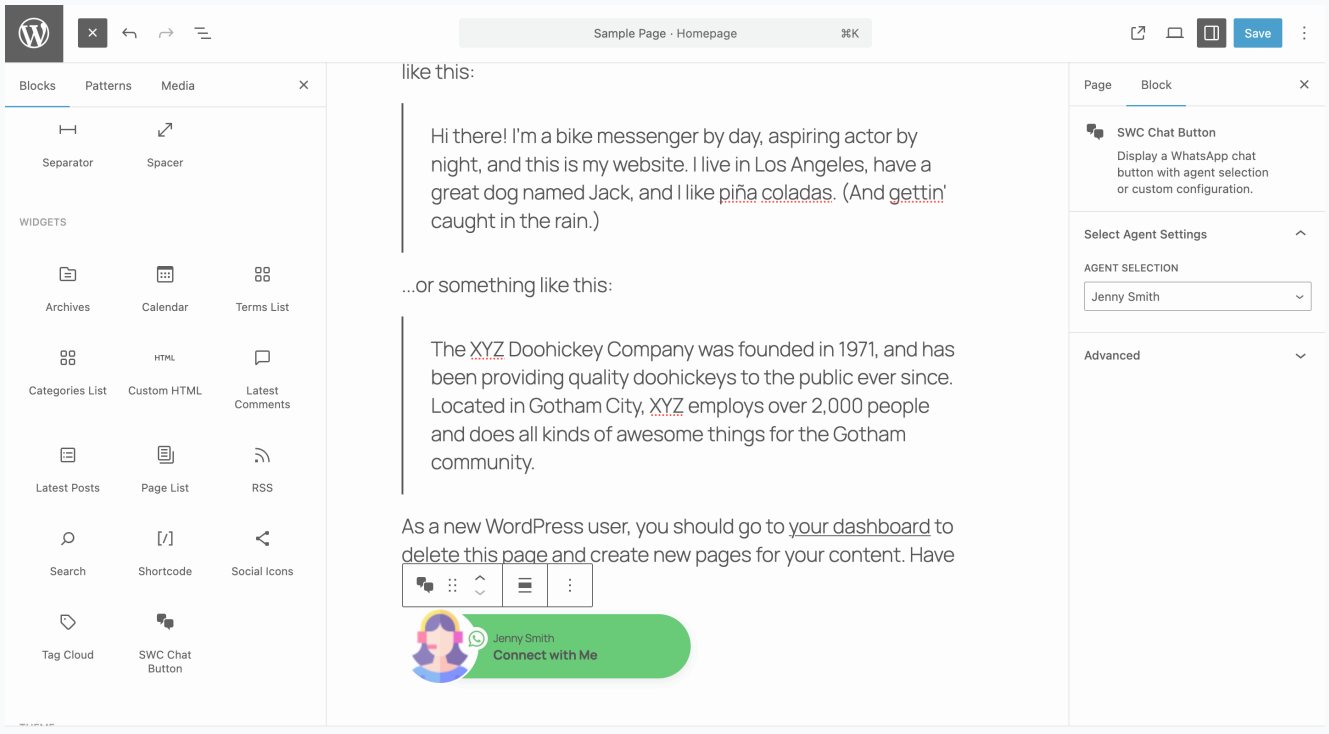
5.4. Shortcode

Smart WP chat Pro plugin provide shortcode functionality to display WhatsApp buttons anywhere on your website. You can use the shortcode `[swc_button id="12"]` to display the WhatsApp buttons. When you use this shortcode, it will display the WhatsApp buttons with the ID "12".

5.5. Gutenberg Block

Smart WP chat Pro plugin provide Gutenberg block functionality to display WhatsApp buttons anywhere on your website. You can use the Gutenberg block **SWC Chat Button** to display the WhatsApp buttons.

In this Gutenberg block, you can select a specific agent from the Select Agent dropdown list. Alternatively, you can choose the Custom Button option, which allows you to enter and use custom data.



Screenshot: Gutenburg Block

---

6. Developer Section

Hooks and Filters

Developers can extend plugin functionality using WordPress hooks and filters without modifying core files.

6.1. Action Hooks

Action Hook: swc_pro_before_chat_agent_list

Fires before rendering the agent list in the floating widget. Allows you to add custom content before the list of agents is displayed.

```
add_action( 'swc_pro_before_chat_agent_list', 'my_before_agent_list

function my_before_agent_list() {
    echo '<div class="custom-welcome-message">Welcome! Choose an ag
}
```

Action Hook: swc_pro_before_chat_agent

Fires before rendering each individual agent entry in the floating widget. Allows you to add custom content or modify the agent display.

```
add_action( 'swc_pro_before_chat_agent', 'my_before_agent', 10, 1 )

function my_before_agent( $agent_id ) {
    $custom_badge = get_post_meta( $agent_id, 'custom_badge', true
    if ( $custom_badge ) {
        echo '' . esc_html( $custom_badge ) . '';
    }
}
```

Action Hook: swc_pro_after_chat_agent

Fires after rendering each individual agent entry in the floating widget. Allows you to add custom content after each agent.

```
add_action( 'swc_pro_after_chat_agent', 'my_after_agent', 10, 1 );

function my_after_agent( $agent_id ) {
    $rating = get_post_meta( $agent_id, 'agent_rating', true );
    if ( $rating ) {
        echo '<div class="agent-rating">Rating: ' . esc_html( $rati
    }
}
```

Action Hook: swc_pro_after_chat_agent_list

Fires after rendering the complete agent list in the floating widget. Allows you to add custom content after all agents are displayed.

```
add_action( 'swc_pro_after_chat_agent_list', 'my_after_agent_list'
function my_after_agent_list() {
    echo '<div class="custom-footer">Need help? Contact us at suppo
}
```

Action Hook: swc_pro_before_woocommerce_chat_button

Fires before rendering a WooCommerce chat button on product pages. Allows you to add custom content or modify the button container.

```
add_action( 'swc_pro_before_woocommerce_chat_button', 'my_before_wo

function my_before_woo_button() {
    echo '<div class="woo-chat-wrapper">';
    echo '<p class="chat-prompt">Have questions about this product?
    echo '</div>';
}
```

Action Hook: swc_pro_after_woocommerce_chat_button

Fires after rendering a WooCommerce chat button on product pages. Allows you to add custom content after the button.

```
add_action( 'swc_pro_after_woocommerce_chat_button', 'my_after_woo_

function my_after_woo_button() {
    echo '<p class="response-time">Average response time: 2 minutes
}
```

6.2. Filter Hooks

Filter Hook: swc_pro_agent_info_metabox_title

Filters the title of the Agent Information meta box.

```
add_filter( 'swc_pro_agent_info_metabox_title', 'my_custom_metabox_

function my_custom_metabox_title( $title ) {
    return 'Agent Details & Contact Information';
}
```

Filter Hook: swc_pro_agent_timing_metabox_title

Filters the title of the Agent Time Availability meta box.

```
add_filter( 'swc_pro_agent_timing_metabox_title', 'my_timing_title'

function my_timing_title( $title ) {
    return 'Working Hours & Schedule';
}
```

Filter Hook: swc_pro_button_style_metabox_title

Filters the title of the Button Style meta box.

```
add_filter( 'swc_pro_button_style_metabox_title', 'my_button_title'

function my_button_title( $title ) {
    return 'Chat Button Appearance';
}
```

Filter Hook: swc_pro_agent_availability_meta_values

Filters agent availability meta values before rendering the availability meta box. Allows you to modify availability data.

```
add_filter( 'swc_pro_agent_availability_meta_values', 'my_modify_av

function my_modify_availability( $availability_data, $post ) {
    // Force always online for specific agents
    if ( (int) $post->ID === 123 ) {
        $availability_data['always_online'] = 1;
    }

    return $availability_data;
}
```

Filter Hook: swc_pro_agent_post_type_args

Filters the WhatsApp Agent post type registration arguments. Allows developers to modify the post type configuration.

```
add_filter( 'swc_pro_agent_post_type_args', 'my_modify_post_type' )

function my_modify_post_type( $args ) {
    // Enable REST API support
    $args['show_in_rest'] = true;
    $args['rest_base']   = 'wa-agents';

    // Add custom capabilities
    $args['capability_type'] = 'agent';
```

```
    $args['map_meta_cap']    = true;

    return $args;
}
```

Filter Hook: swc_pro_default_agent_avatar

Filters the default agent avatar URL used when an agent doesn't have a custom avatar.

```
add_filter( 'swc_pro_default_agent_avatar', 'my_default_avatar' );

function my_default_avatar( $default_avatar ) {
    return 'https://your-site.com/wp-content/uploads/default-agent.
}
```

Filter Hook: swc_pro_woocommerce_exclude_agent_by_sku

Filters whether to exclude an agent from display on a WooCommerce product page based on product SKU. Return true to exclude the agent, false to show.

```
add_filter( 'swc_pro_woocommerce_exclude_agent_by_sku', 'my_exclude

function my_exclude_agent_by_sku( $exclude_agent, $agent_id, $produ
    // Exclude agent 123 for products with SKU starting with "PREMI
    if ( (int) $agent_id === 123 && strpos( $product_sku, 'PREMIUM-
        return true;
    }

    // Exclude agent for specific product categories
    if ( $product instanceof WC_Product && has_term( 'wholesale', '
        return true;
    }

    return $exclude_agent;
}
```

Filter Hook: swc_allowed_visibility_post_types

Filters the allowed post types for visibility settings. Allows you to add custom post types to the visibility configuration.

```
add_filter( 'swc_allowed_visibility_post_types', 'my_add_post_types

function my_add_post_types( $allowed_post_types ) {
    // Add custom post types
    $allowed_post_types[] = 'product';
    $allowed_post_types[] = 'event';

    return $allowed_post_types;
}
```

Filter Hook: swc_pro_agent_online_status

Filters agent online status. Provides a single extension point for developers to override or modify availability logic (holidays, CRM, AI routing).

```
add_filter( 'swc_pro_agent_online_status', 'my_modify_online_status

function my_modify_online_status( $status, $agent_id ) {
    // Check if agent is on holiday
    $is_holiday = check_if_holiday( $agent_id );
    if ( $is_holiday ) {
        return 0; // Force offline on holidays
    }

    // Check CRM for agent availability
    $crm_status = get_agent_status_from_crm( $agent_id );
    if ( $crm_status !== null ) {
        return $crm_status ? 1 : 0;
    }

    // Use default status
    return $status;
}
```

7. Change Log

Version 1.0.0

- Initial release
- Multi-agent WhatsApp support
- Floating widget
- WooCommerce integration
- Analytics tracking
- Gutenberg block support

8. Support

If you need assistance or have questions about Smart WP Chat Pro, please send an email to support@magexweb.com for help.

We offer support through email. Please provide detailed information about your issue to help us assist you effectively.

Thank you for choosing Smart WP Chat Pro!

© Smart WP Chat Pro – Documentation