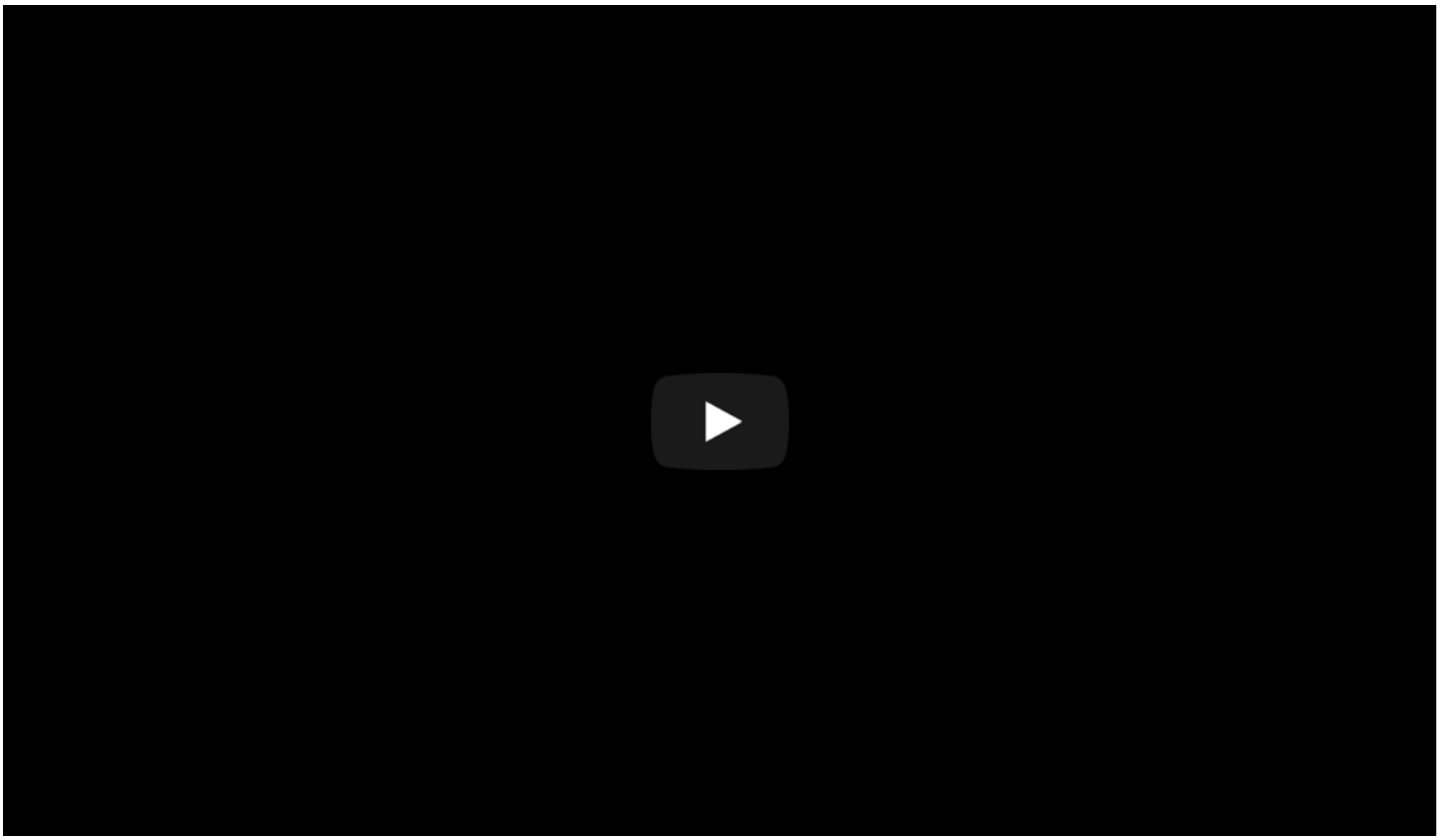
Getting Started

Overview

For more details go to our on-line documentation

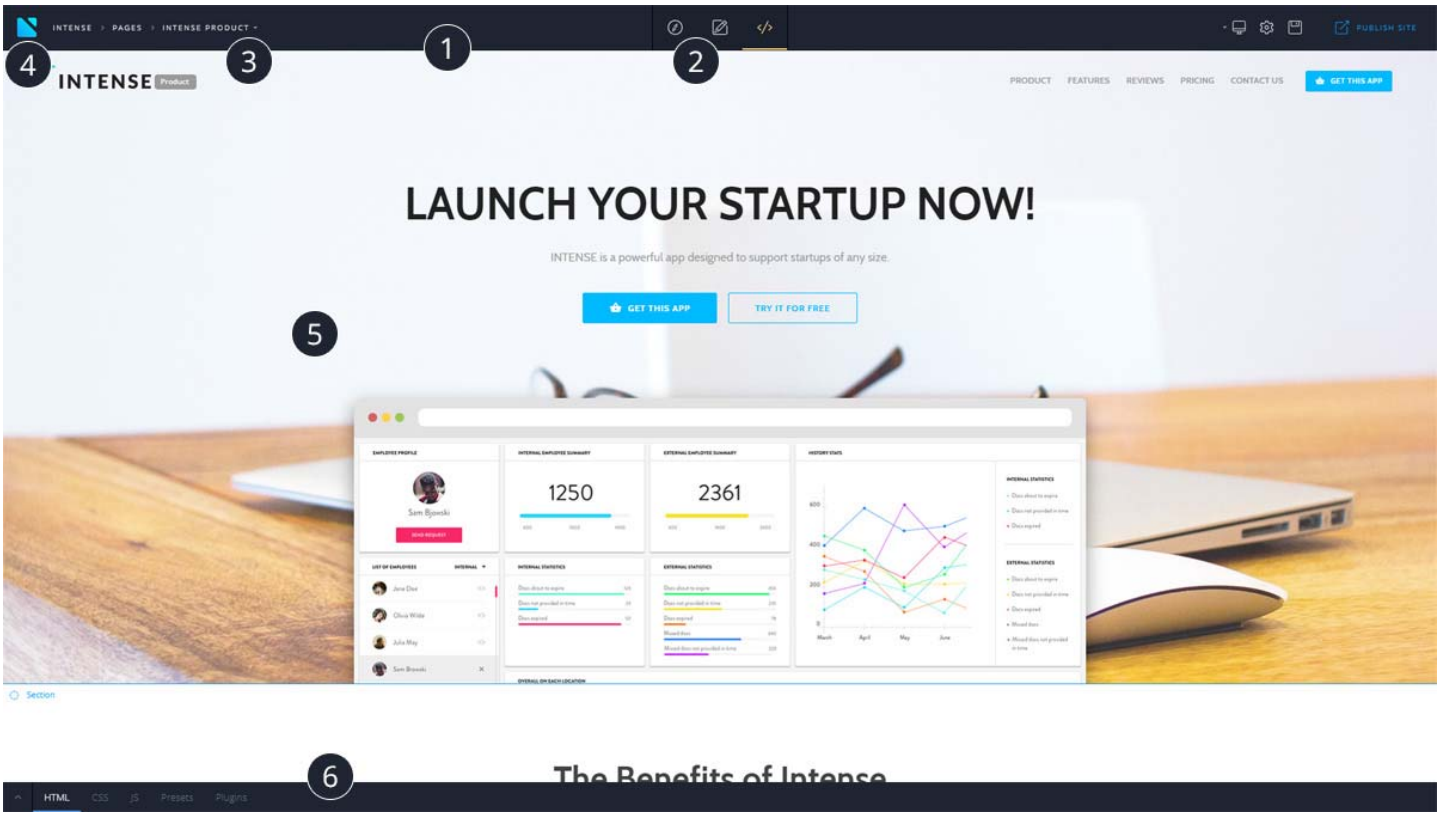
The brief overview of the Novi Builder plugin.



Welcome to Novi Builder!

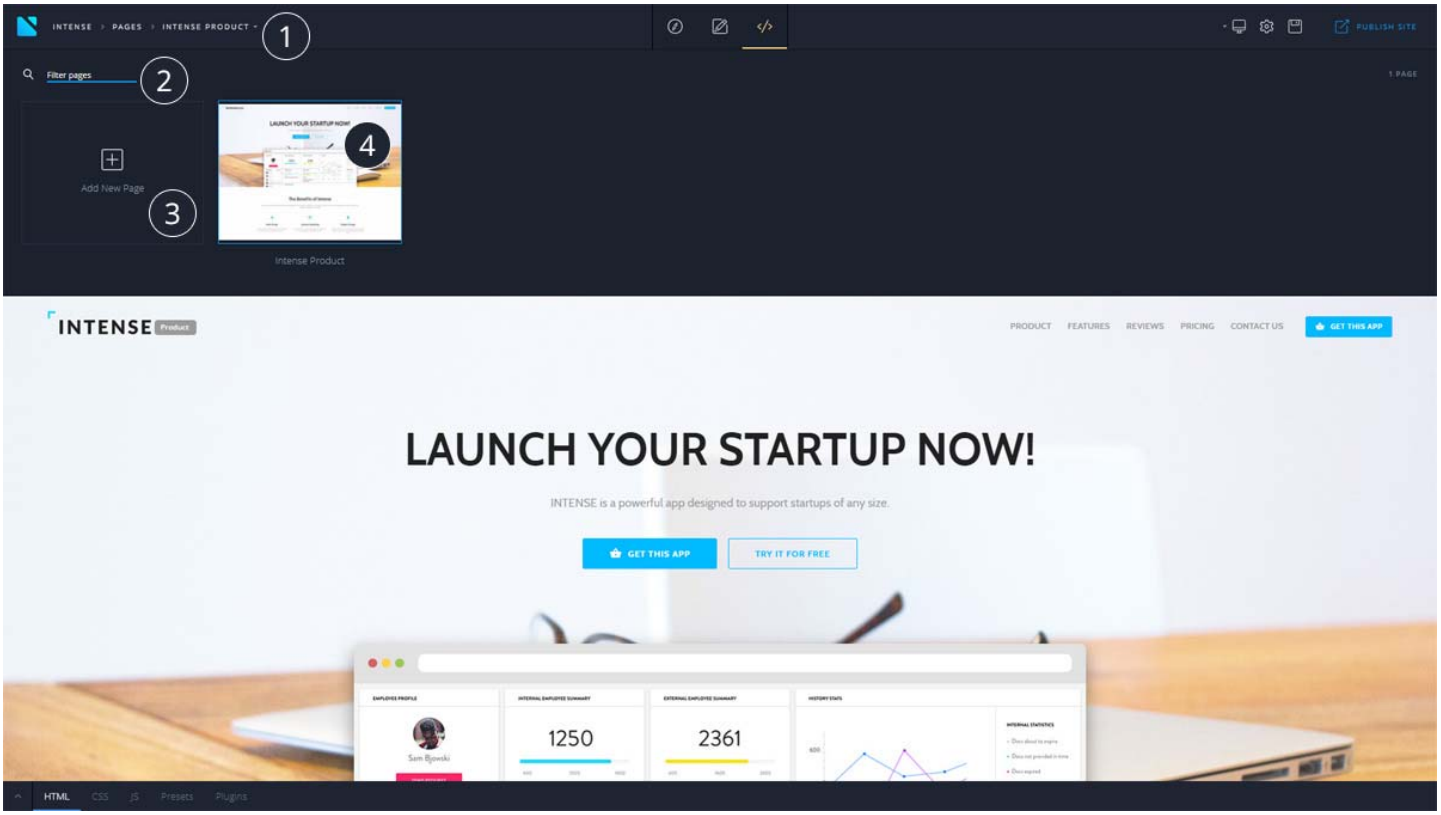
Novi interface:

1. Main panel
2. Novi modes switcher
3. Page Manager
4. Main menu
5. Visual Editor area
6. Developer's toolbar



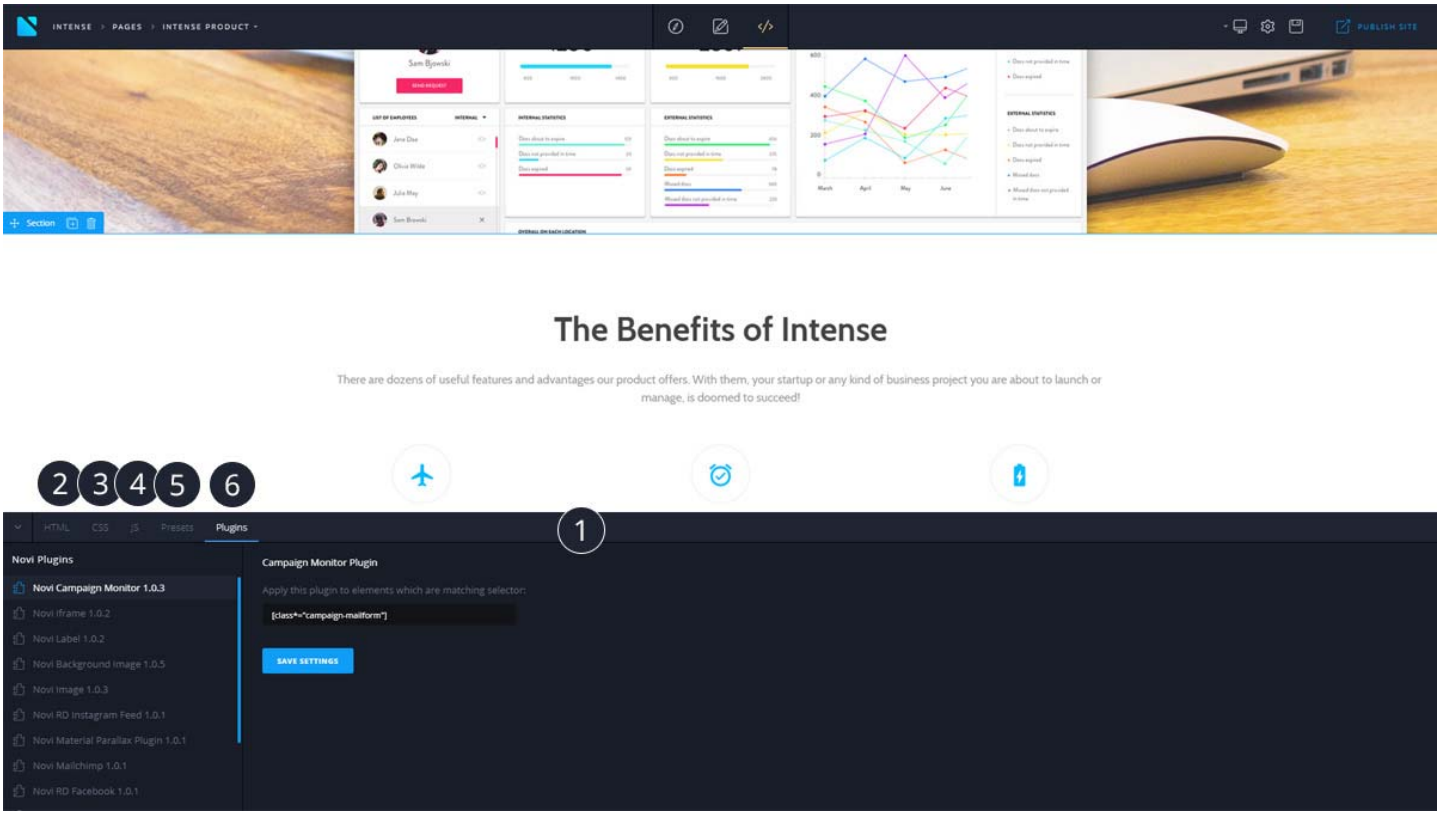
Working with pages

1. Page manager trigger
2. Pages Search filter
3. Adding a page
4. Existing pages list



Developer's toolbar

1. Developers tools area
2. HTML code editor
3. CSS code editor
4. JS code editor
5. Presets - presets manager
6. Plugins - plugins manager



System Requirements

To ensure comfortable work with Novi Builder, your system should meet the following requirements:

Browser: Chrome 50+ or Safari 10+
4GB RAM or more
Display with minimal resolution of 1024 x 768
CPU with minimal clock rate of 1GHz
Keyboard and mouse
The minimum speed of the Internet network connection - 1 Mbit.

Package Structure

After extracting Novi builder package you'll see the following files structure:

documentation/

demo/

- builder/
  - config - builder configurations files folder
  - plugins - builder external plugins folder
- ...Locally published website files

Installation

Installing on a local server

You can install Novi Builder on a local server in a few simple steps:

1. Download and install the software to launch your local server, e.g. WAMP, MAMP, or XAMP.
2. Set up the local server according to the requirements of "Web server requirements" section.
3. Extract the contents of the compressed Novi Builder archive into one of local server directories, e.g. your-disk/wamp/www/novi/.
4. Launch Novi Builder using the link, which corresponds to the local server path, e.g. http://localhost/novi/

Installation on a remote server

To install Novi Builder on a remote server, you need to:

1. Perform remote server installation according to the requirements described in "Web Server requirements" section.
2. Download and install the software for work with FTP, e.g. Total Commander 7+, FileZilla etc.
3. Upload the contents of the Novi Builder archive to your hosting server. Let us consider the example of Total Commander:
   - Establish the connection with the server:
     - Navigate to the tab Net->FTP connect... (Ctrl + F)
     - Press the "New connection..." button
     - Fill in the fields, specifying the data given you by your hosting provider:
     - Session (the name of your connection, not granted by the provider, specify any suitable name)
     - Host name[Port]:,
     - User name:,
     - Password
   - Press "OK"
   - Upload the necessary files to the server:
     - Navigate to the tab Net->FTP connect... (Ctrl + F)
     - Select the previously created connection name, "Session" field
     - Press "Connect ..."

Copy Novi Builder archive contents
Navigate to the proper directory (root directory of your domain name by default).
Paste Novi Builder archive contents.

Web Server Requirements

To ensure the correct work of Novi Builder, your server should meet the following requirements:

Apache 2.2+
PHP 5.4+

Apache Settings

Turn off ModSecurity module or other security module
Turn on mod_rewrite module

PHP Settings

php_memory_limit = 64M
upload_max_filesize = 64M
post_max_size = 64M
max_input_time = 60
php_max_input_vars = 1000
max_execution_time = 30
short_open_tag = 1
error_reporting = E_ALL ^ E_WARNING

Features

Page Manager

How to add a page

Open the builder page.
Press "Open page manager", which will appear on hovering over the breadcrumbs to the right of Novi logo.

Press "Add new page" on the appeared panel.

Enter the page name and press "Create page".

To save changes done to the project, you need to press the "Save changes" button, located in the upper right part of the interface.

How to switch a page

Novi interface:

Open the builder page.

Press "Open page manager", which will appear on hovering over the breadcrumbs to the right of Novi logo.

Select the proper page by pressing LMB on the appeared panel.

How to delete a page

Open the builder page.

Press "Open page manager", which will appear on hovering over the breadcrumbs to the right of Novi logo.

Press "Delete" on the appeared panel.

To save changes done to the project, you need to press the "Save changes" button, located in the upper right part of the interface.

How to duplicate a page

Open the builder page.

Press "Open page manager", which will appear on hovering over the breadcrumbs to the right of Novi logo.

Press "Duplicate" on the appeared panel.

To save changes done to the project, you need to press the "Save changes" button, located in the upper right part of the interface.

How to add the page preview

Open the builder page.

Press "Open page manager", which will appear on hovering over the breadcrumbs to the right of Novi logo.

Press "Upload preview" on the appeared panel.

Press "Choose file" in the appeared modal window.

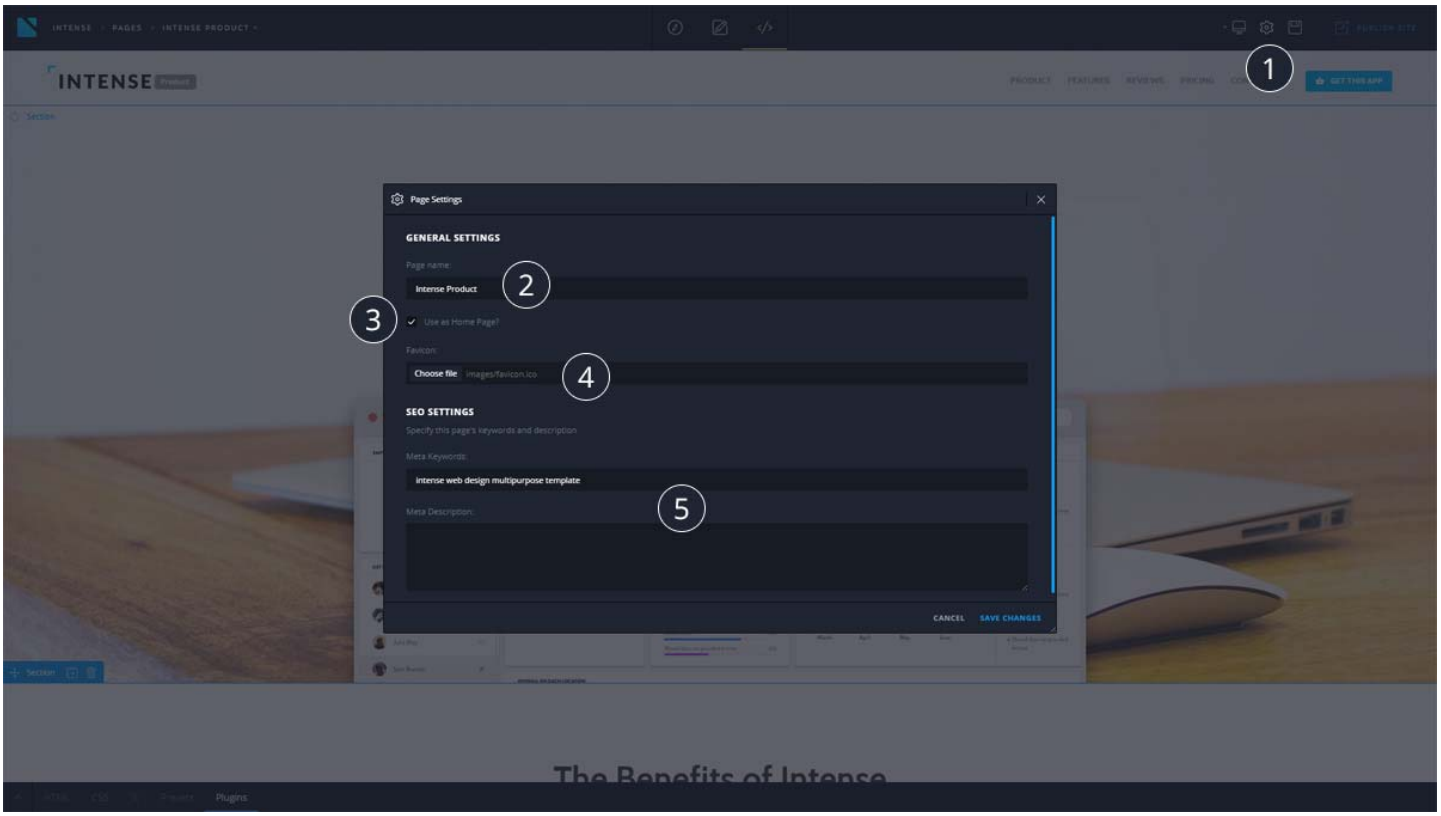
Press "Upload Image

To save changes done to the project, you need to press the "Save changes" button, located in the upper right part of the interface.

Page Settings

Page settings will be applied only to the page, opened in the visual editor at the moment.

1. Trigger for the current page settings display
2. Page name - the title of the page
3. Use as home page - set the current page as ‘front’ one
4. Favicon - and option to upload the favicon of the website
5. Page’s metadata



Press the "Save changes" button after you’re done with the settings. Then save the project by pressing the "Save project" button.

Design Mode

This mode is mainly intended for developers. It provides you with the following options:

Configure the builder’s plugins
Edit the source code of the page (HTML, CSS, JS) with the help of the code editor tool
Manage the project’s presets
Manage the page’s content using the visual editor

Edit Mode

This mode is mainly intended for the end-users. It allows you to use all the visual tools for the content editing, including the builder’s plugins. "Edit Mode" is the one that loads by default.

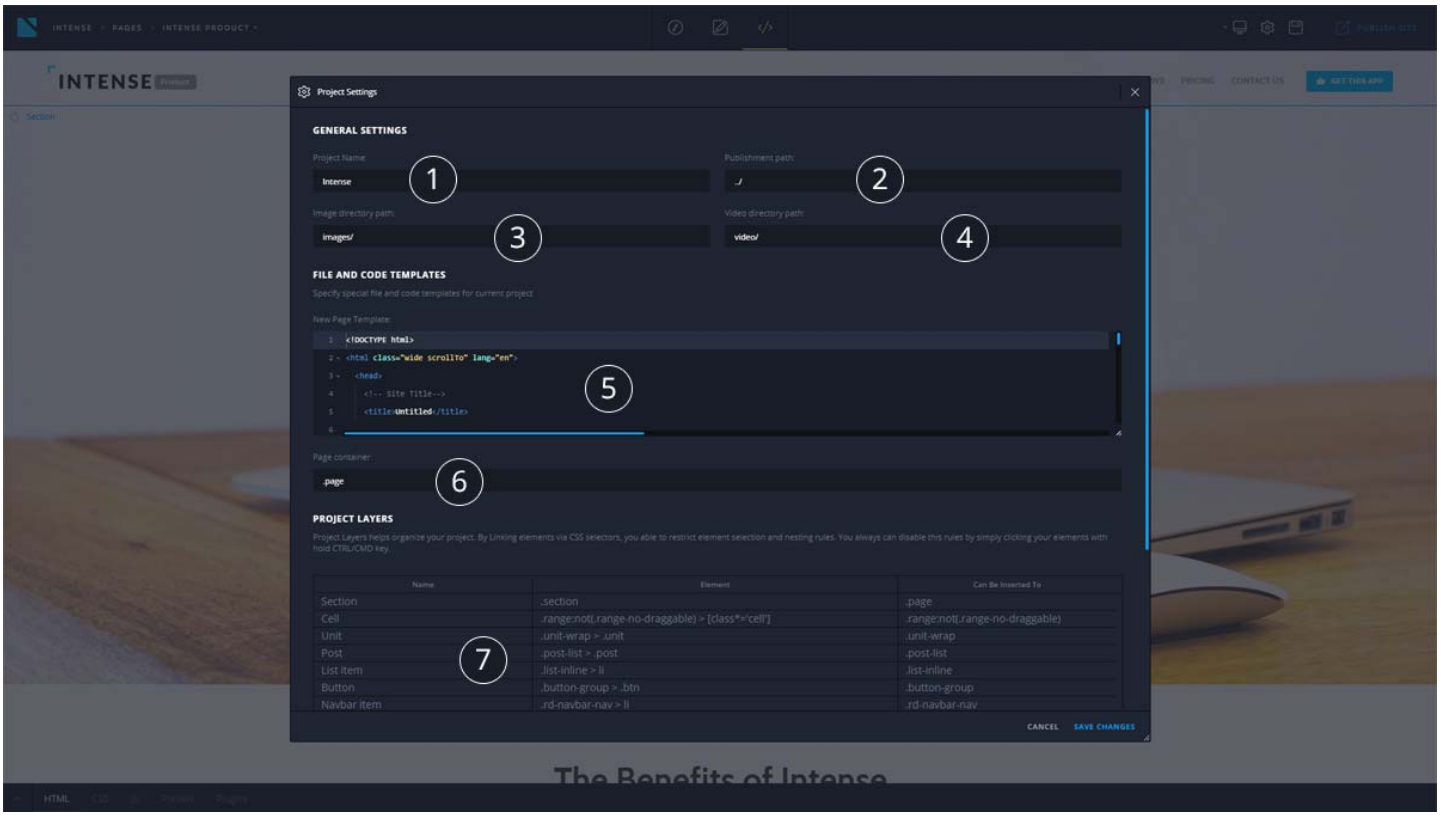
Preview Mode

This mode allows to preview your website.

Project settings (Layer tool)

You can access the project’s setting from the main builder menu directly (via the builder interface).

1. Project Name - the title of the project
2. Publishment path - the path to the published version of the project on your server
3. Image directory path - path to the directory that contains project image files.
4. Video directory path - path to the directory that contains project video files.
5. New Page Template - basic mark-up of the new page
6. Page container - page’s container
7. Project Layers - layers’ system setup



Layer creating rules

To detect the user's interaction with the project layout elements, Novi uses the layer system. A layer is a rule, which defines the logical group of layout elements for manipulations in the visual editor (dragging, pasting, deleting, duplicating, transporting, and others).You can also manipulate any page elements disregarding layers as described in “Extending the features of working with elements”.

How to add/change a layer

To add a layer, you need to:

Open the builder page.

Open the main menu (press the Novi Builder logo icon) located in the upper left corner of the interface.

Select "Project configuration".

In the "Project Layers" section, press "+" in the bottom of the table.

Fill in the fields according to the "Rules of creating layers" section.

Press "Save changes".

To save changes done to the project, you need to press the "Save changes" button, located in the upper right part of the interface.

Rules of creating layers

When creating layers, you need to enter the following parameters:

Name – a name, which will be displayed on hovering over an element in the visual editor.

Element – a CSS selector that defines the logical group, to which this rule applies.

Can be inserted to – a CSS selector determining an area of manipulations with the logical group.

Example of layer usage
Let's take as an example an empty project without any layers. After entering the "Edit mode", you don't have the ability to interact with layout elements, and you need to add a layer.

Create a layer according to the paragraph "How to add or change a layer" with the following parameters:

Name - Section

Element - .section

Can be inserted to - body

Thus, a layer called "Section" is created, which will contain all elements of HTML layout with .section class. All manipulations (dragging, adding etc.) will be happening in the context of its parent - .

After saving the layer in "Edit mode" (when elements with .section class are present on the target page), and while hovering over the layout element, which corresponds to the created layer, for example:

```
<section class="section">
    ...
</section>
```

an editor panel will be shown with options of dragging, duplicating, and deleting.

The area of dragging, in this case, will be limited to the "body" tag listed in the parameters of Section layer, i.e. you will not be able to move an element with .section class outside the borders of , for example to

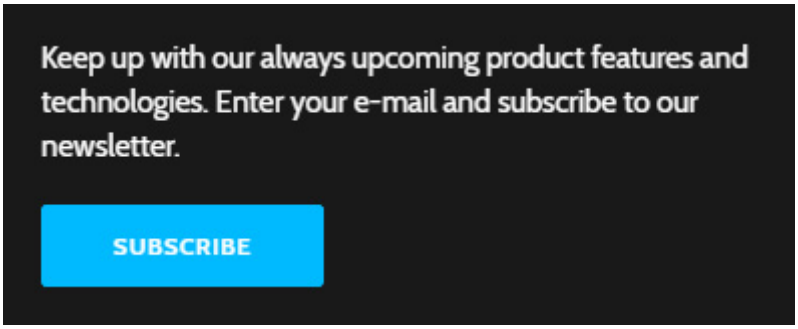
.

Also if there are saved presets in the project (how to work with them is described in the "Working with Presets" section), after hovering over an area where presets (HTML code specified in them) can be applied according to layers, a module of adding presets will be displayed.

Extending the features of working with elements

Novi uses the layer system for visual editing of elements.
If you need to edit elements that are not described in the layer system, you can use Ctrl button in Windows OS or Cmd in Mac OS.

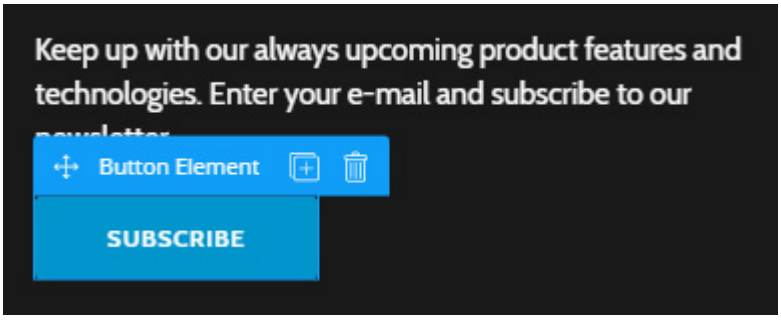
When this key is pressed, you get access to all elements of the current page. Let us consider the following example:



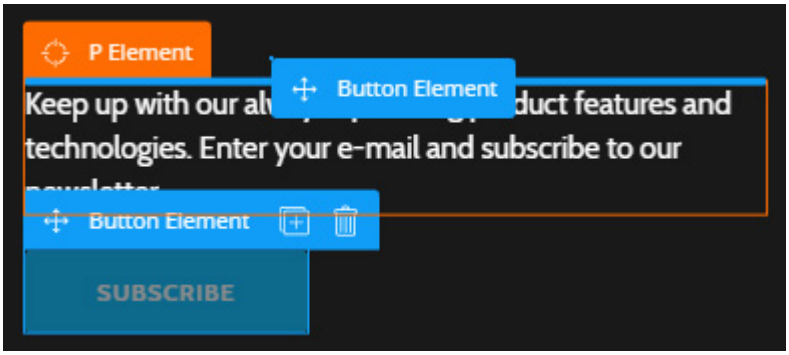
Here we need to place a button before the text.

If “Design Mode” is enabled, then:

Press and hold Ctrl on Windows OS or Cmd on Mac OS
Select the button by clicking LMB
Click and hold LMB on the appeared panel

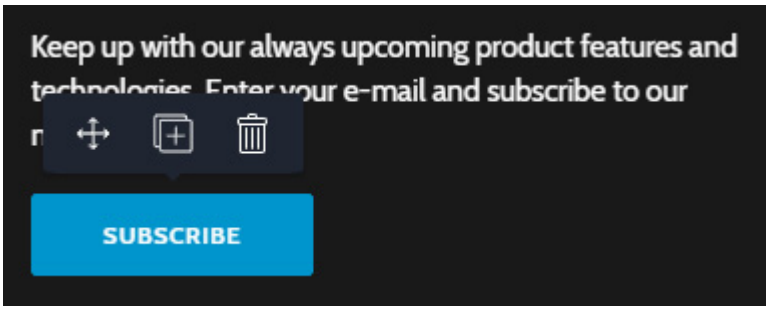


Extending the features of working with elements

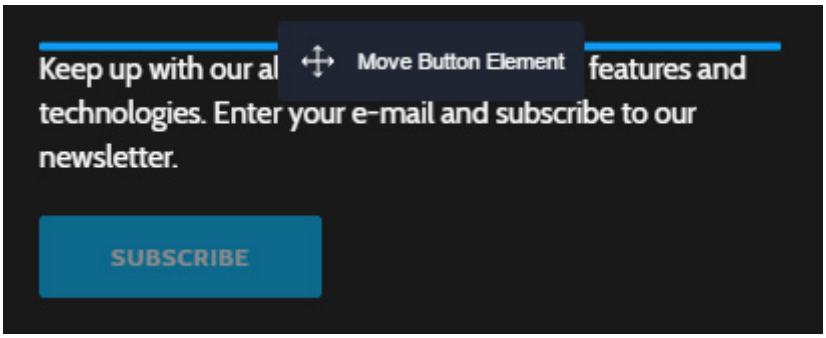


Release LMB and Ctrl on Windows OS or Cmd on Mac OS
If “Edit Mode” is enabled:

Press and hold Ctrl on Windows OS or Cmd on Mac OS, then hover over a button
Click and hold LMB on the icon of the appeared panel



Placing an element before the text area



Release LMB and Ctrl on Windows OS or Cmd on Mac OS.

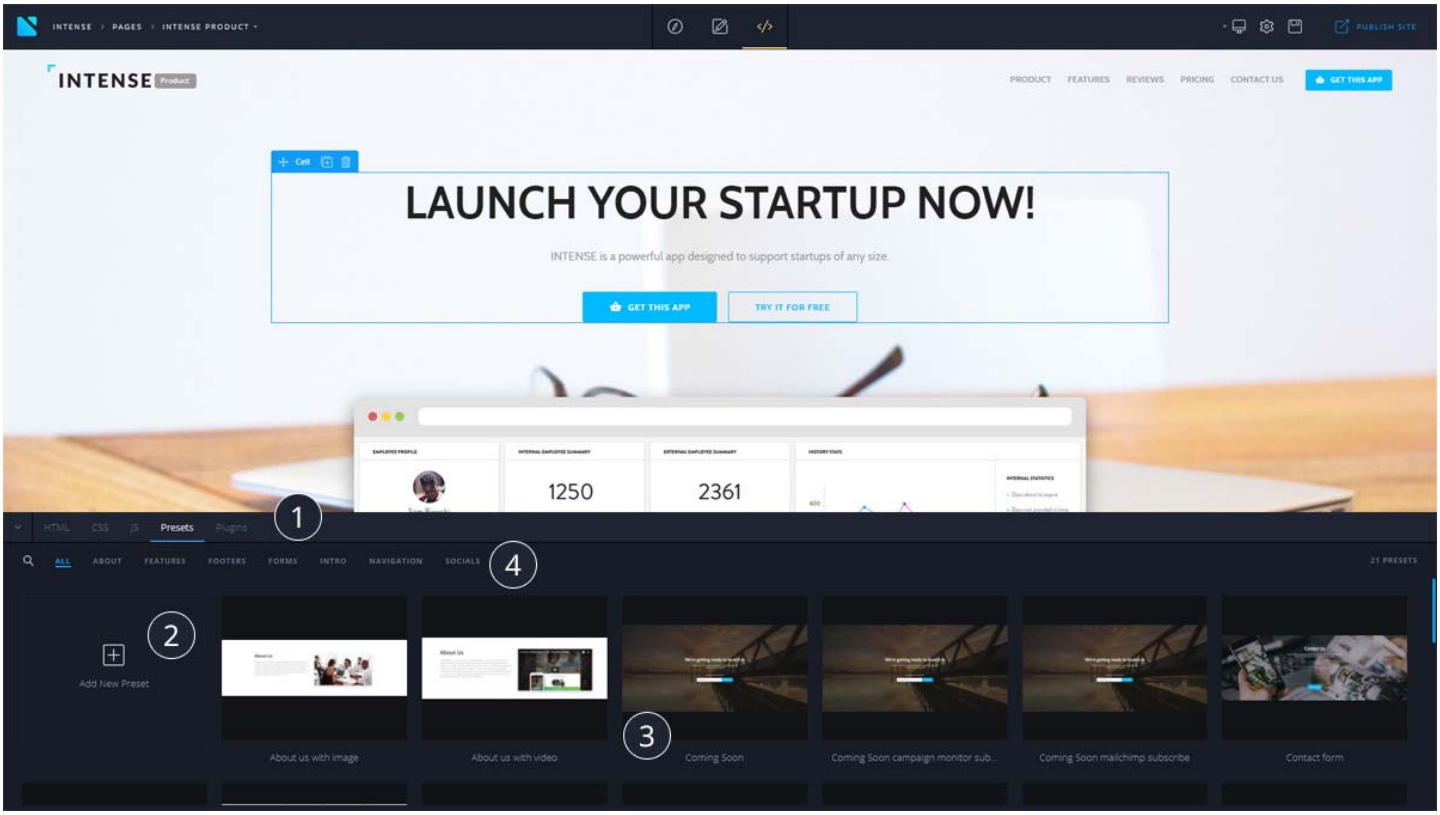
Please pay your attention! This method allows you to manipulate any layout elements. Follow the rules of HTML code validity and semanticity.

Presets tool

A preset is a separate independent part of an HTML page. Actually, it is a set of HTML tags with its own structure, CSS classes, and nesting. Any part of HTML page may act as a preset but for the convenience, it is better to store independent components of a page in presets, for example, a banner, a section, a form, a column, a sidebar etc. You can learn more about working with presets in this section below. [#br] Before starting to work with presets, we recommend you to read the section "Rules of creating layers".

Presets panel includes the following items:

1. Developer’s panel
2. ‘Create a preset’ option
3. Existing presets list
4. Presets categories and a search bar



How to create a preset

There are 2 ways of creating a preset:

Create a preset using the preset panel

Use the context menu of visual editor

In the first case you need to:

Open the builder page.

Go to the "Design Mode" by pressing a proper button in the middle of the upper panel.

Open the bottom panel by pressing "Presets".

On the appeared panel, press "Add new preset".

Fill in the fields as shown in the paragraph "How to correctly set the parameters when creating a preset" of this section.

Press "Save preset".

In the second case:

Open the builder page.

Go to the "Design Mode" by pressing a proper button in the middle of the upper panel.

Select the necessary area.

Open the context menu by pressing RMB.

Choose "Save as preset".

Fill in the fields as shown in the paragraph "How to correctly set the parameters when creating a preset" of this section.

Press "Save preset".

In the first case, you need to set the preset layout. In the second case, the layout will be taken from the element you selected in the "HTML Code" text field.

How to correctly set the parameters when creating a preset

In the "Preset name" field, specify the preset name.[#br] You also need to specify keywords in the "Keywords" field that allows finding the proper preset. We recommend you to state the preset keys relying on the content written in them. For example, the section with our clients will have the key named "Clients" or "Partners".

The text area "HTML Code" displays the HTML code, which will be inserted into the selected page after the preset is applied.

Checkbox "Deploy on preset insertion" defines whether the page will be reloaded when the preset is inserted.

How to add a preset preview

Open the builder page.

Go to the "Design Mode" by pressing a proper button in the middle of the upper panel.

Open the bottom panel by pressing "Presets".

On the appeared panel, select "Upload preview" from the context menu of the target preset.

Press "Choose file" in the opened modal window.

Press "Upload Image"

How to edit a preset

Open the builder page.
Go to the "Design Mode" by pressing a proper button in the middle of the upper panel.
Open the bottom panel by pressing "Presets".
On the appeared panel, select "Edit" from the context menu of the target preset.

How to duplicate a preset

Open the builder page
Go to the "Design Mode" by pressing a proper button in the middle of the upper panel.
Open the bottom panel by pressing "Presets".
On the appeared panel, select "Duplicate" from the context menu of the target preset.

How to delete a preset

Open the builder page
Go to the "Design Mode" by pressing a proper button in the middle of the upper panel.
Open the bottom panel by pressing "Presets".
On the appeared panel, select "Delete" from the context menu of the target preset.

Adding the preset to the page

There are 2 ways to add a preset to the page:

With the help of the preset panel.
Using the preset insertion module.

In the first case you need to:

Open the builder page
Go to the "Design Mode" by pressing a proper button in the middle of the upper panel.
Open the bottom panel by pressing "Presets".
On the appeared panel, find the necessary preset.
Drag the preset into the visual editor area by holding LMB, therewith the area of insertion will be highlighted by a solid blue line.
In the second case:
Open the builder page
Go to the "Edit Mode" by pressing a proper button in the middle of the upper panel.
Hover over the area where you need to add the preset.
If the preset is present and if it meets the layer rules, "+" will be displayed, press it.
Select a necessary preset in the appeared dialogue window.

Content Editor (basic tools)

You can use the page editing tools only on the items located on the visual editor working area (see the "Overview" section). Access to the items in the visual editor area is defined at the project's layers system settings (see the "Project Settings" section).

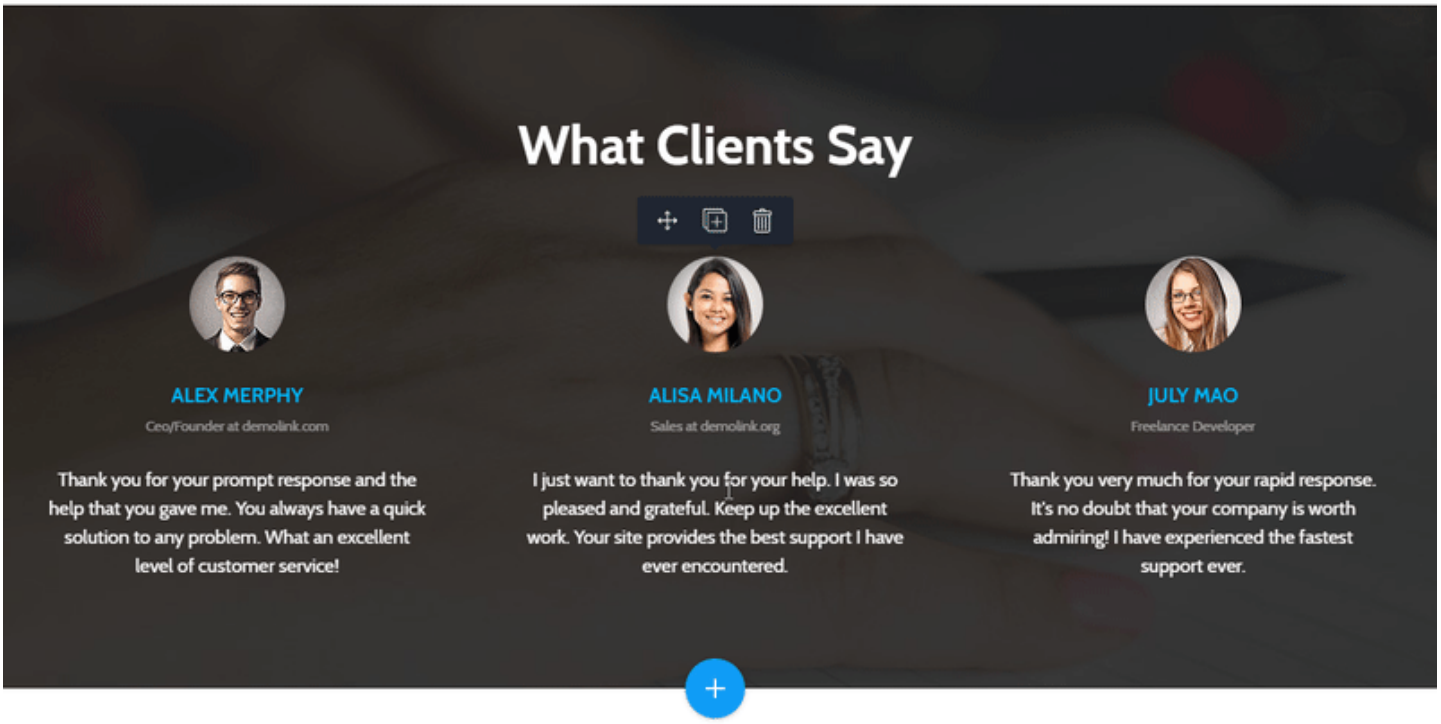
The basic content editing operations available at the visual editor are the following:

- Removing
- Pasting from a preset
- Duplicating
- Replacing

All of these options are available on the item hover - just place the mouse cursor on the item in the visual editor.

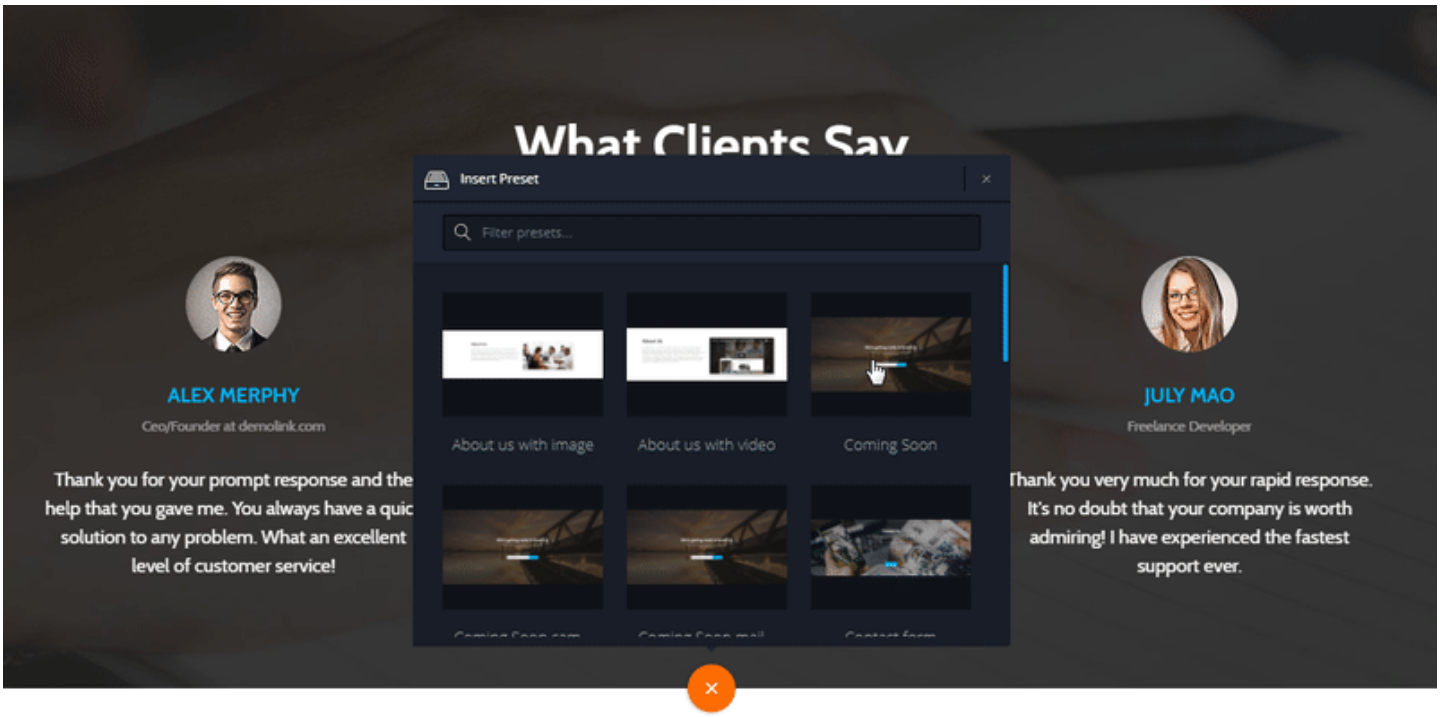
Removing the content

Hover the target item in order to delete some of the page's content and press the "Remove layer" button.



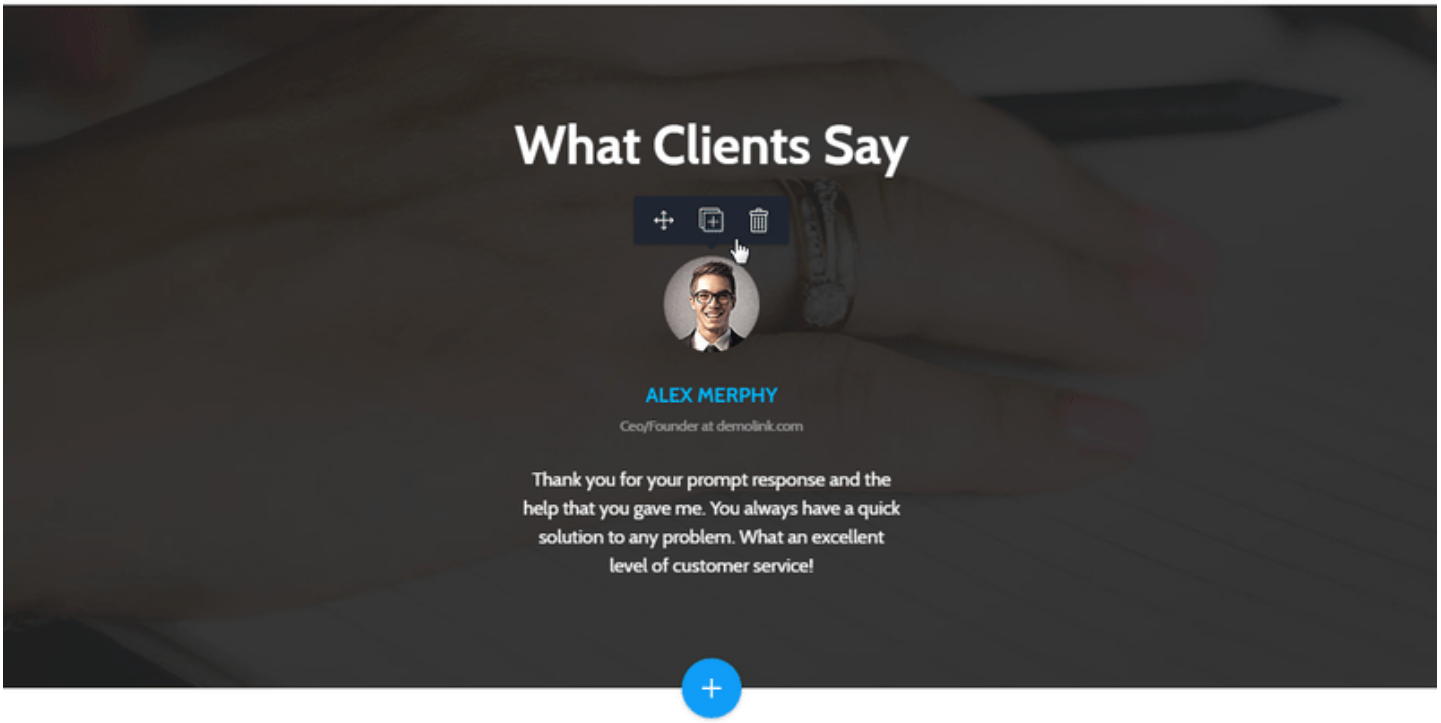
Pasting the content from a preset

Hover the target area in order to add some content from a preset. In case there is a preset, matching the layers adding rules requirements (listed in the "Project Settings" section), you'll see the '+' icon. Press it and choose the preset among the list of the available ones.



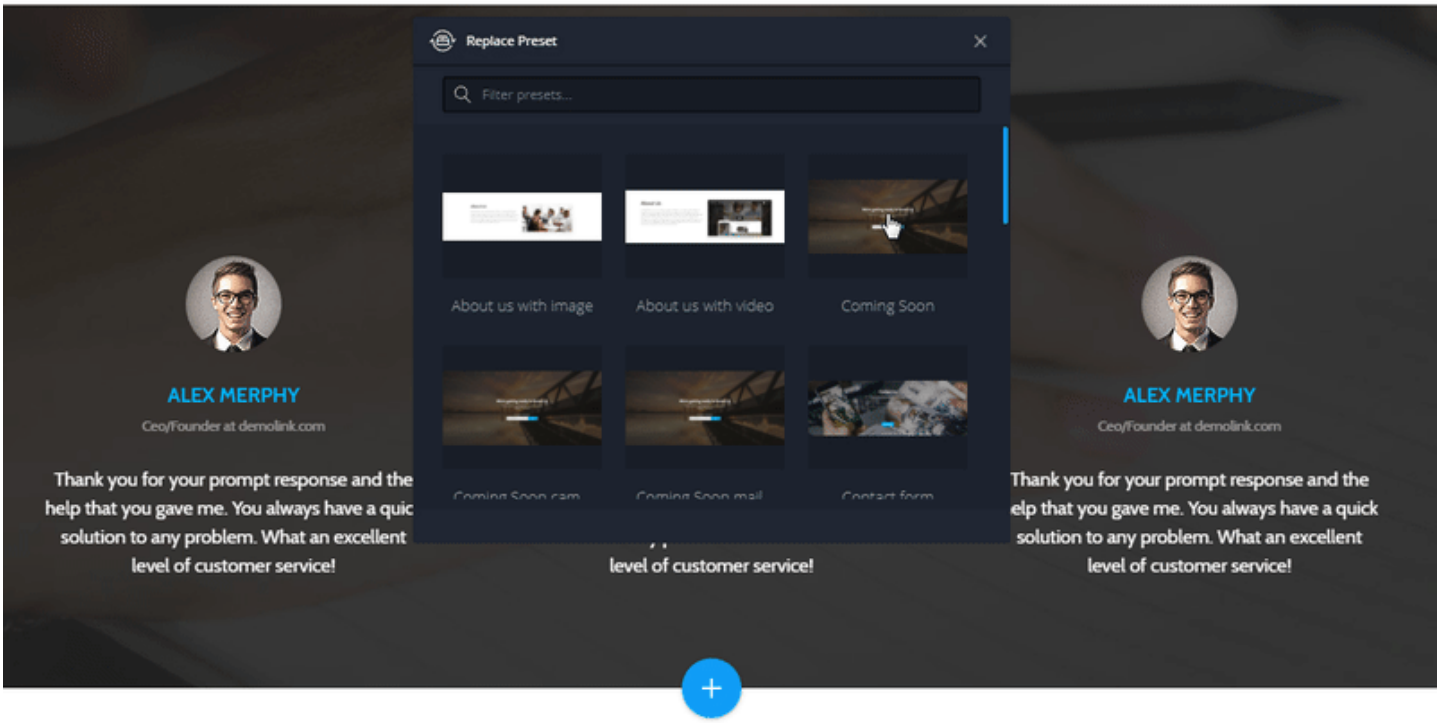
Duplicating the content

Hover the target item in order to duplicate some of the page's content and press the "Duplicate layer" button.



Replacing the content

Hover the target area in order to replace some of the page's content. In case there is a preset, matching the layers adding rules requirements (listed in the "Project Settings" section), you'll see the 'Replace layer' icon. Press it and choose the preset among the list of the available ones.

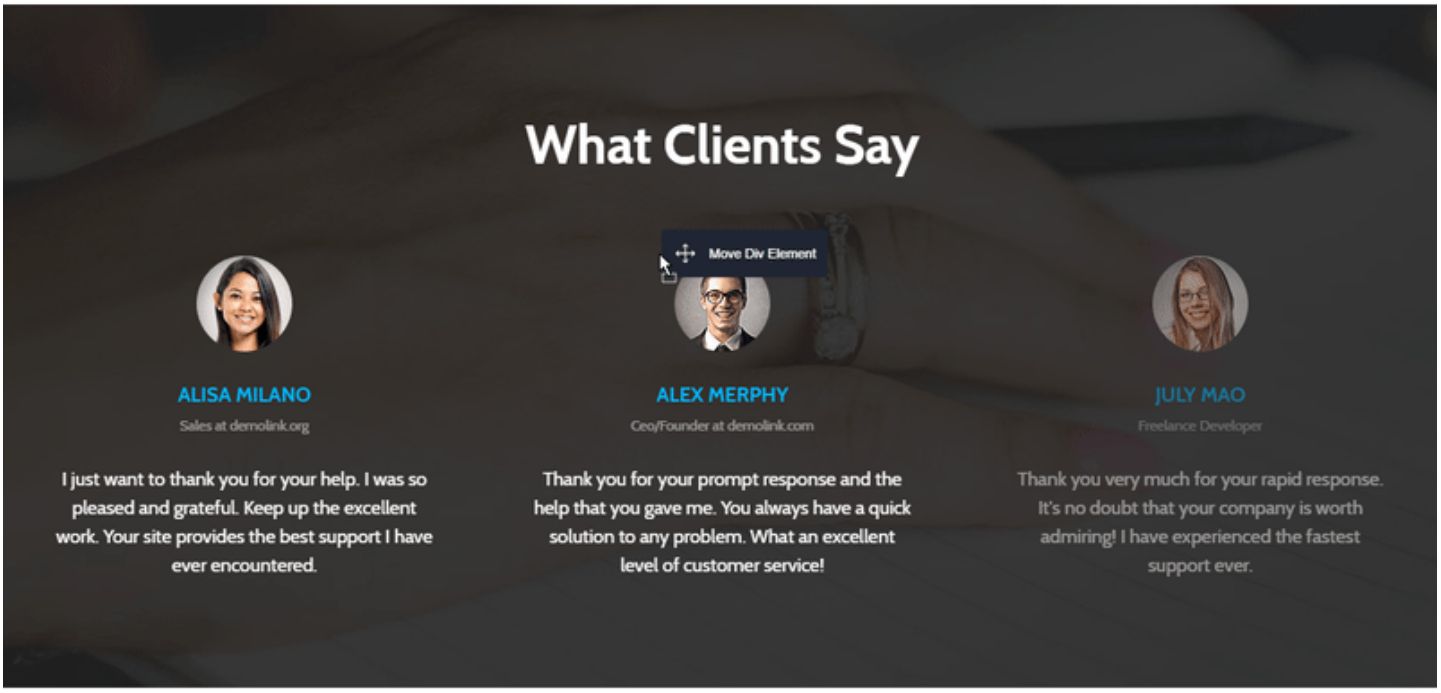


Drag & Drop tool

Drag & Drop Tool allows you to re-arrange the items within the visual editor area.
The items dragging rules are described in the Novi layers system in the "Project Settings" section, the dragging area is specified at the Projects Settings under the 'Page Container'.

First, you should hover the target item within the visual editor area, click on the "Move Layer" and then drag the item to any visual editor area keeping the left mouse button pressed.

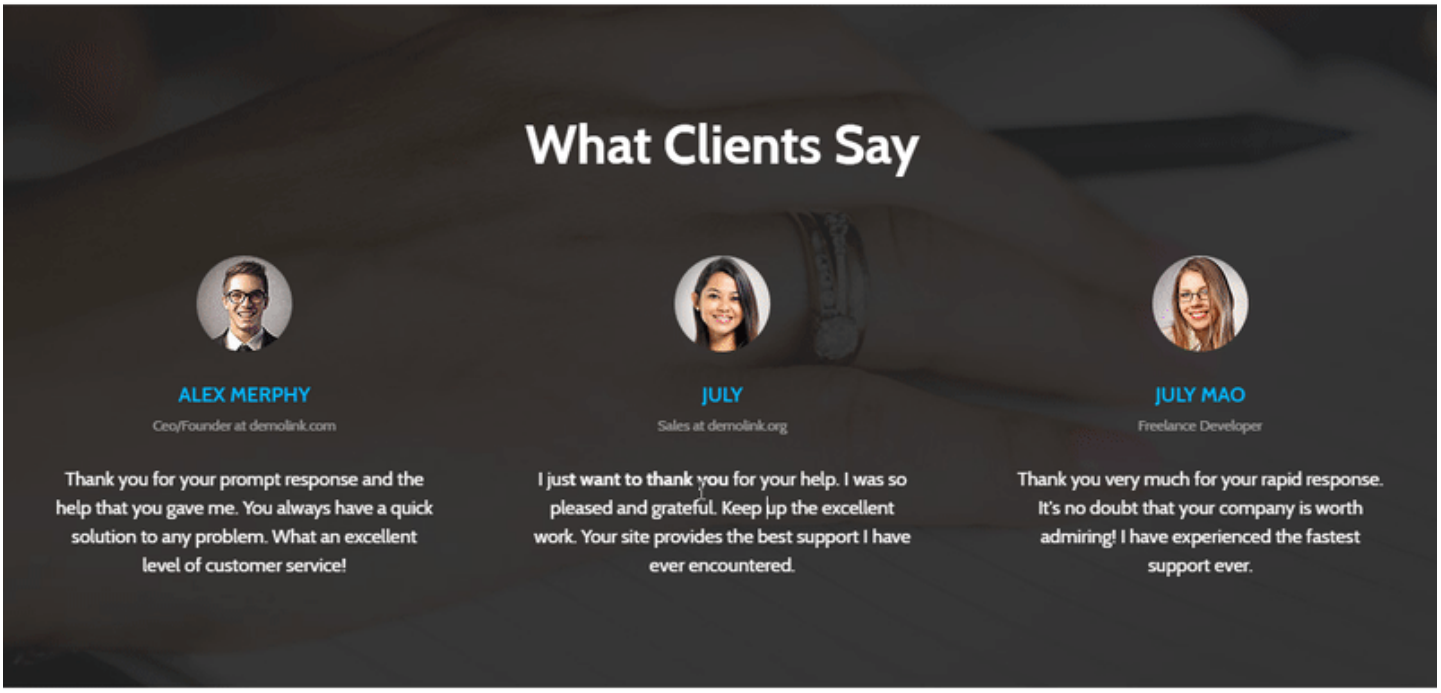
The areas, available for dragging will be pointed up with the solid blue border line.



Text Editor

Text editor allows you to edit the text within the visual editor area.
It provides you with the following options:

- Make the text bolder
- Change the text to italic
- Underline the text
- Link the text
- Change the text color
- Change the text size
- Change the height of the line
- Change the font

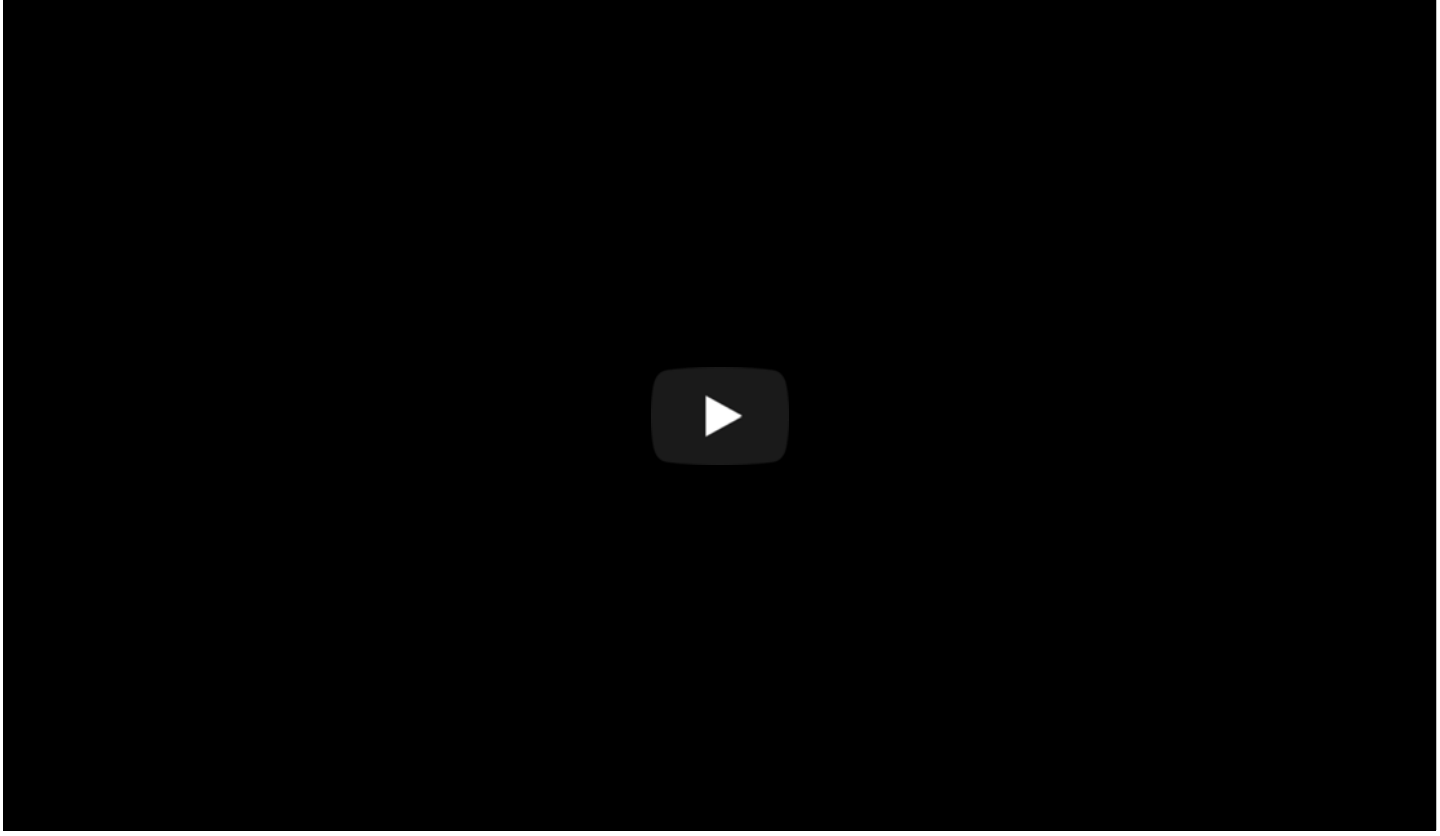


Code Editor

Novi code editor is available in the design mode only.

You may find the list of its features below:

- Enables the source code editing (HTML, CSS and JS)
- Is fully compatible with Emmet
- Includes the HTML, CSS validation by default
- Applies your edits right away when the "Autodeploy" is enabled
- Provides an option to switch between all the HTML, CSS, JS files used on the target page
- Highlights the source code of the items in the code editor when you press on them with the left mouse button in the visual editor for the intuitive navigation on the HTML code.



Context menu

Context menu allows you to manage the items in the visual editor.
A set of the operations performed with the help of the context menu depends on the current Novi mode.

Context menu in the "Edit mode"

Context menu in the "Edit mode" allows you to do the following:

- Cut an item
- Copy an item
- Paste before, within or after the selected item
- Remove an item
- Duplicate an item
- Check the source code of the item

In order to cut the item you should hover it with the mouse cursor, click on it with the right mouse button and choose the "Cut" operation in the context menu.

In order to copy the item you should hover it with the mouse cursor, click on it with the right mouse button and choose the "Copy" operation in the context menu.

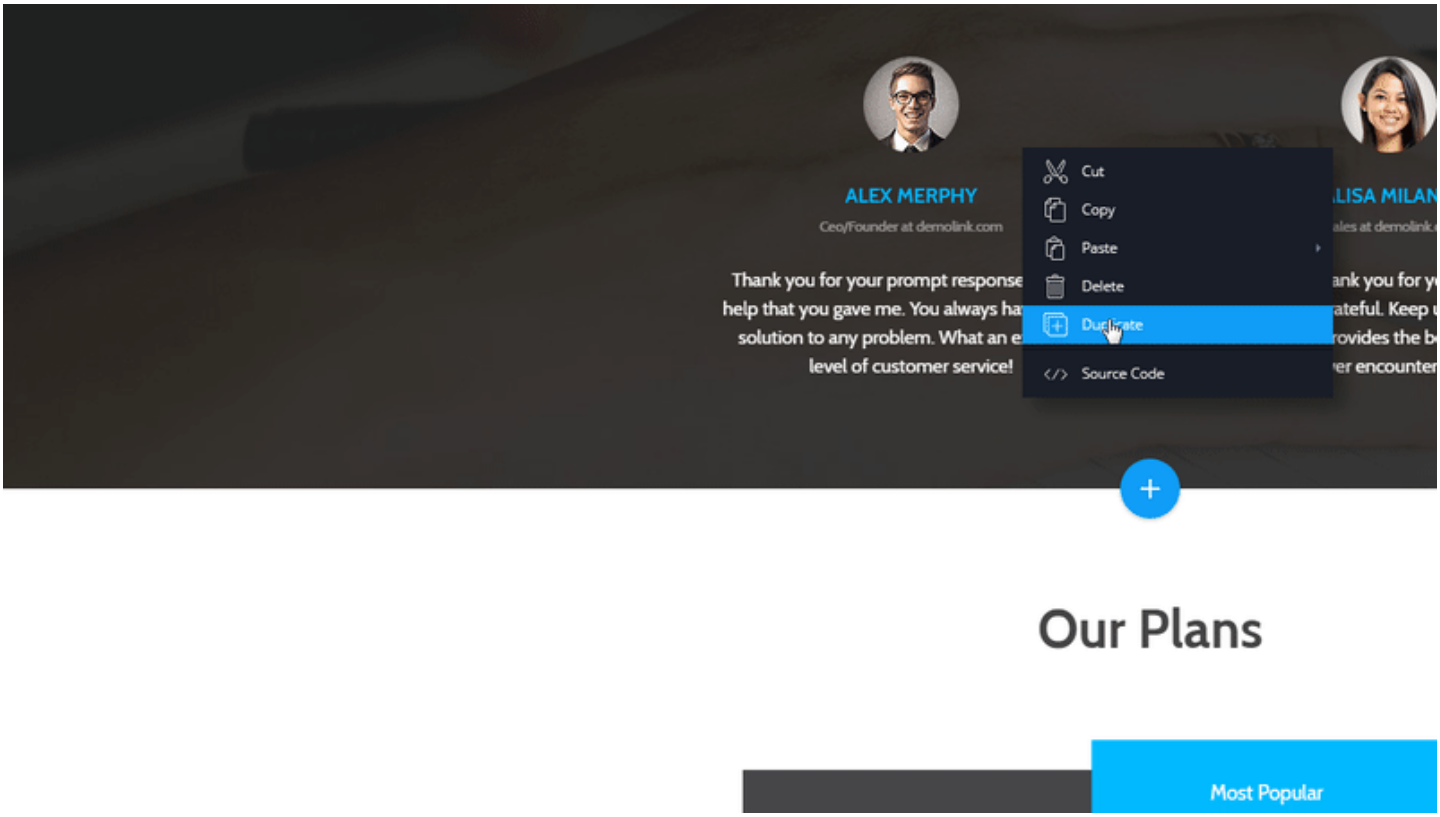
In order to paste the item that was previously cut or copied you should hover the target area with the mouse cursor, click on it with the right mouse button and choose the "Paste" operation in the context menu. You'll also need to specify the place to insert the item.

In order to remove the item you should hover it with the mouse cursor, click on it with the right mouse button and choose the "Delete" operation in the context menu.

In order to clone the item you should hover it with the mouse cursor, click on it with the right mouse button and choose the "Duplicate" operation in the context menu.

In order to check the source code of the item you should hover it with the mouse cursor, click on it with the right mouse button and choose the "Source Code" operation in the context menu. Builder will switch to the Design mode and the item's source code will be highlighted in the code editor.

Below you may find an example of using the visual editor context menu:



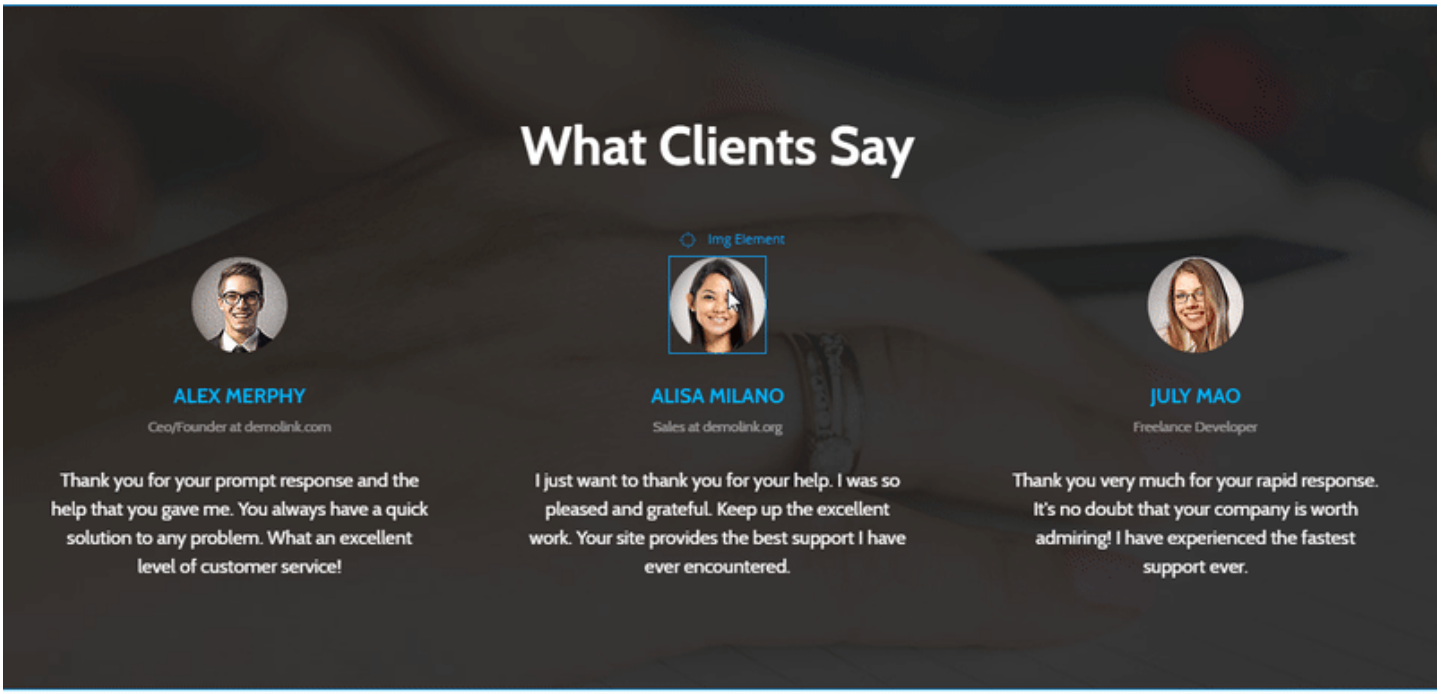
Context menu in the "Design mode"

The context menu of the Novi "Design mode" is an extended version of the one, available in the "Edit mode".

The additional options are the following:

- Select the parent item
- Save the item as a preset

Below you may find an example of using the visual editor context menu in the Design mode:



Media library

Media library contains all the graphic files that are used in the template. It allows you to easily add and delete all graphic and video files of the project.

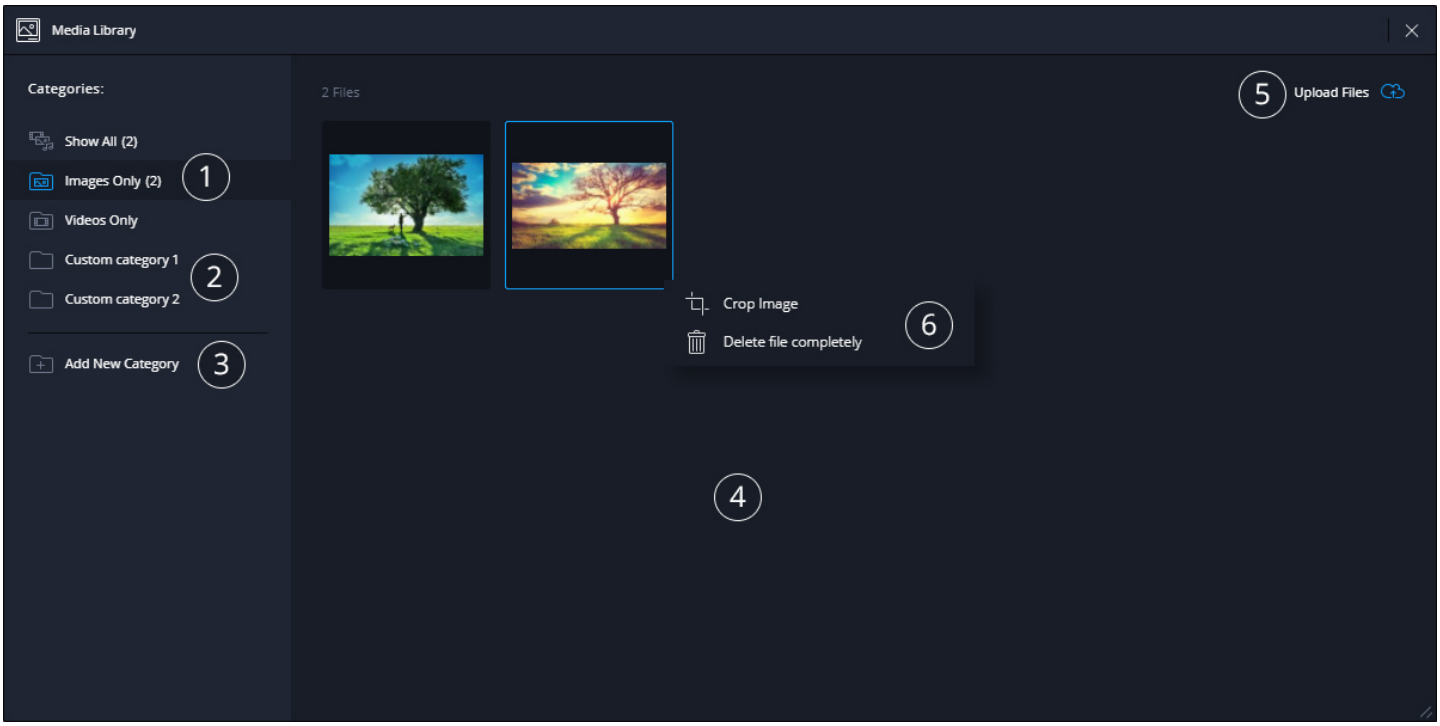
To access the media library use the main Novi Builder menu.

Media library looks the following way:

1. category panel, 3 categories are reserved by Novi ("Show all", "Images Only", "Videos Only");
2. custom project categories;
3. button for adding custom categories;
4. media panel;
5. media upload panel;
6. context menu of the media panel.

** Custom categories are those that were created by the user. Custom categories support the following functionality:*

changing category name;
category media cleanup;
category removal;
adding media files to the category.



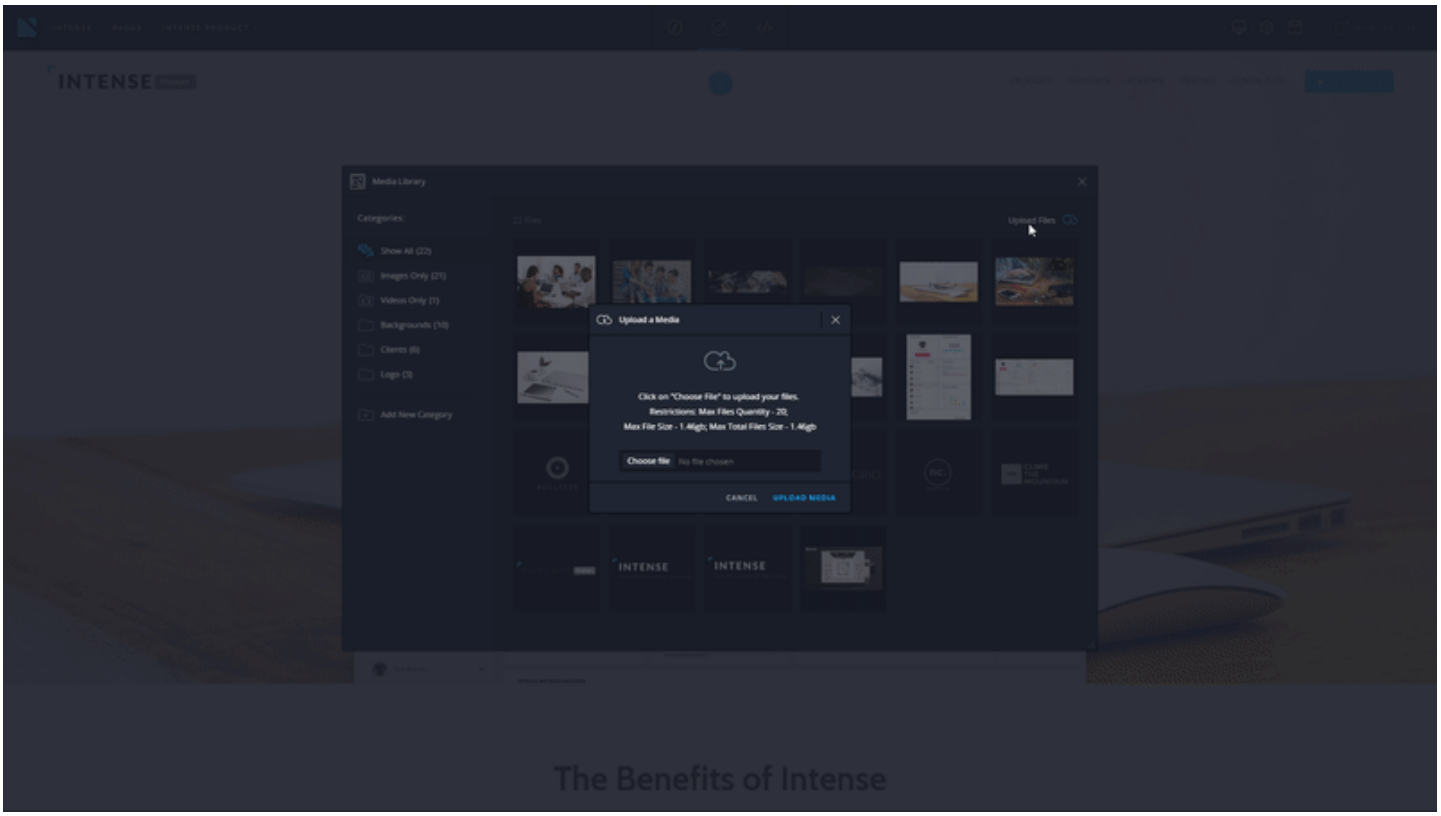
How to add media files to the library

To add a file to the media library, you should:

open the main menu of the Novi Builder;
choose "Media Library" item;
press the "Upload Files" button;
select a file, press the "Upload" button.

To save changes, use the "Save" button that resides in the right corner of the top bar.

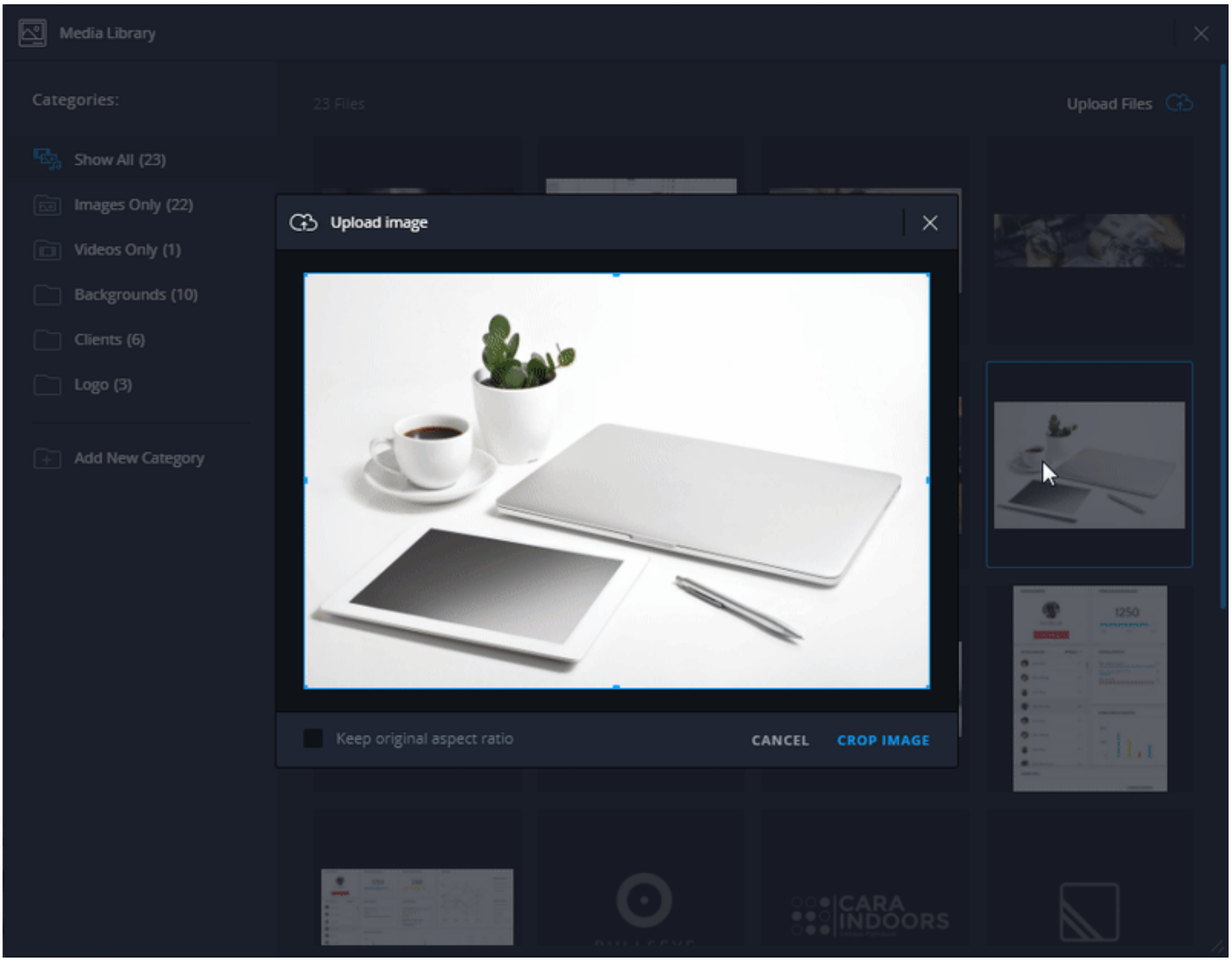
**When you're uploading a new file to the media gallery, it's added to the currently active category, except for the case of uploading a video file when the active category is "Images Only" and vice versa.*



How to crop an image in media library

To crop an image in the media library, you should:

open the media library;
call the context menu of the target file by right-clicking the file;
choose the "Crop Image" item;
select the needed area;
press the "Crop image" button.

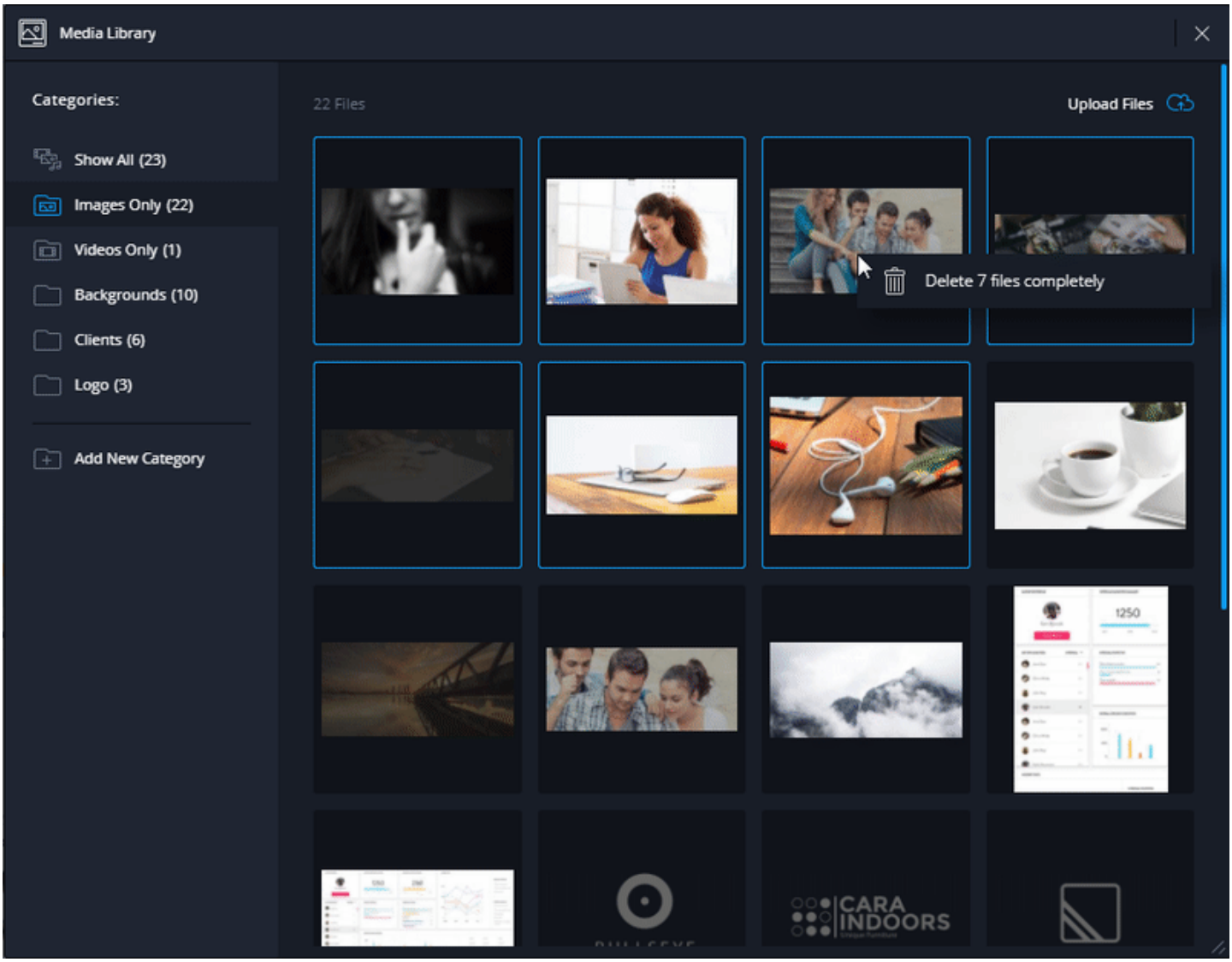


How to delete files from the media library

To delete files from the media library you should:

open the media library;
choose the needed file(s) (the library supports multiple selection with Ctrl or Shift);
call the context menu of the selected file(s) by pressing right mouse button;
choose the "Delete file completely" item.

To save changes, use the "Save" button that resides in the right corner of the top bar.

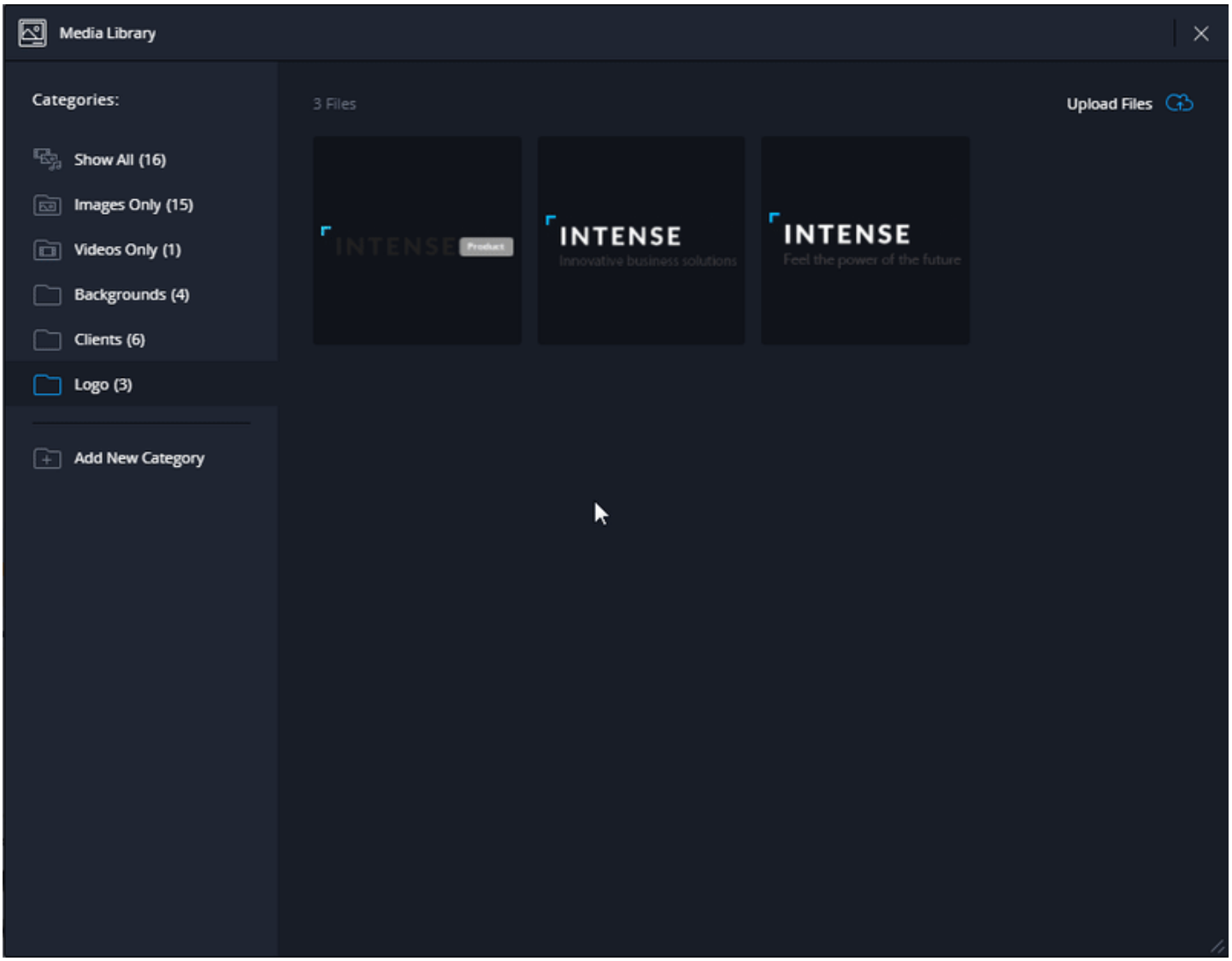


How to add a media file to the category

To add a media file to the category, you should:

open the media library;
choose the necessary file(s) (media library supports multiple selection with Ctrl or Shift);
press the right mouse button, drag file(s) to the target category name area.

To save changes, use the "Save" button that resides in the right corner of the top bar.

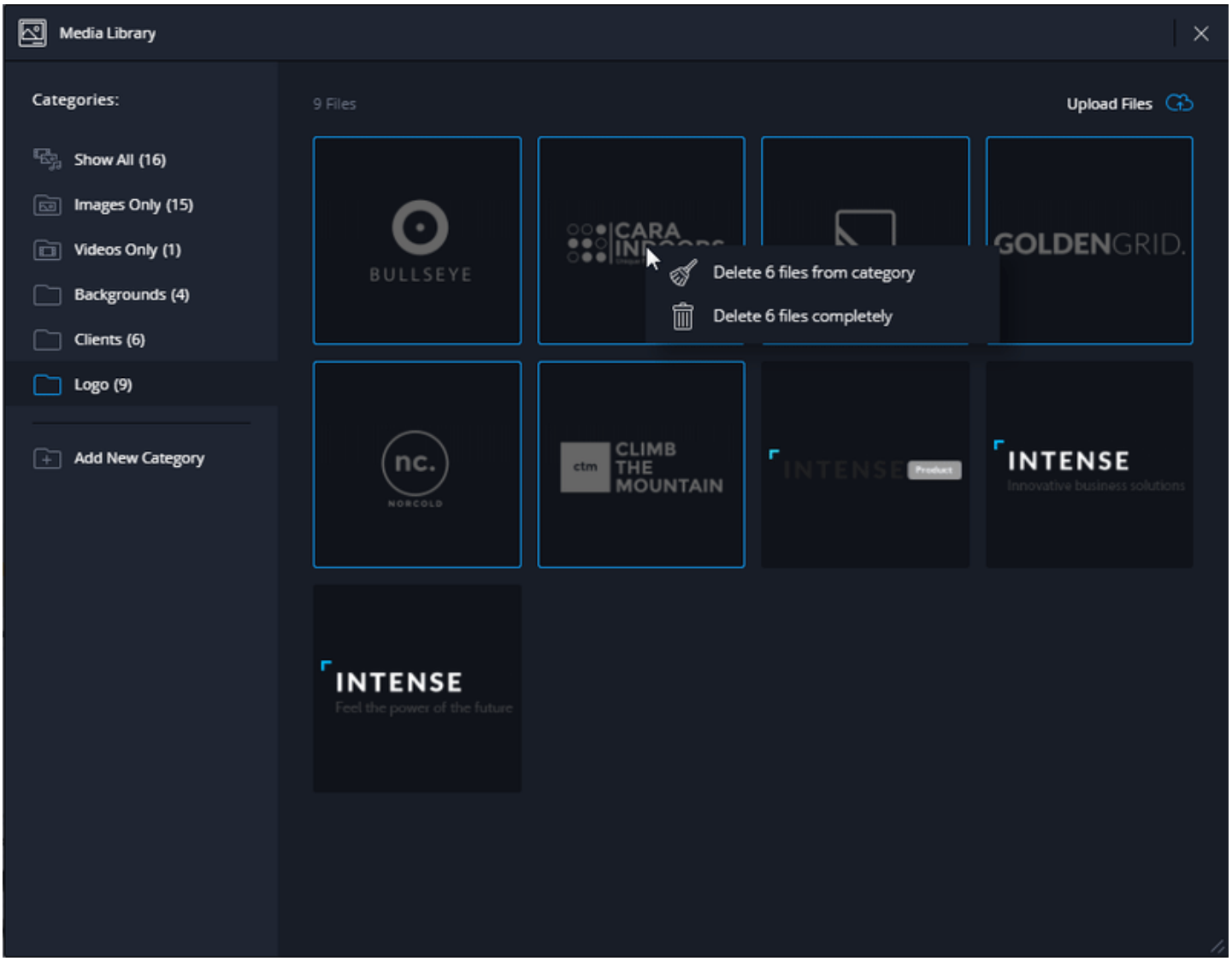


How to remove a media file from a category

To remove a media file from a category, you should:

open the media library;
select a custom category;
select the needed file(s) (media library supports multiple selection with Ctrl or Shift);
call the context menu of the selected file(s) by pressing right mouse button;
choose the "Delete file from category" item for the target file(s).

To save changes, use the "Save" button that resides in the right corner of the top bar.

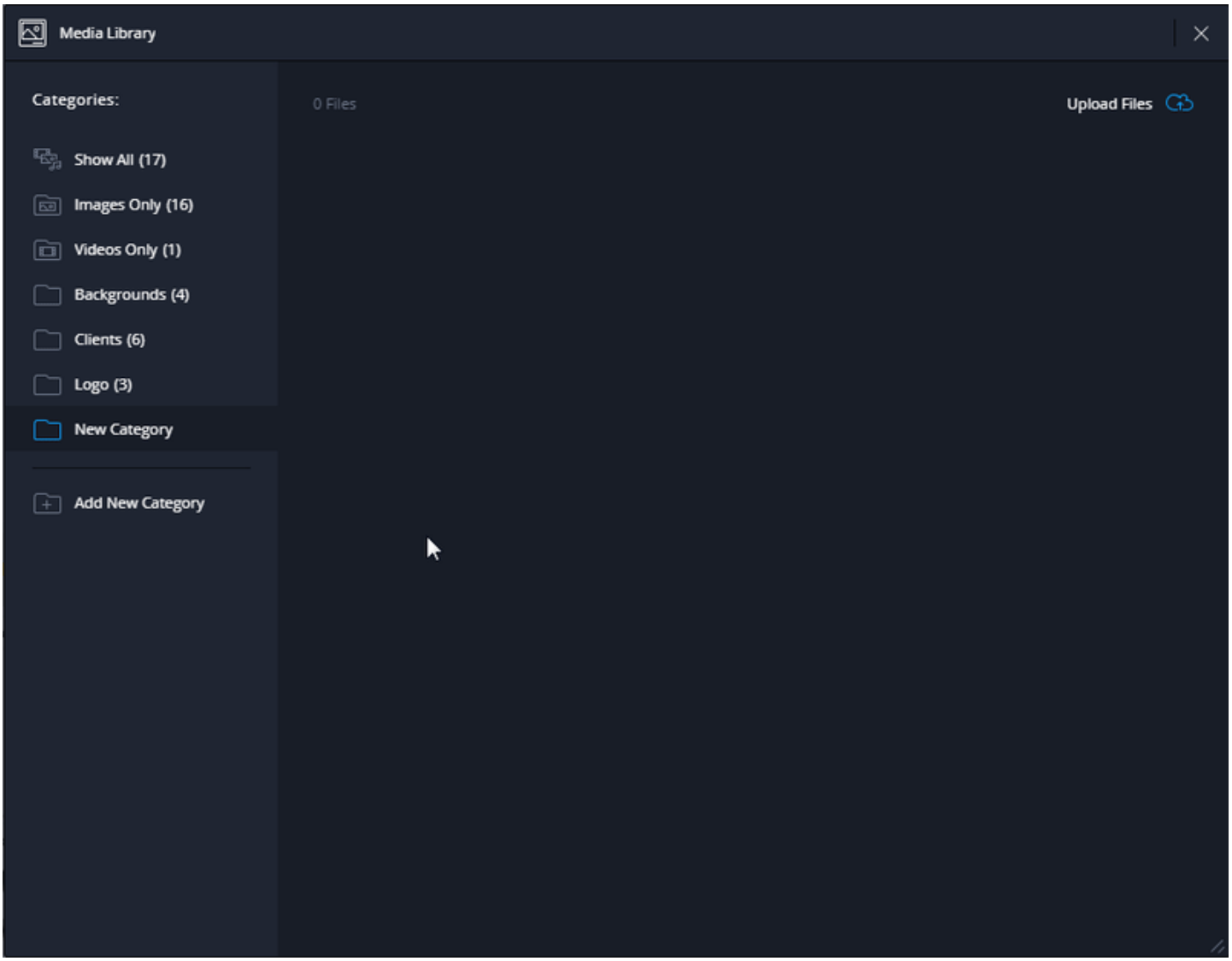


How to add a category to media library

To add a category to the library, you should:

Open the media library;
Press the "Add new Category" button.

To save changes, use the "Save" button that resides in the right corner of the top bar.

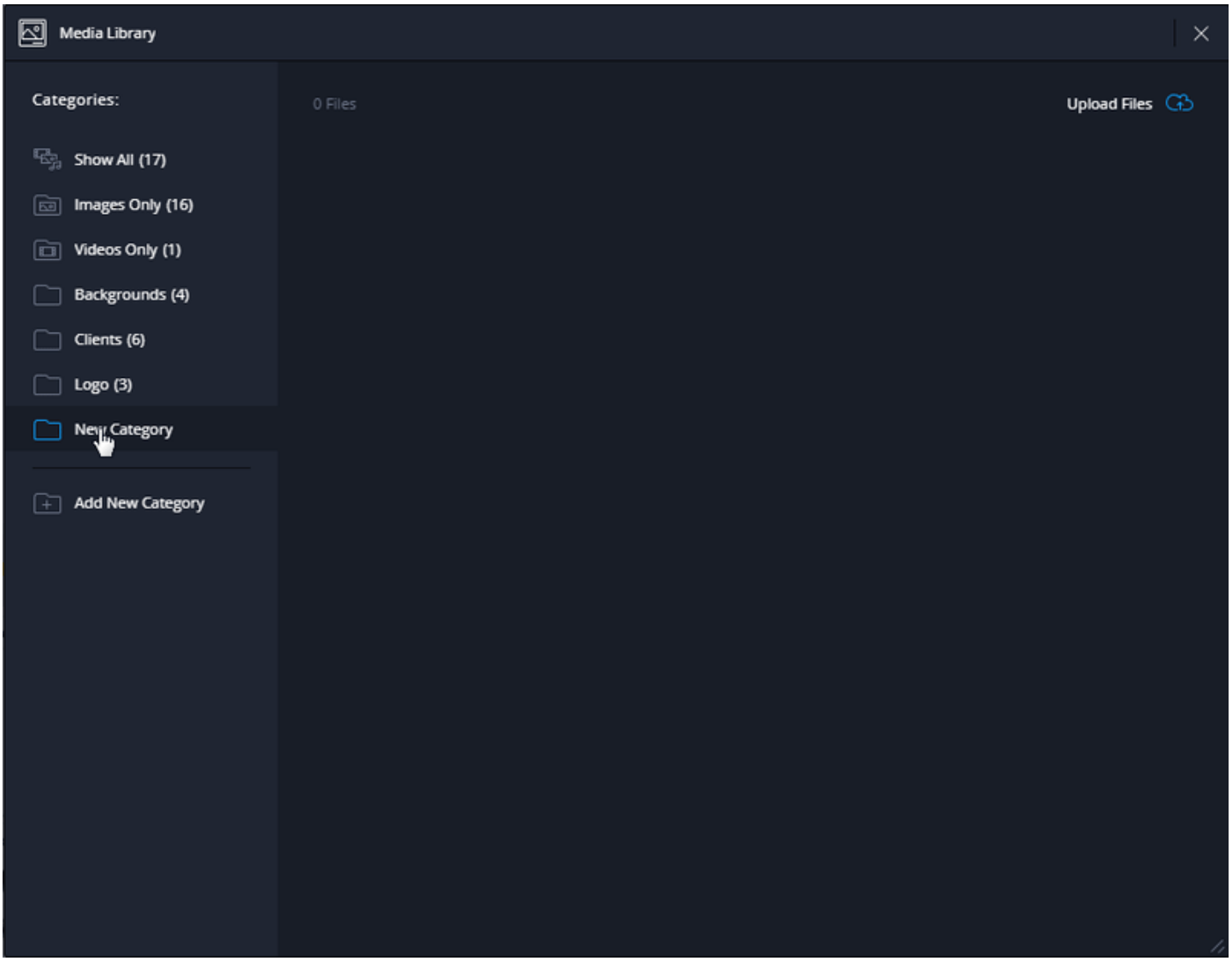


How to remove a category from media library

To remove a category from media library, you should:

open the media library;
select the needed custom category;
call the context menu of the selected category by right-clicking it;
select the "Delete category" item.

To save changes, use the "Save" button that resides in the right corner of the top bar.

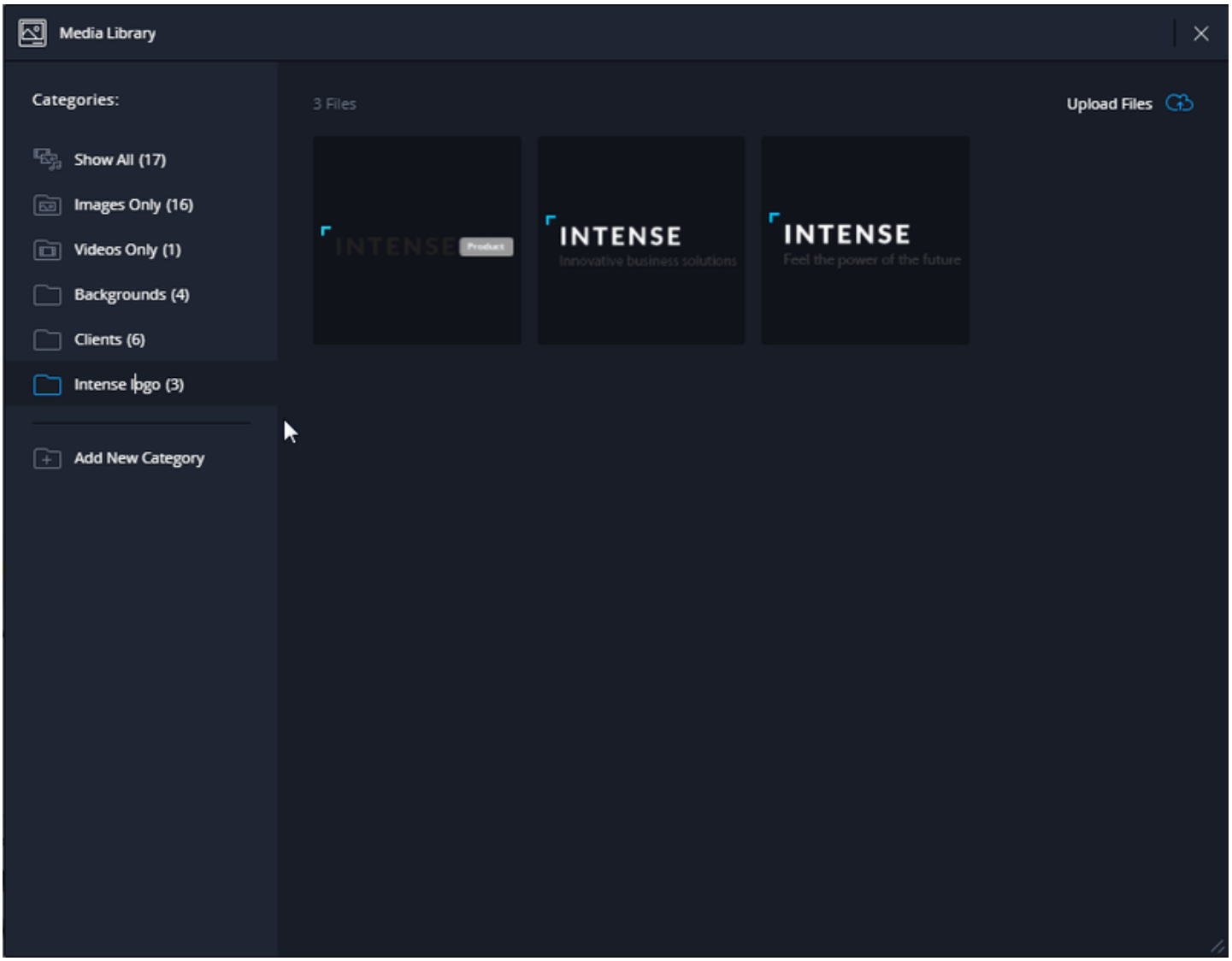


How to rename a category

To rename a category, you should:

- open the media library;
- select the needed custom category;
- call the context menu of the selected category by right-clicking or double-clicking the left mouse button on the category name;
- choose the "Rename category" item, enter the needed name, press Enter.

To save changes, use the "Save" button that resides in the right corner of the top bar.

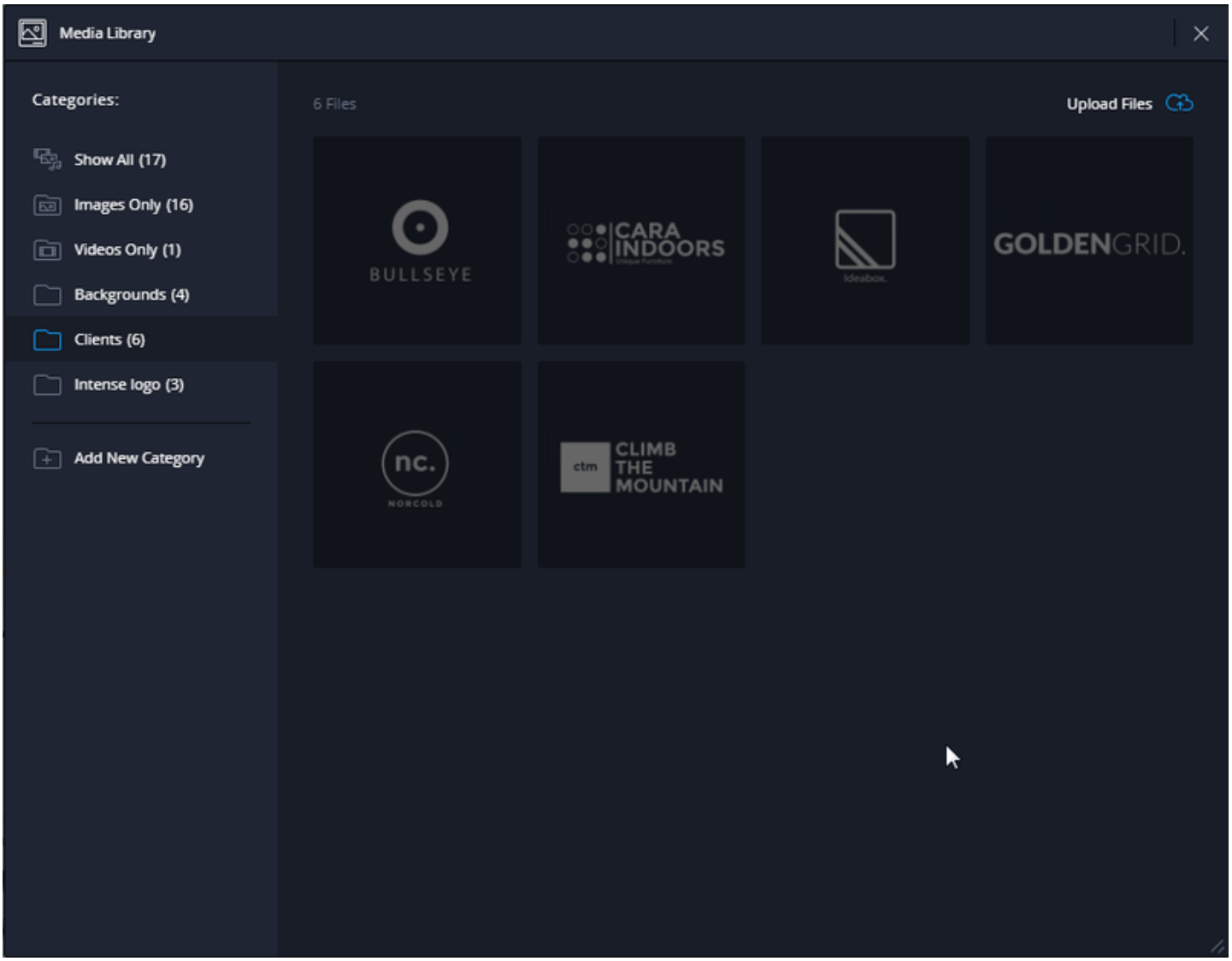


How to delete media files from the category

To delete all media files from the category, you should:

open the media library;
select the needed custom category;
call the context menu of the selected category by right-clicking it;
choose the "Clear category" item.

To save changes, use the "Save" button that resides in the right corner of the top bar.



Icons manager

This tool provides you with an option to manage the project's iconic fonts. By default, the builder has 7 popular iconic fonts available for use:

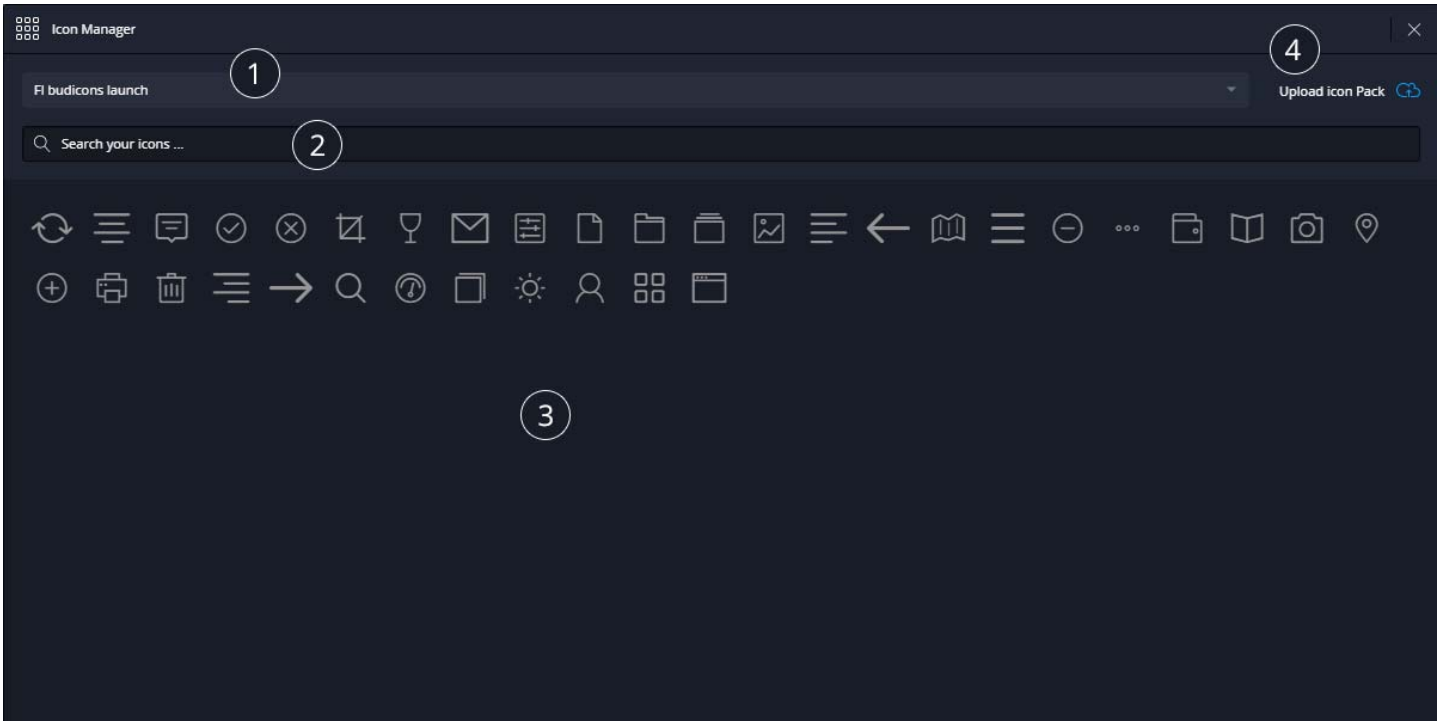
- FontAwesome
- Material Design Icons
- Material Icons by Google
- Flat Icons Bigmug Line
- Flat Icons Budicons Launch
- Flat Icons Budicons Free
- GlyphIcons Halflings

Use the main Novi Builder menu to reach the icons manager.

Icons manager looks the following way:

1. Select option to choose the font family (among the 7 basic font families, included by default).
2. Search field to search within the selected font family.
3. Output area of the selected iconic fonts.

4. Button to upload the archive of a custom iconic font.



How to add a custom font to the builder project

In order to add a custom iconic font to the project, perform the following steps:

Open the icons manager, it can be reached from the main builder menu.

Click the "Upload icon Pack" button

Browse your font archive. It should meet the requirements, listed in the "Creating an archive for an iconic font import" section.

Import the font and save the changes by pressing the Save Project button in the top right corner of the main builder panel.

Creating an archive for an iconic font import

You should prepare the font archive to successfully import the font into the builder. It has to comply with the following rules:

1. The archive should not include parent directories for the files.
   - Incorrect:
     font-awesome.zip
     font-awesome
     font-awesome.css
     fonts
     FontAwesome.otf
     fontawesome-webfont.eot
     fontawesome-webfont.ttf
   - Correct:
     font-awesome.zip
     font-awesome.css
     FontAwesome.otf
     fontawesome-webfont.eot

fontawesome-webfont.ttf

2. The archive must have a CSS font file.
3. The CSS font file has to contain a @font-face rule with all the references to font files and font-family property specified.
4. The CSS font file has to contain a set of classes to define the icons.

Replacing the icon in a project

You should use the icons replacement plugin to change the icons within the project. The detailed description of the plugin is available at the Icons plugin section, under the Plugins.

Resolution Preview Tool

You can find the resizer tool at the main Novi interface. It can be used for previewing the visual editor on the different screen sizes.

Resizer has 7 different dimensions, by default the “Desktop Full-HD” is active. There are also the following dimensions available:

Mobile Portrait 320x480 px
Mobile Landscape 480x320 px
Tablet Portrait 768x1024 px
Tablet Landscape 1024x768 px
Notebook 1280x720 px
Desktop PC 1600x900 px

Keep in mind, the resizer shows your project on the different screen sizes, but it does not emulate

---

Export & Publish Project

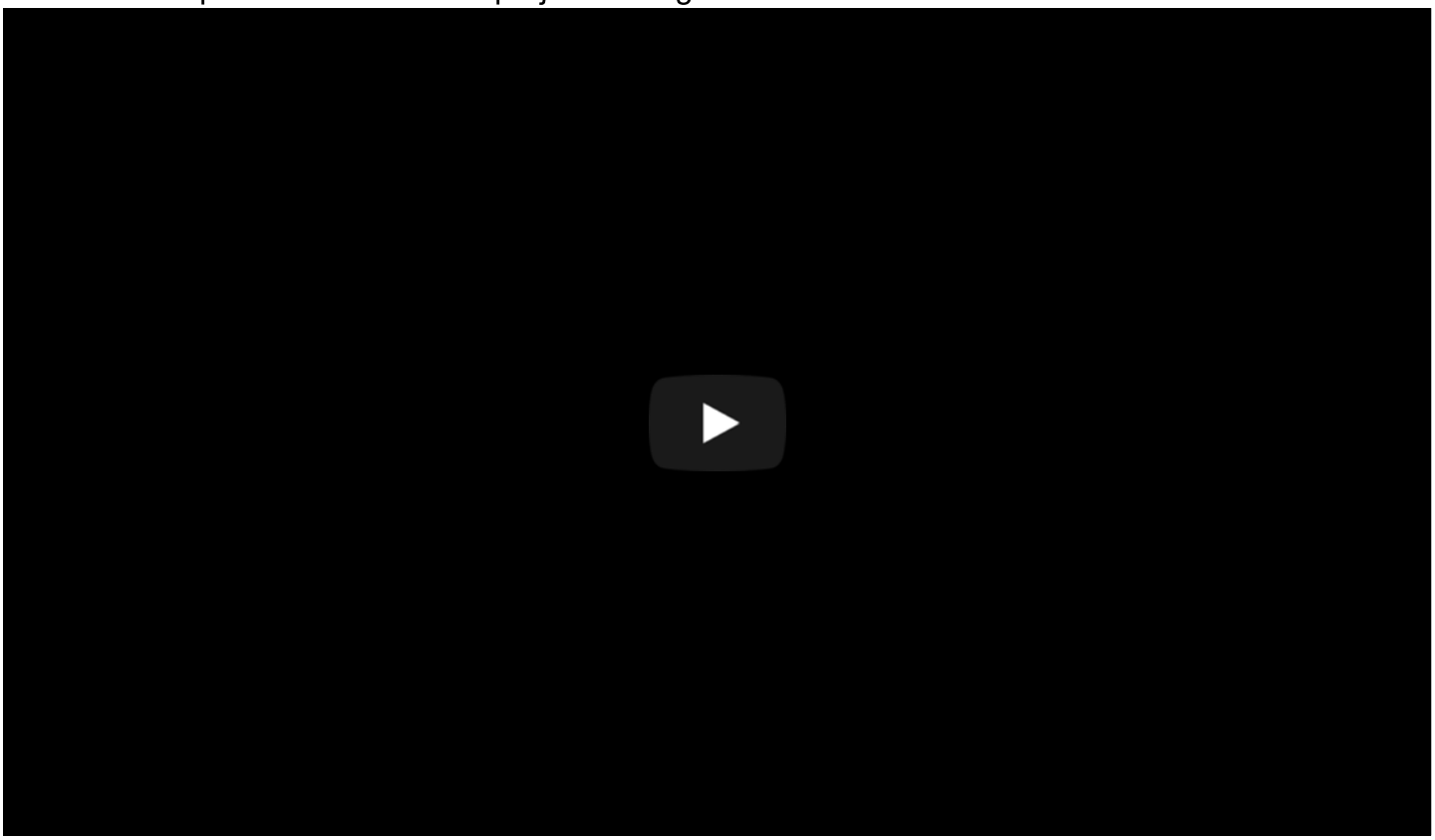
You can export the Novi Builder project from the main builder interface, using the "Export Code" option in the main menu.

Export allows you to save the .zip archive of your project to the specified destination. The template structure will be completely saved.

Same, you can publish the Novi Builder project from the main builder interface. The project will be published at the specified path that you set at the Project Settings under the "Publishment path".

At this moment the project can be published exclusively on the server where the Novi Builder is installed.

Be careful! Re-publishing will update the content of your website if you don’t change the "Publishment path" attribute at the project settings section.



Hotkeys

There is a range of useful hotkeys in various Novi Builder modules.

Application

Ctrl + S - Save the project

Visual Editor (Edit Mode)

Ctrl + Enter - deselect the current element
Esc - deselect the current element

Visual Editor (Design Mode)

Alt + ~ - navigate to the parent container of the selected item
Ctrl + X - cut the selected item
Ctrl + C - copy the selected item
Ctrl + Alt + V - paste before the selected item
Ctrl + V - paste into the selected item
Ctrl + Shift + V - paste after the selected item
Delete - remove the item
Ctrl + D - duplicate the selected item
Ctrl + Q - save the selected item as a preset

Code Editor

Ctrl + S - Save in the editor
Ctrl + L - file formatting
Ctrl + D - remove the entire line
Ctrl + Shift + D - duplicate the entire line
Ctrl + Z - undo the last change
Ctrl + C - copy into the clipboard
Ctrl + V - paste from the clipboard
Alt + Down - shift the line down
Alt + Up - shift the line up
Alt + Delete - remove to the end of the line
Alt + Backspace - remove to the start of the line
Ctrl + A - select all
Shift + Left - select one symbol to the left
Shift + Right - select one symbol to the right
Shift + Home - select to the start of the line
Shift + End - select to the end of the line
Ctrl + Shift + Home - select to the start of the file
Ctrl + Shift + End - select to the end of the file
Ctrl + LMB (left mouse button) - set multicursor
Ctrl + Alt + Right - select the next reiteration setting of multicursor
Ctrl + Alt + Left - select the previous reiteration setting of multicursor
Ctrl + F - find
Ctrl + H - replace
Ctrl + K - find the next item
Ctrl + Shift + K - find the previous item

Updater

Builder updates are available starting from the v. 0.8.3.

By default Novi updates are disabled, you should tick the "Check for updates" item under the system settings in order to enable them.

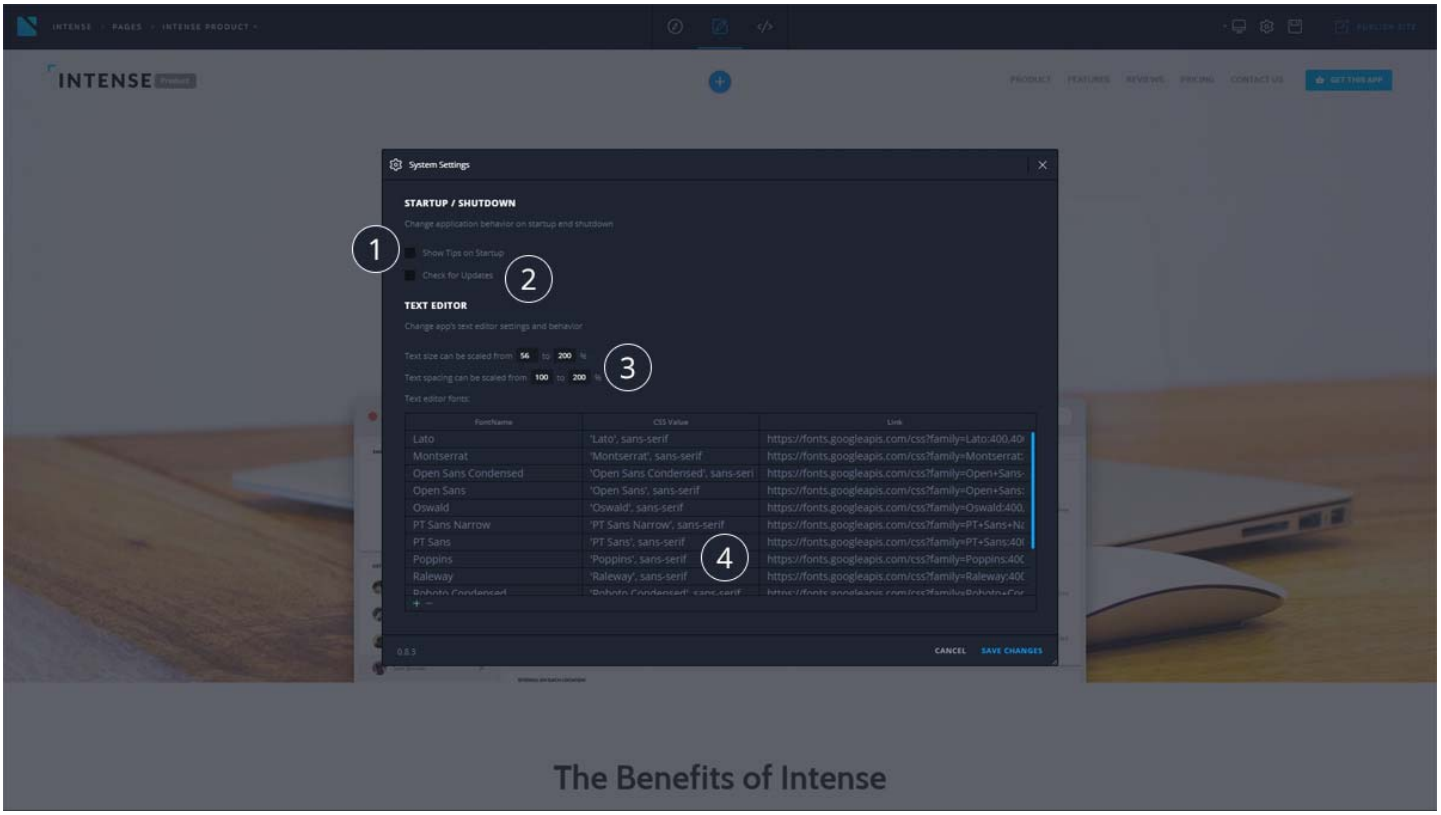
If there is a newer version of the Novi Builder, you will see the dialog window after the loading of the builder. You'll be able to update the builder by following its instructions.

The current project will remain unchanged after the update applied.

System Settings

Novi system settings are accessed via the main builder menu.

1. 'Show tips on startup' defines whether to load the start presentation while the builder loads.
2. 'Check for Updates' indicates whether to check for the builder updates (starting from v.0.8.3).
3. Setting of the project font size limit (is applied in the text editor)
4. Table of the fonts used in the template, by default includes the 13 popular Google Fonts service fonts.



Plugins

Overview

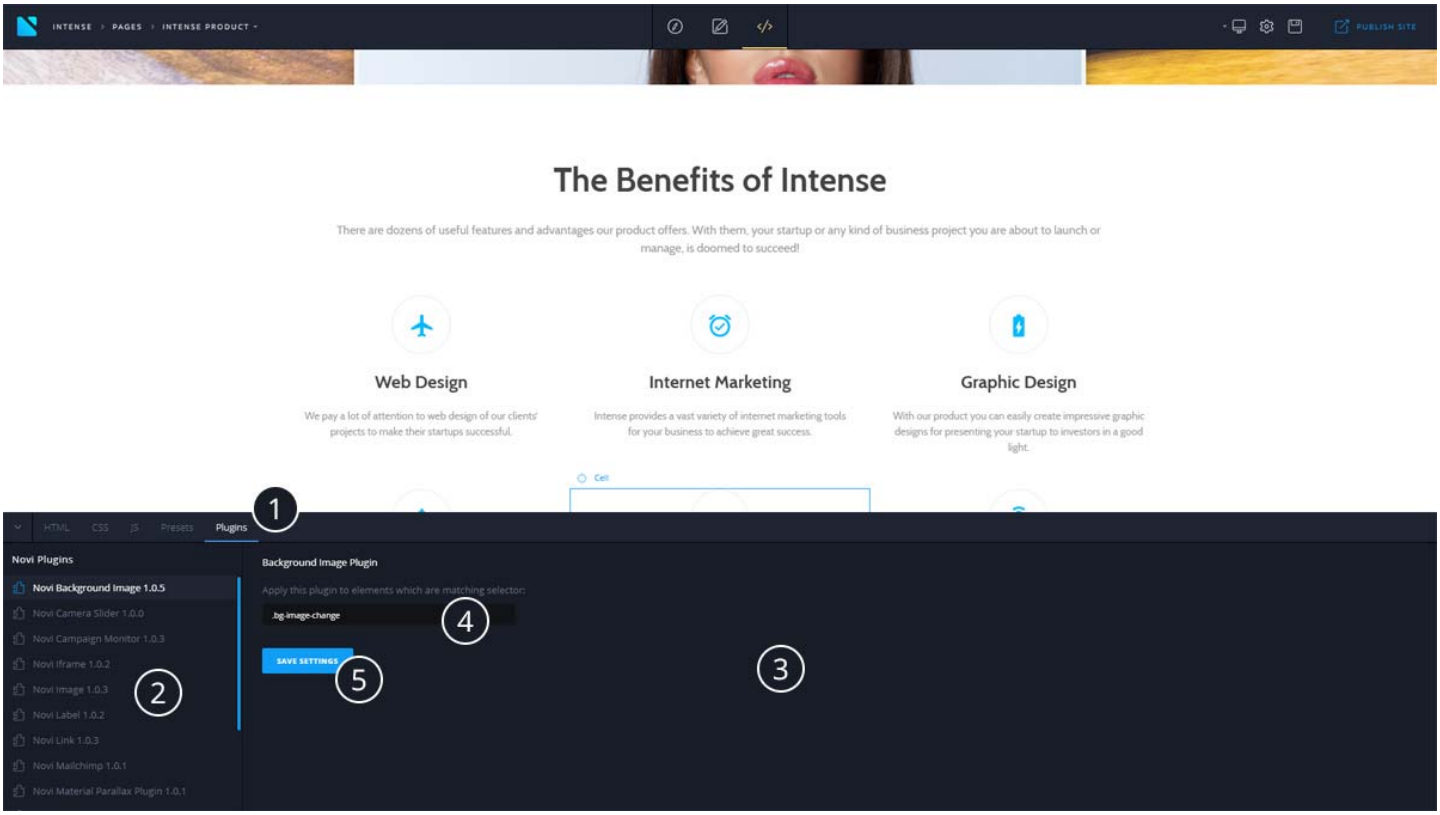
Novi plugins are the separately compiled modules.

The work of plugins can be traced exclusively in the Edit mode, while the configuration of the plugins takes place in the Design mode.

Plugins management panel is located on the developer's toolbar.

Plugins configuration area includes the following:

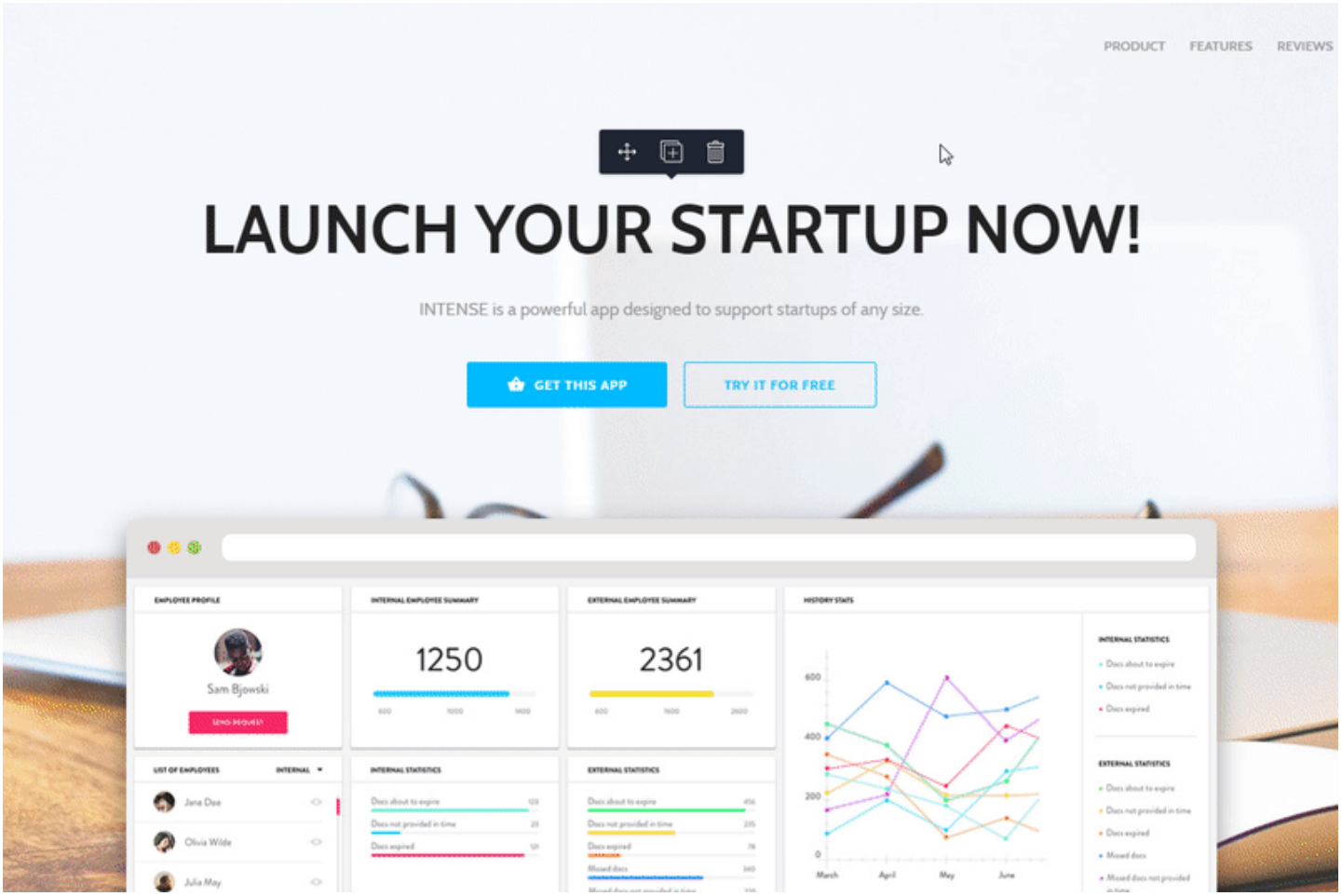
1. Trigger for the panel opening
2. The list of the installed plugins
3. Configuration area of the selected plugin
4. querySelector field (required) - is indicated as a CSS rule (class, id, etc.). The item defined by the selector will be processed by the plugin.
5. ‘Save’ option.



The plugin settings may differ, the only required field is the querySelector.

Background Image Plugin

Allows you to change the items background image.



Plugin setup:

querySelector (by default is set to ".novi-background") - allows you to define the item processed by the plugin.

Editor panel:

- Trigger for the plugin display

Image select area (Media Gallery)

The image to be added is selected by double-clicking the left mouse button on the target image.

You can learn more about media gallery capabilities under "Media gallery" item (link) in the "Features" section.

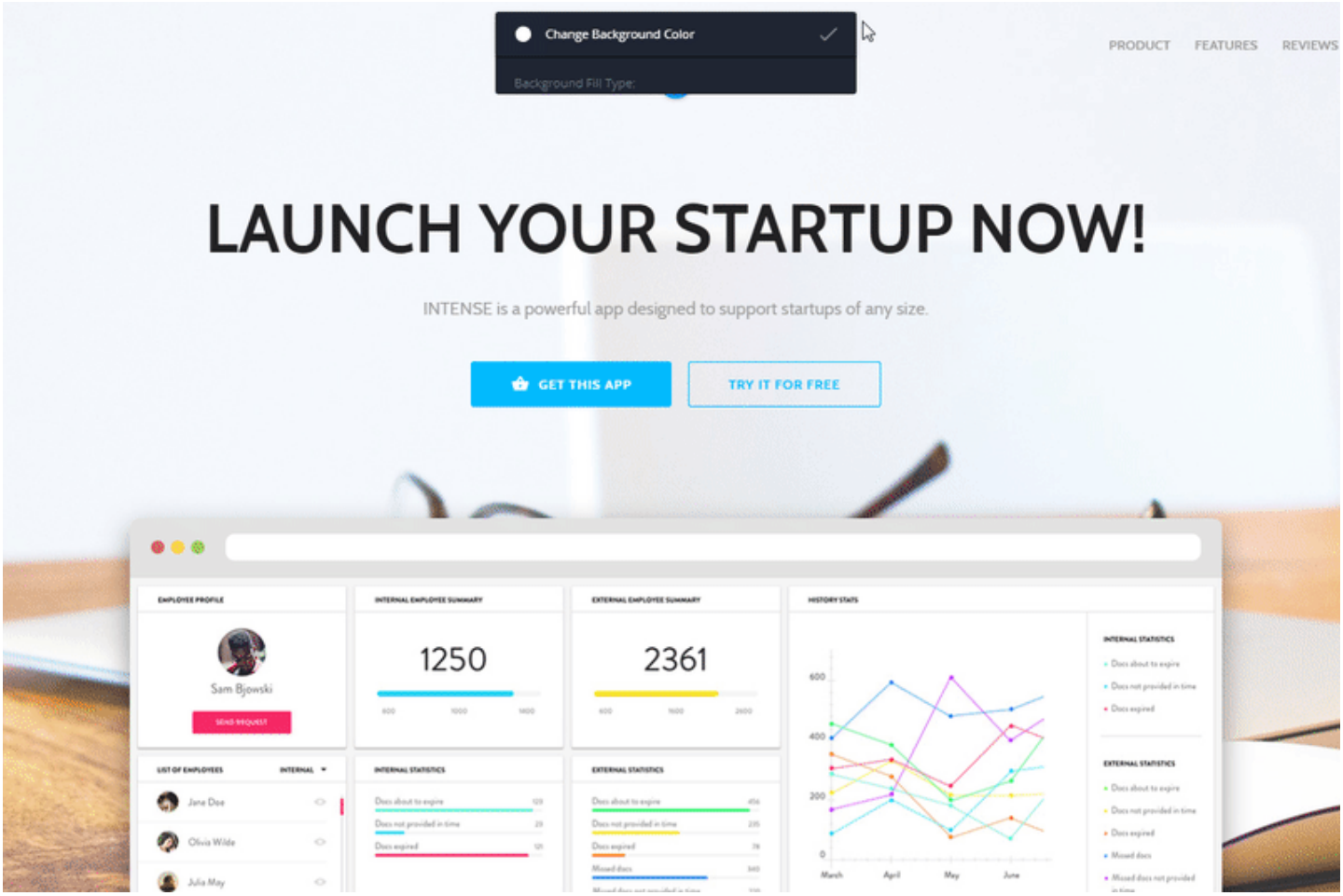
Image upload area

"Keep original aspect ratio" option that allows to maintain the proportions of the replaced image.

The detailed plugin documentation is available at the following link (https://github.com/NoviBuilder/novi-plugin-background-image)

Background Plugin

Lets you change the background color of the element.



Plugin setup:

querySelector (".novi-background" by default) lets you define the element, to which the plugin functionality will be applied.

Editor panel:

- trigger for plugin display (the color depends on the element background).

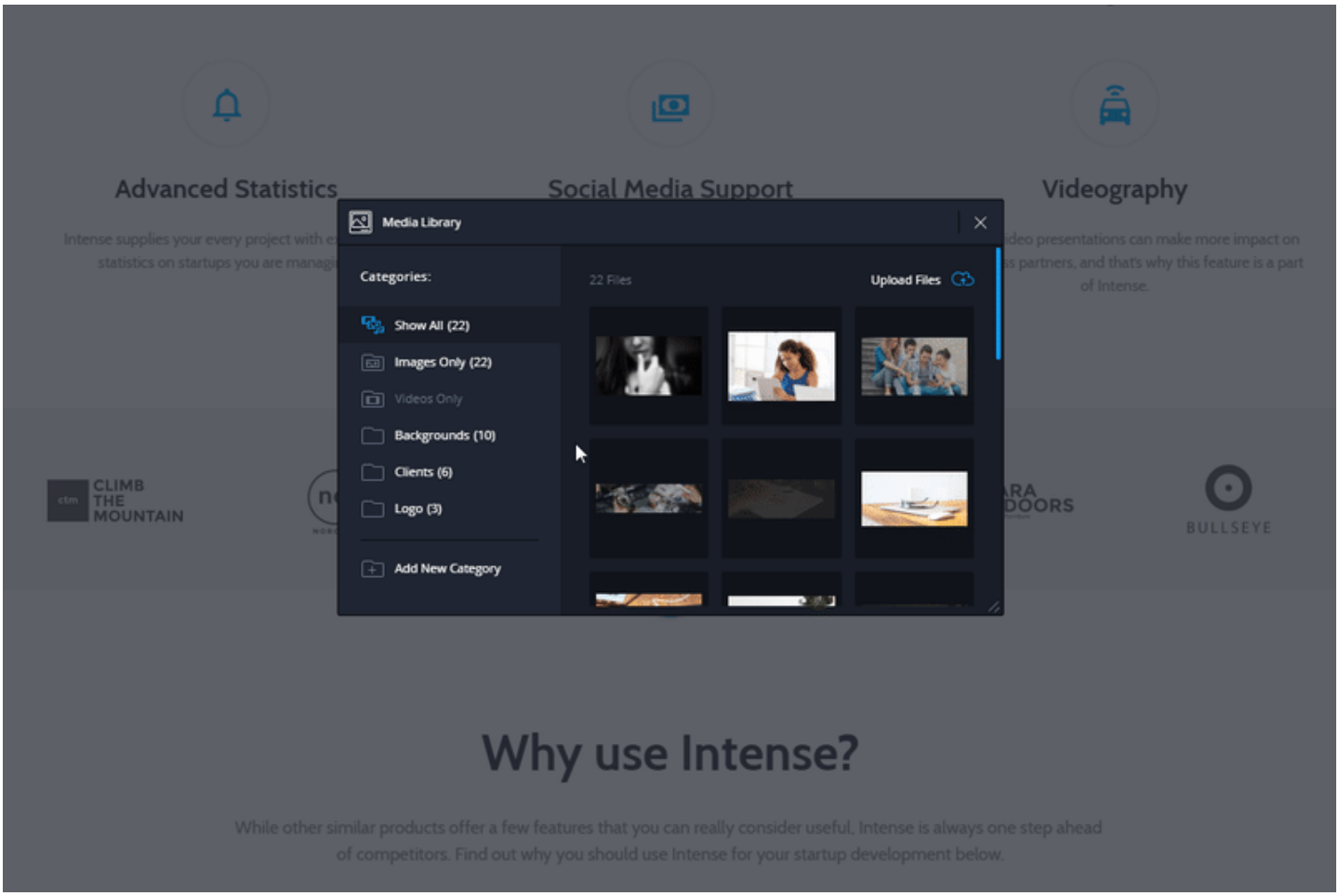
Color selection area

Background Fill Type" switcher displays a panel for color or element gradient change.

More detailed plugin documentation is available under the followinglink (https://github.com/NoviBuilder/novi-plugin-background)

Image Plugin

Allows you to replace an image.



Plugin setup:

querySelector (by default is set to "img[src]") - allows you to define the item processed by the plugin.

Editor panel:

- Trigger for the plugin display

Image selection area (Media Gallery)

The image to be added is selected by double-clicking the left mouse button on the target image.

You can learn more about media gallery capabilities under "Media gallery" item (link) in the "Features" section.

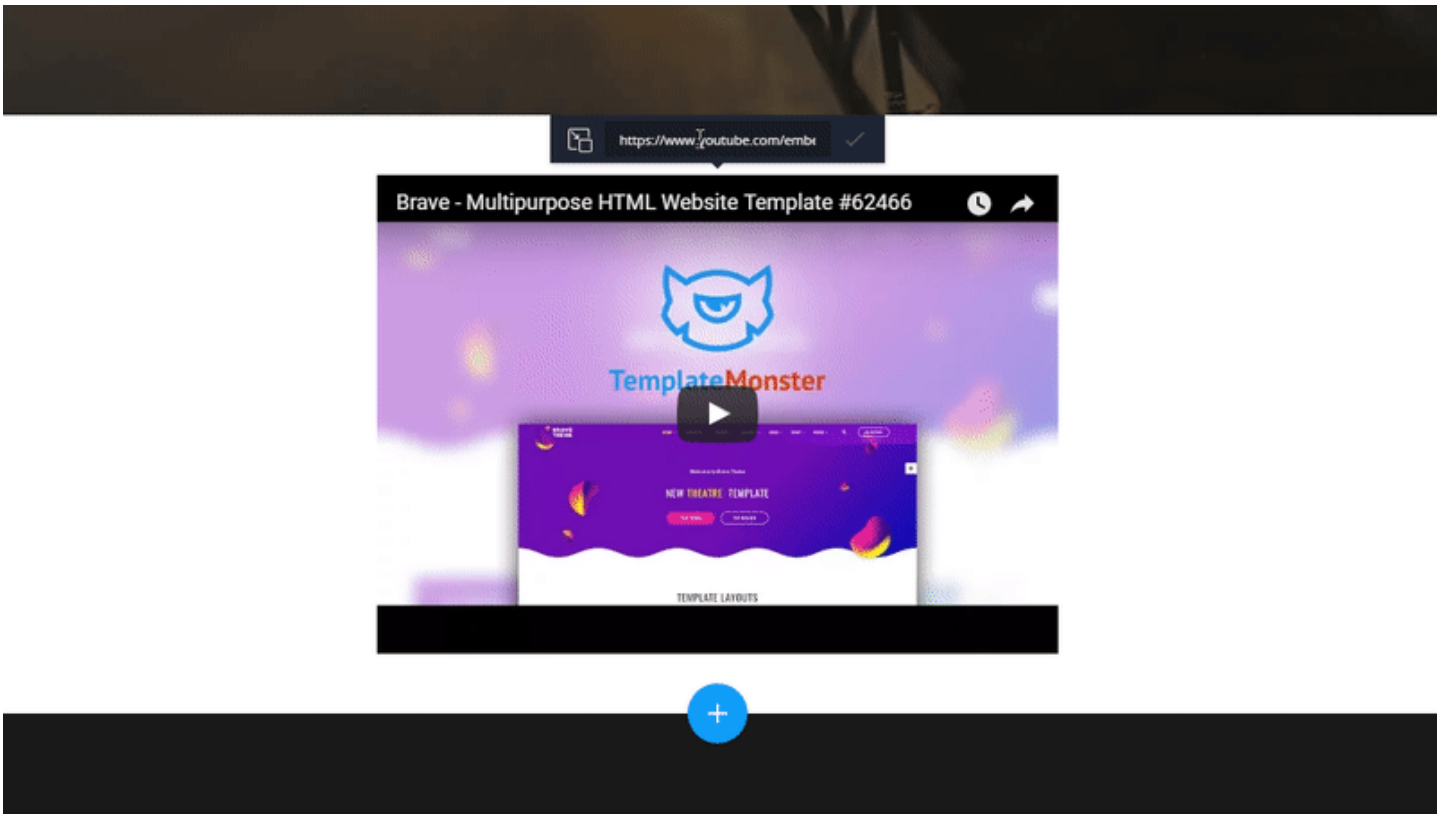
Image crop area:

"Keep original aspect ratio" setting is used for preserving the aspect ratio of the original image.

The detailed plugin documentation is available at the following link. (https://github.com/NoviBuilder/novi-plugin-image)

Iframe Plugin

Allows you to specify a resource for displaying the content within a frame.



Plugin setup:

querySelector (by default is set to "iframe[src]") - allows you to define the item processed by the plugin.

Editor panel:

- Trigger for the plugin display

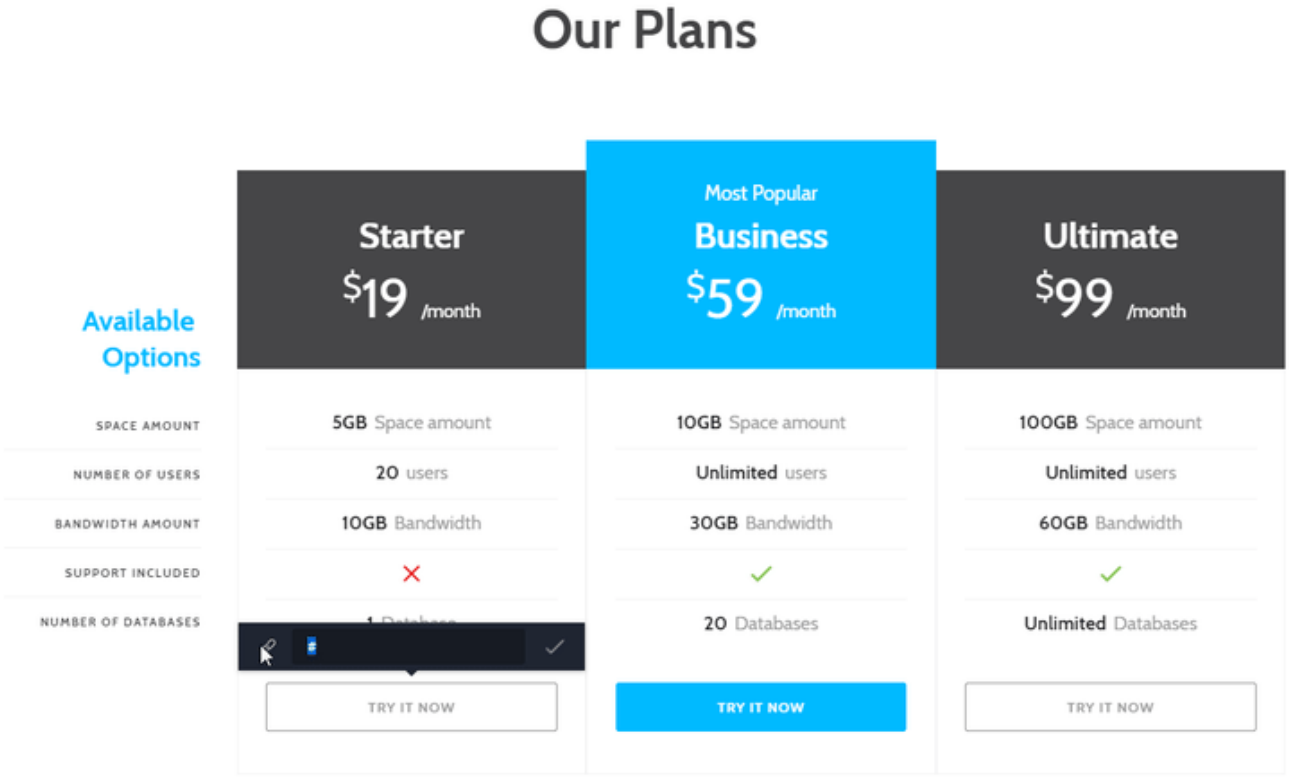
Frame configuration area:

Field for specifying the resource for the frame

The detailed plugin documentation is available at the following link. (https://github.com/NoviBuilder/novi-plugin-iframe)

Link Plugin

Allows you to change the link.



Plugin setup:

querySelector (by default is set to "a[href]") - allows you to define the item processed by the plugin.

Editor panel:

- Trigger for the plugin display

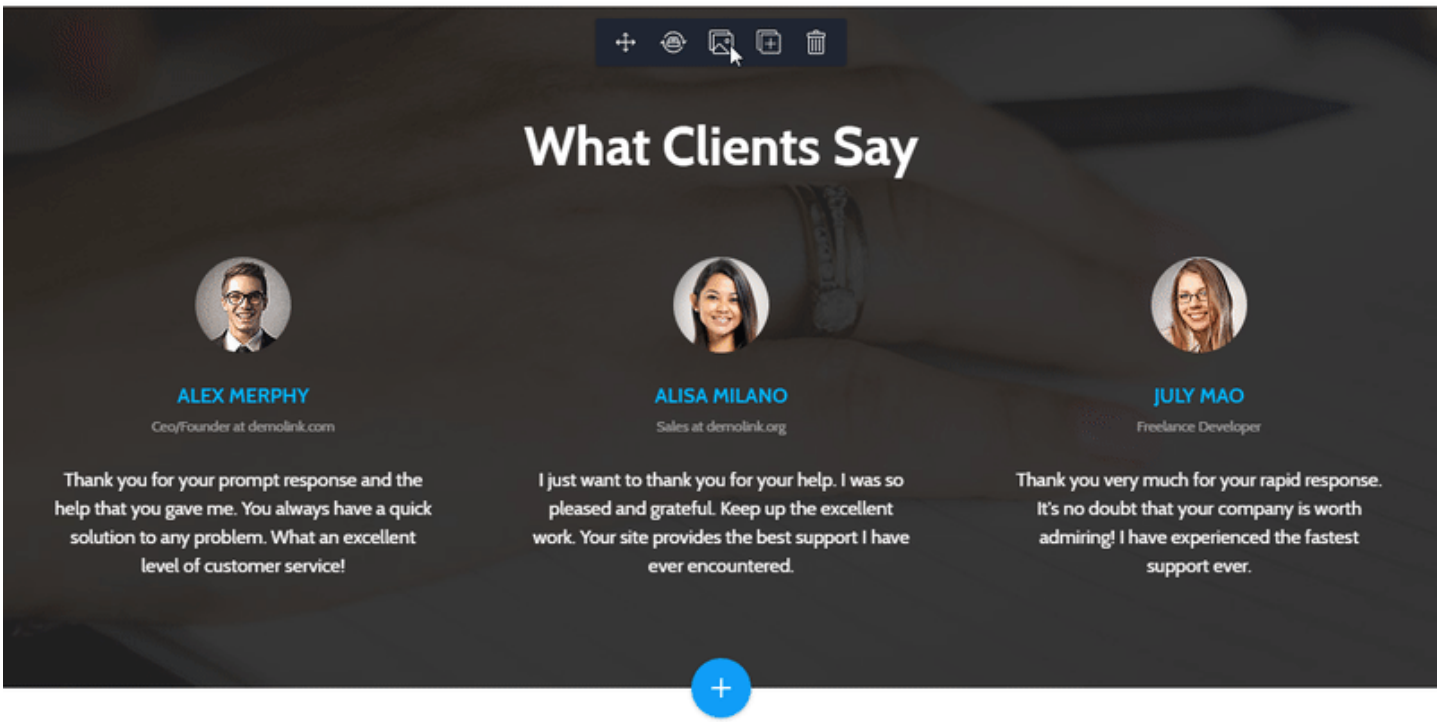
Plugin configuration area

Field for specifying the src attribute of the link.

The detailed plugin documentation is available at the following link.
(https://github.com/NoviBuilder/novi-plugin-link)

Material Parallax Plugin

Works with the Materianize Parallax script, it allows you to replace the source images for parallax using the visual editing tools.



Plugin setup:

querySelector (by default is set to ".parallax-container") - allows you to define the item processed by the plugin.

Editor panel:

- Trigger for the plugin display

Image upload area:

"Choose file" button that allows to select an image

"Upload background image" button that allows to upload the selected image

Image selection area (Media Gallery)

The image to be added is selected by double-clicking the left mouse button on the target image.

You can learn more about media gallery capabilities under "Media gallery" item (link) in the "Features" section.

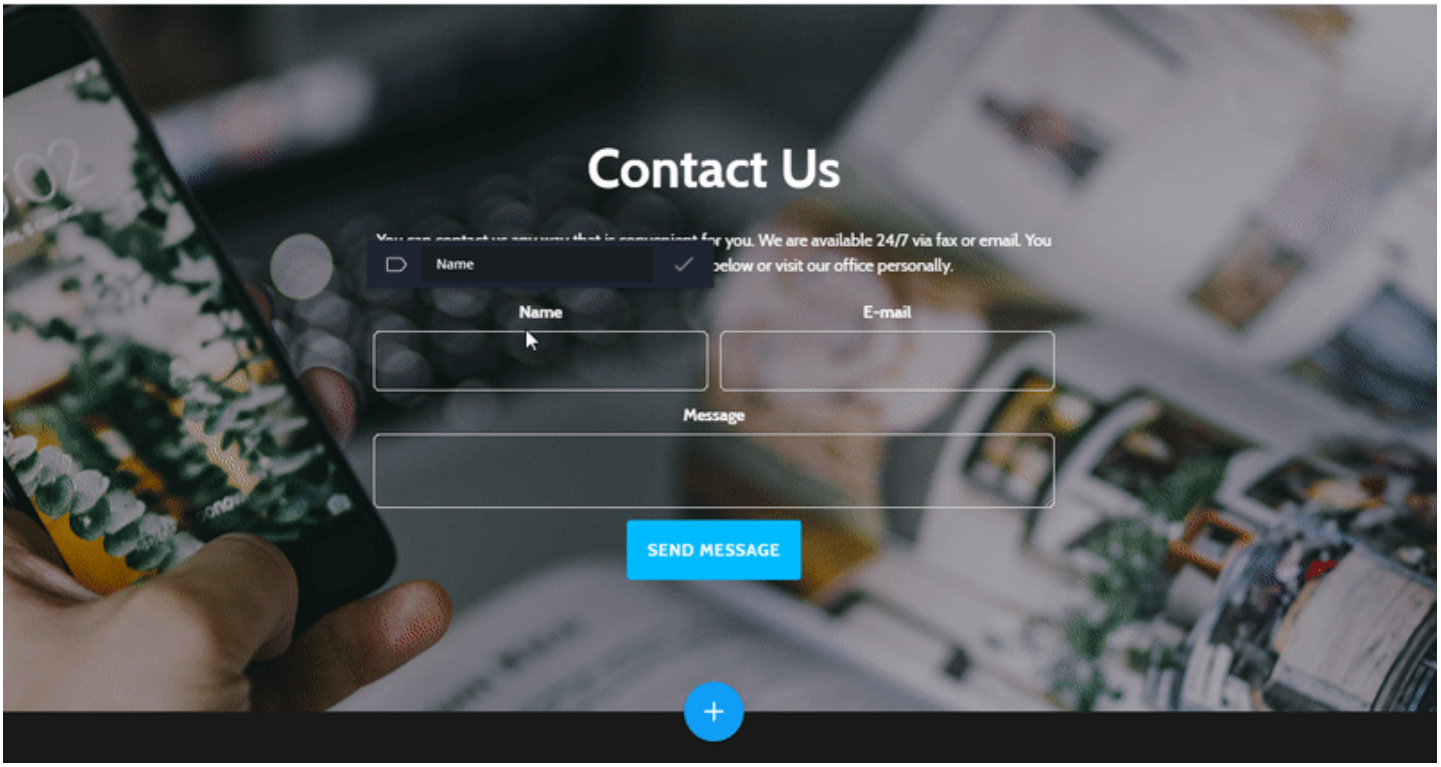
Image crop area:

"Keep original aspect ratio" setting is used for preserving the aspect ratio of the original image.

The detailed plugin documentation is available at the following link. (https://github.com/NoviBuilder/novi-plugin-material-parallax)

Label Plugin

Allows you to change the label of the form field.



Plugin setup:

querySelector (by default is set to "label") - allows you to define the item processed by the plugin.

Editor panel:

- Trigger for the plugin display

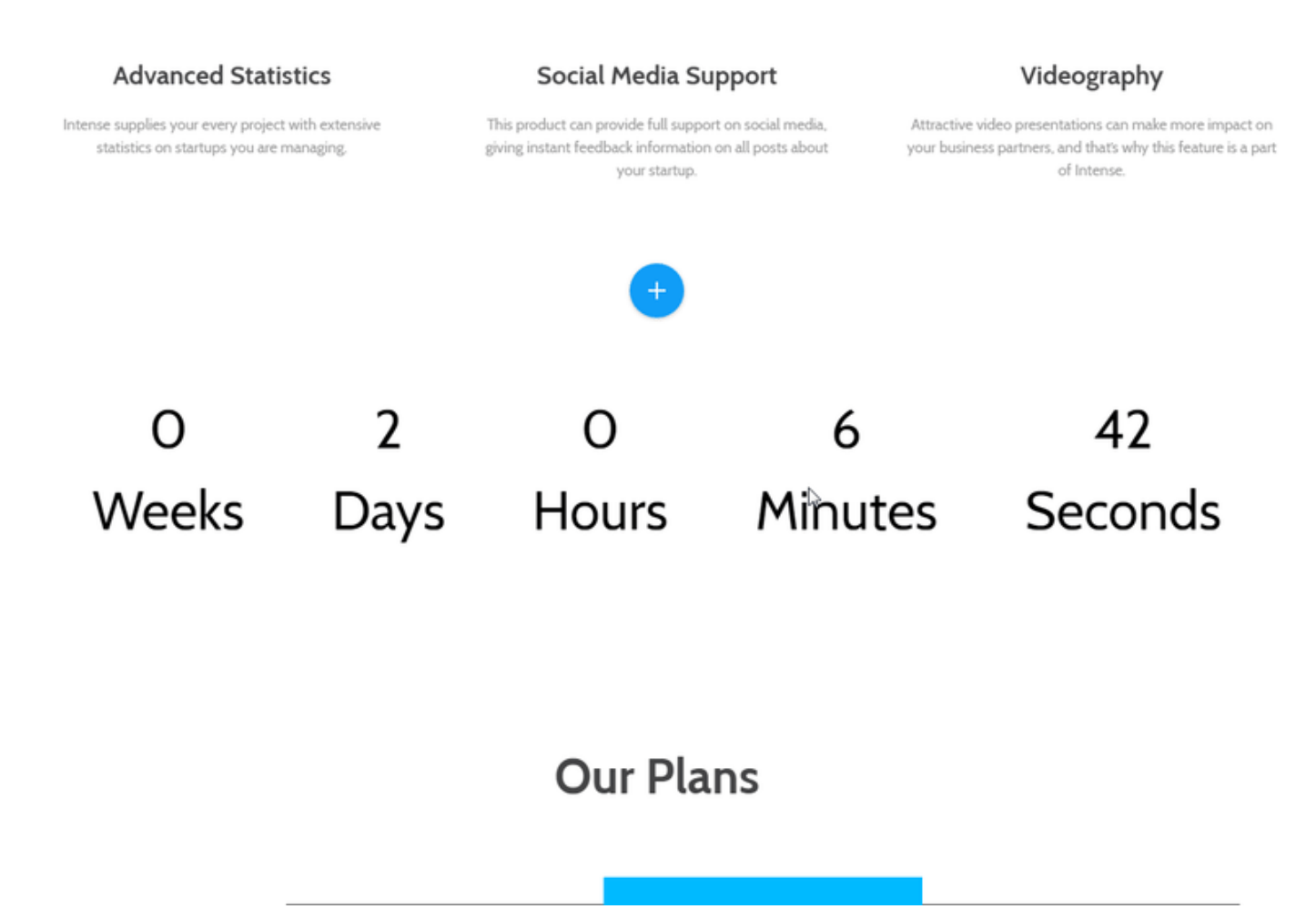
Plugin configuration area

Field for specifying the label value

The detailed plugin documentation is available at the following link.
(https://github.com/NoviBuilder/novi-label)

Countdown Plugin

Lets you set up jQuery using visual tools.



Plugin setup:

querySelector (".novi-countdown" by default) - lets you define the element, to which the plugin functionality will be applied.

Editor panel:

- trigger for plugin settings display.

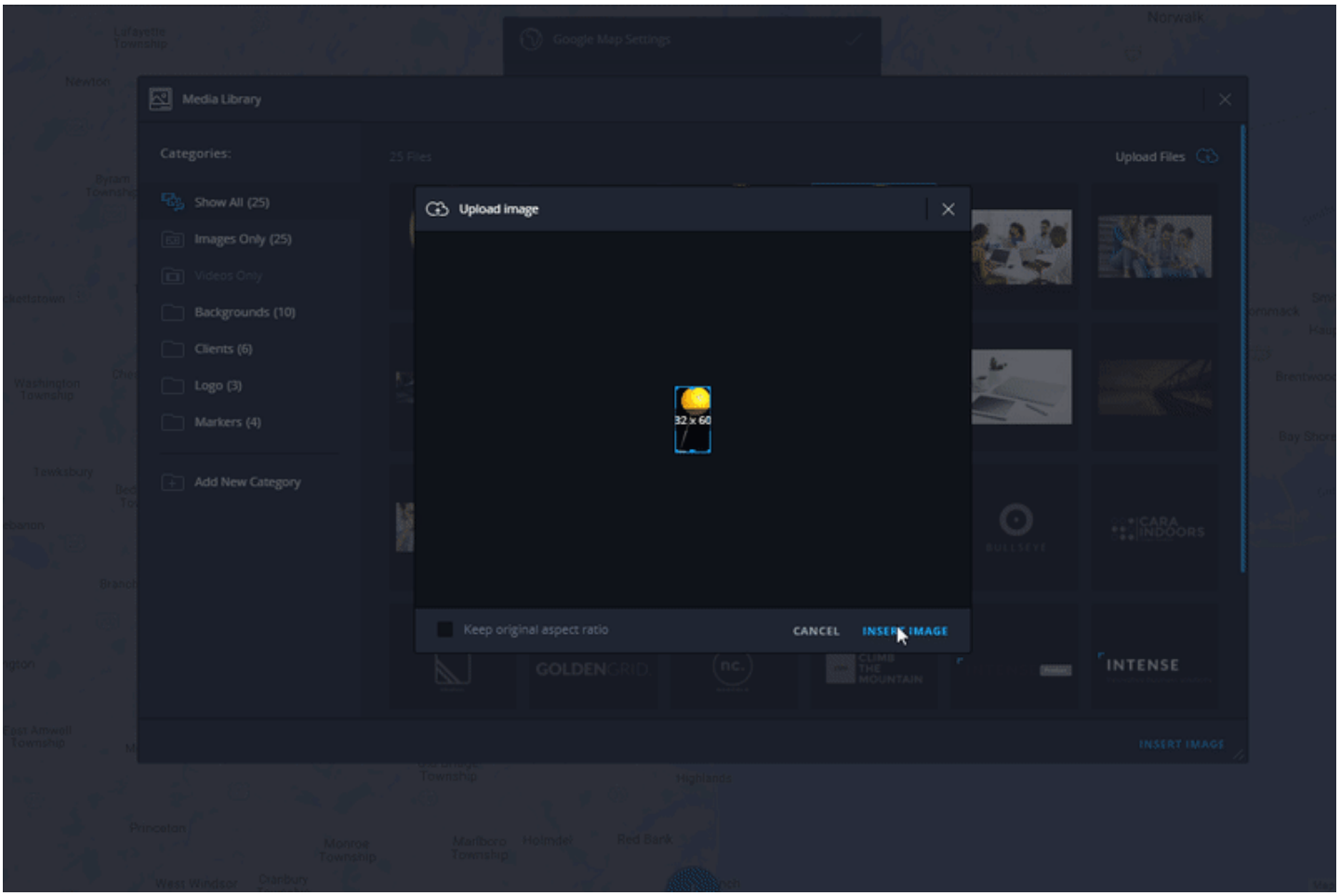
Plugin settings area:

"Type" switcher lets you choose the countdown type.

More detailed plugin documentation is available under the followinglink. (https://github.com/NoviBuilder/novi-plugin-countdown)

Google Map Plugin

Map plugin allows you to change the map settings visually.



Plugin configuration:

querySelector (by default is set to ".google-map-container") - allows you to define the item processed by the plugin.

Editor panel:

- Trigger for the map settings display

- Trigger for the map settings display

Map settings panel:

"Map Center" field - allows you to set the coordinates of the map center.

"Zoom" field - allows you to define the zoom level.

"Style" select option - allows choosing the map styling (https://snazzymaps.com/ service)

"Marker Icon" and "Active Marker Icon" select options - allow you to set the marker for the default and active states.

Map pins settings panel:

"Marker Location" field - allows you to set the marker location

"Marker Description" field - allows you to specify the marker description (shows up in the active state)

"Add marker" button - allows to add the "Marker Location" and the "Marker Description" group for adding the new marker.

More detailed plugin documentation is available under the followinglink.
(https://github.com/NoviBuilder/novi-plugin-google-map)

Icons plugin

Allows you to replace the icons.

Plugin configuration:

querySelector (by default is set to ".novi-icon") - allows you to define the item processed by the plugin.

Editor panel:

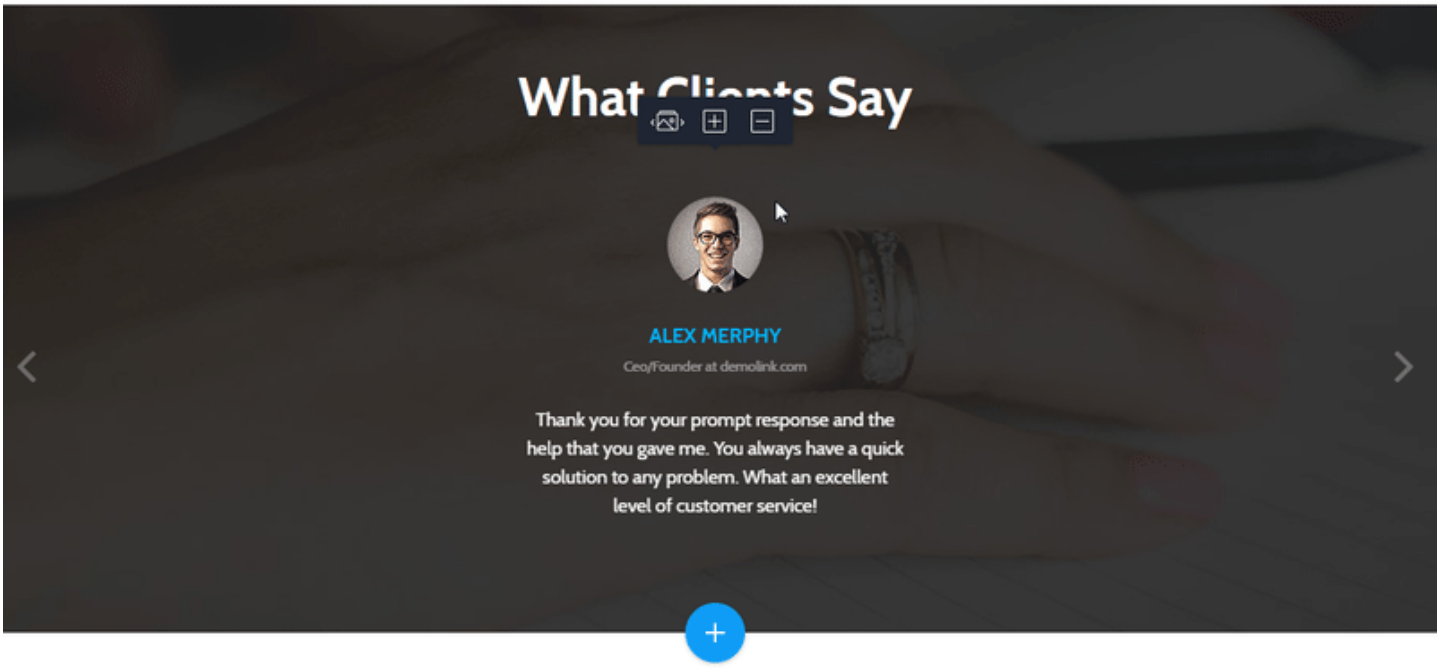
- Trigger for the icons manager display

Double-click the target item with the left mouse button or press the "Insert Icon" button to choose the icon for pasting.

The detailed plugin documentation is available at the following link. (https://github.com/NoviBuilder/novi-plugin-icon-manager)

Owl Carousel plugin

Allows you to configure the Owl carousel in the visual mode.



Our Plans

Plugin configuration:

querySelector (by default is set to ".owl-carousel") - allows you to define the item processed by the plugin.

Editor panel:

- Trigger for adding the slide (in case of hovering the carousel wrapper, the last slide will be cloned)

- Trigger for deleting the slide (in case of hovering the carousel wrapper, the last slide will be removed)

- Trigger for the carousel settings display

Carousel settings panel:

"Visible items" field - allows you to define the number of items to show up. You can make the number of the visible items vary depending on the screen size. In order to use this functionality, switch to the new screen size using the builder resizer tool and then specify the new value for the "Visible items" option.

"Indent between items" field - allows you to set the spaces between the carousel items. You can make the distance between the items vary depending on the screen size. In order to use this functionality, switch to the new screen size using the builder resizer tool and then specify the new value for the "Indent between items" option.

"Enable Loop" switcher - allows enabling/disabling the carousel loop. You can try it in the builder preview mode.

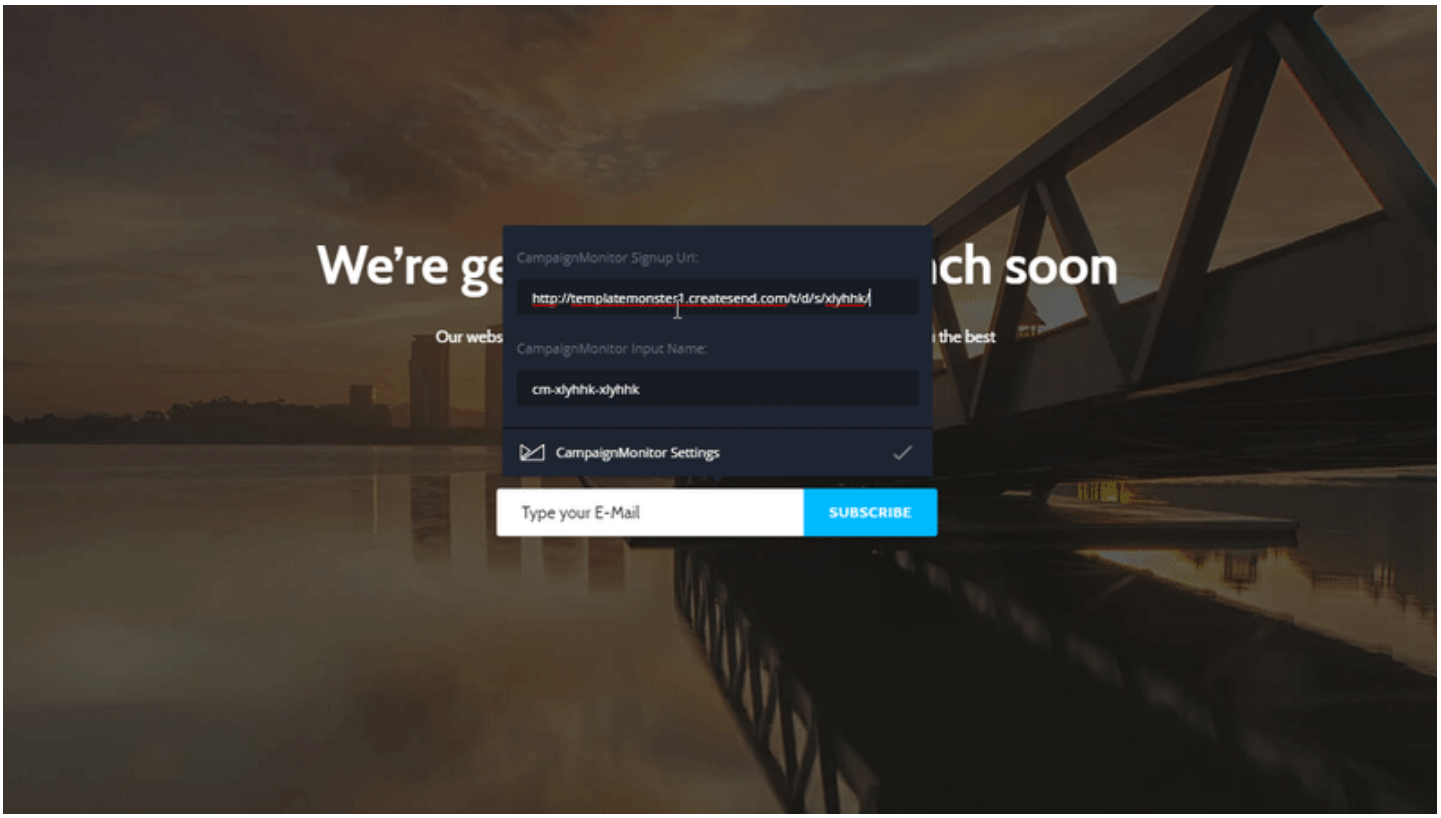
"Carousel Autoplay" switcher - allows enabling/disabling the carousel autoplay. You can try it in the builder preview mode.

"Carousel Autoplay Delay" field - allows you to set the sliding delay. It works only if the "Carousel Autoplay" option is enabled, and is available for testing in the builder preview mode.

The detailed plugin documentation is available at the following link.
(https://github.com/NoviBuilder/novi-plugin-owl-carousel)

Campaign Monitor Plugin

Allows you to configure the form to work with the Campaign Monitor service.



Plugin setup:

querySelector (by default is set to ".novi-campaign-monitor") - allows you to define the item processed by the plugin.

Editor panel:

- Trigger for the plugin display

Form configuration area:

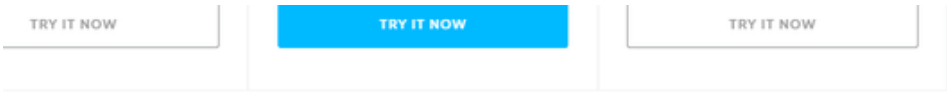
"Campaign monitor signup url" field defined the action of the form

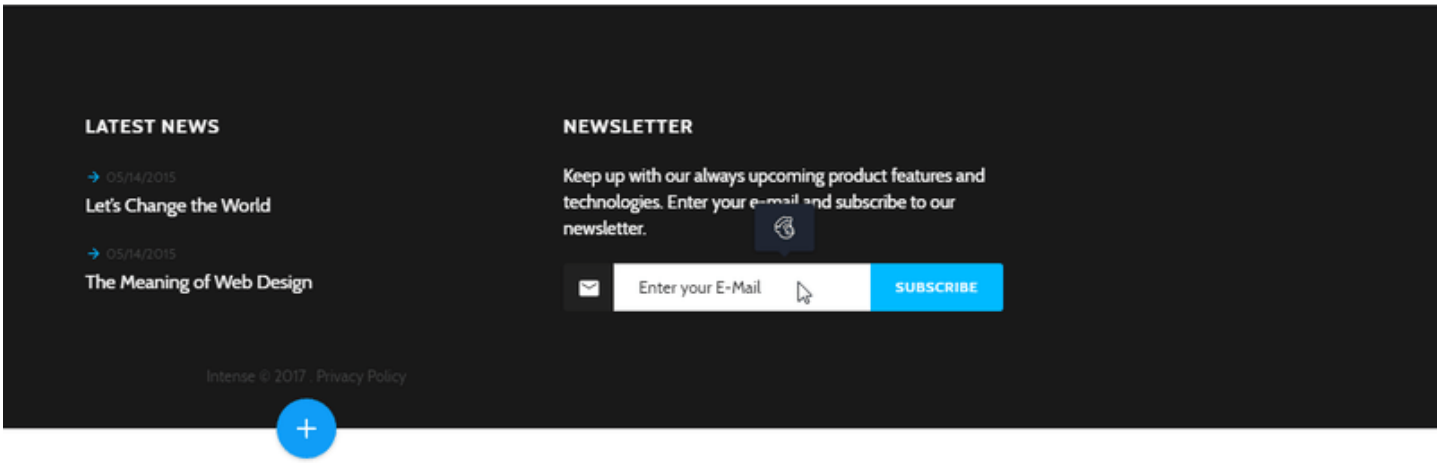
"Campaign monitor input name" field defines the value of the name attribute of the input field.

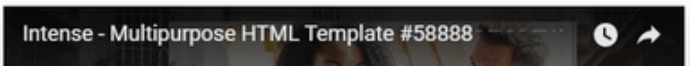
The detailed plugin documentation is available at the following link.
(https://github.com/NoviBuilder/novi-plugin-campaign-monitor)

Mailchimp Plugin

Allows you to configure the form to work with the Mailchimp service.







Plugin setup:

querySelector (by default is set to ".novi-mailchimp") - allows you to define the item processed by the plugin.

Editor panel:

- Trigger for the plugin display

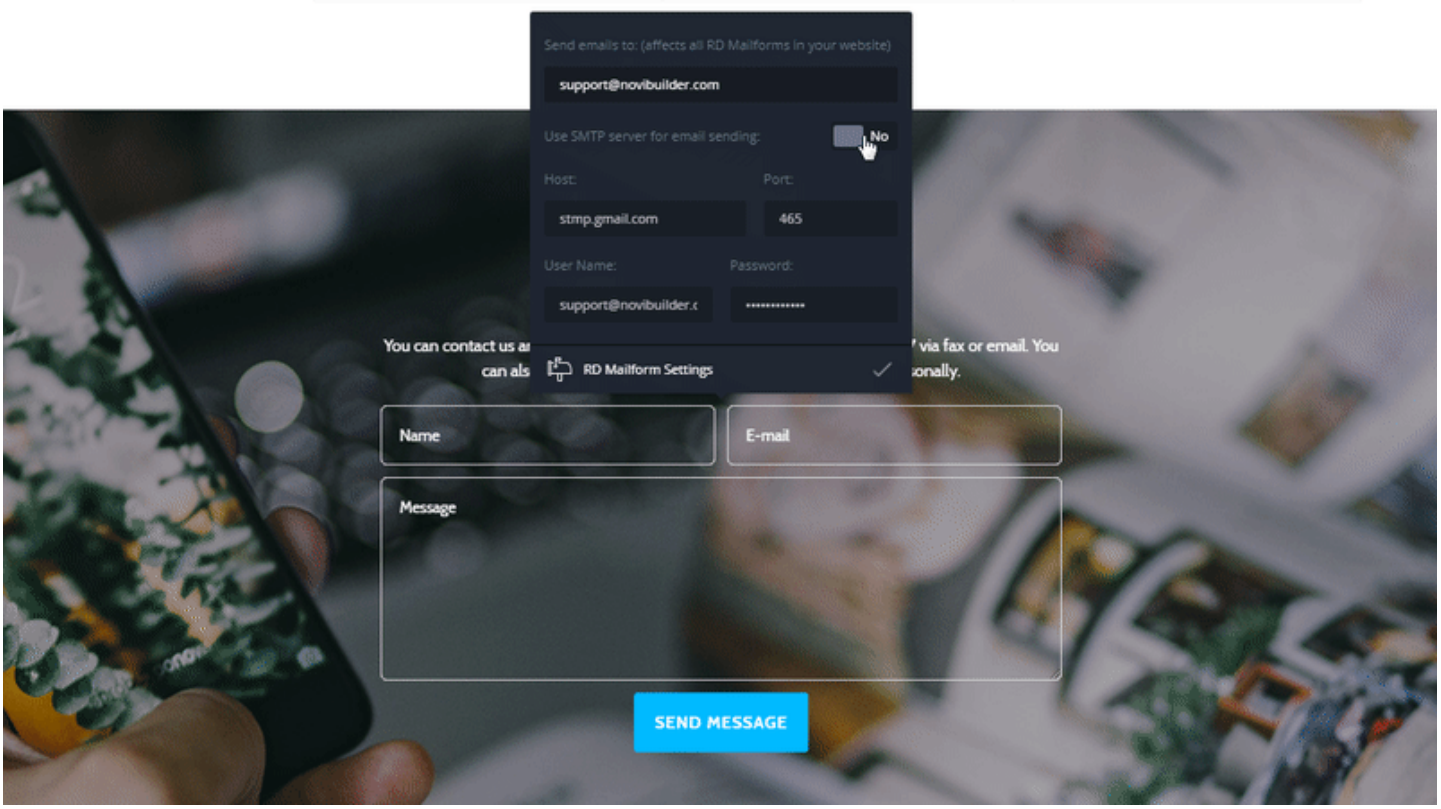
Form configuration area

"Mailchimp signup url" field indicates the action of the form

The detailed plugin documentation is available at the following link. (https://github.com/NoviBuilder/novi-plugin-mailchimp)

RD Mailform Plugin

Allows you to configure the RD Mailform form plugin using the visual editing tools.



Plugin setup:

querySelector (by default is set to ".rd-mailform") - allows you to define the item processed by the plugin.

configPath ("bat/rd-mailform.config.json") - allows you to specify the path to the form settings file.

Editor panel:

- Trigger for the plugin display

Plugin configuration area

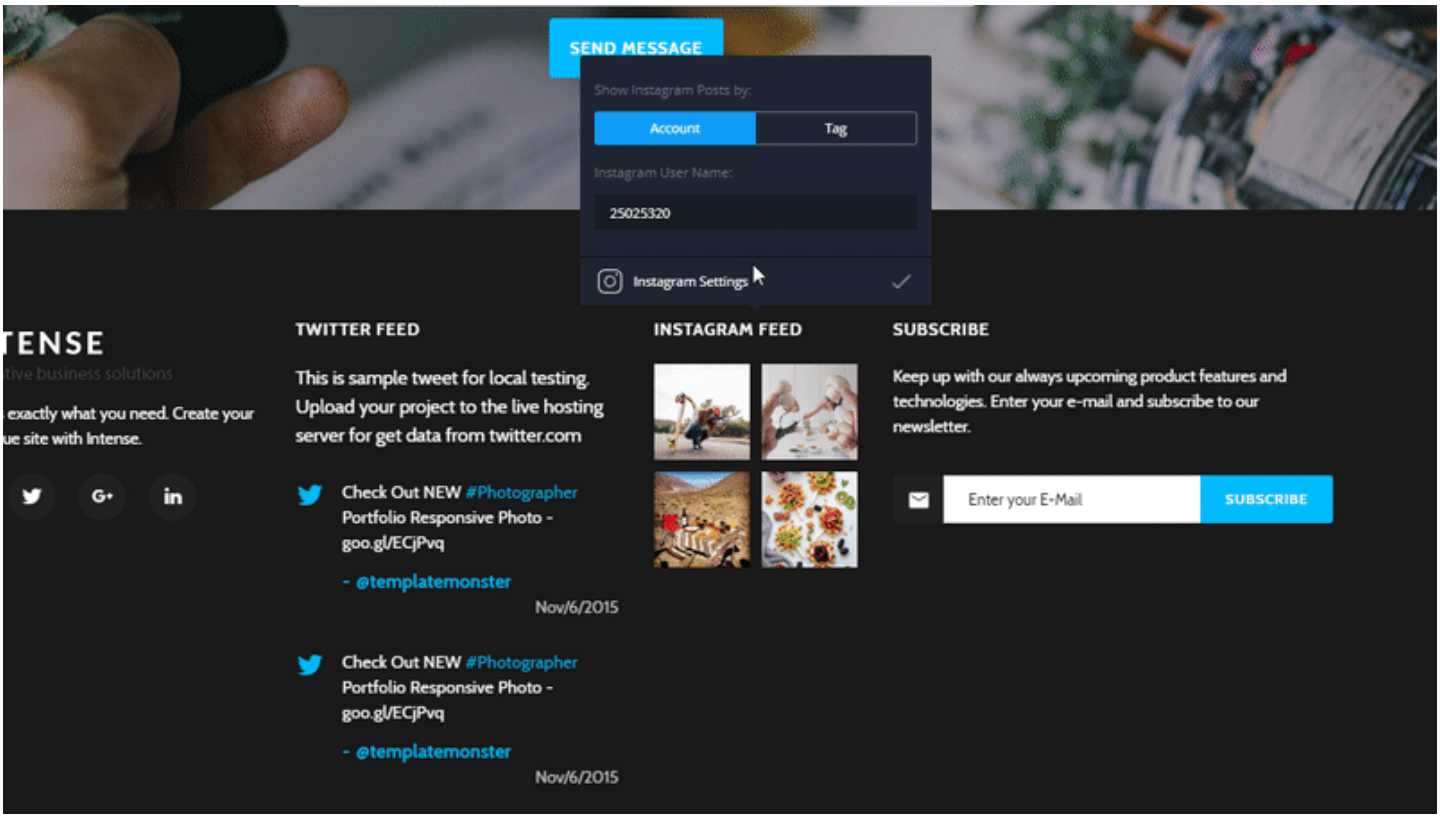
"Send emails to:" field indicates the email recipients.

"Use SMTP server settings for email sending:" option allows you to configure the sending of messages using SMTP.

The detailed plugin documentation is available at the following link. (https://github.com/NoviBuilder/novi-plugin-rd-mailform)

RD Instagram Feed Plugin

Allows you to configure the RD Instafeed instagram plugin using the visual editing tools.



Plugin setup:

querySelector (by default is set to ".novi-instafeed") - allows you to define the item processed by the plugin.

Editor panel:

- Trigger for the plugin display

Plugin configuration area

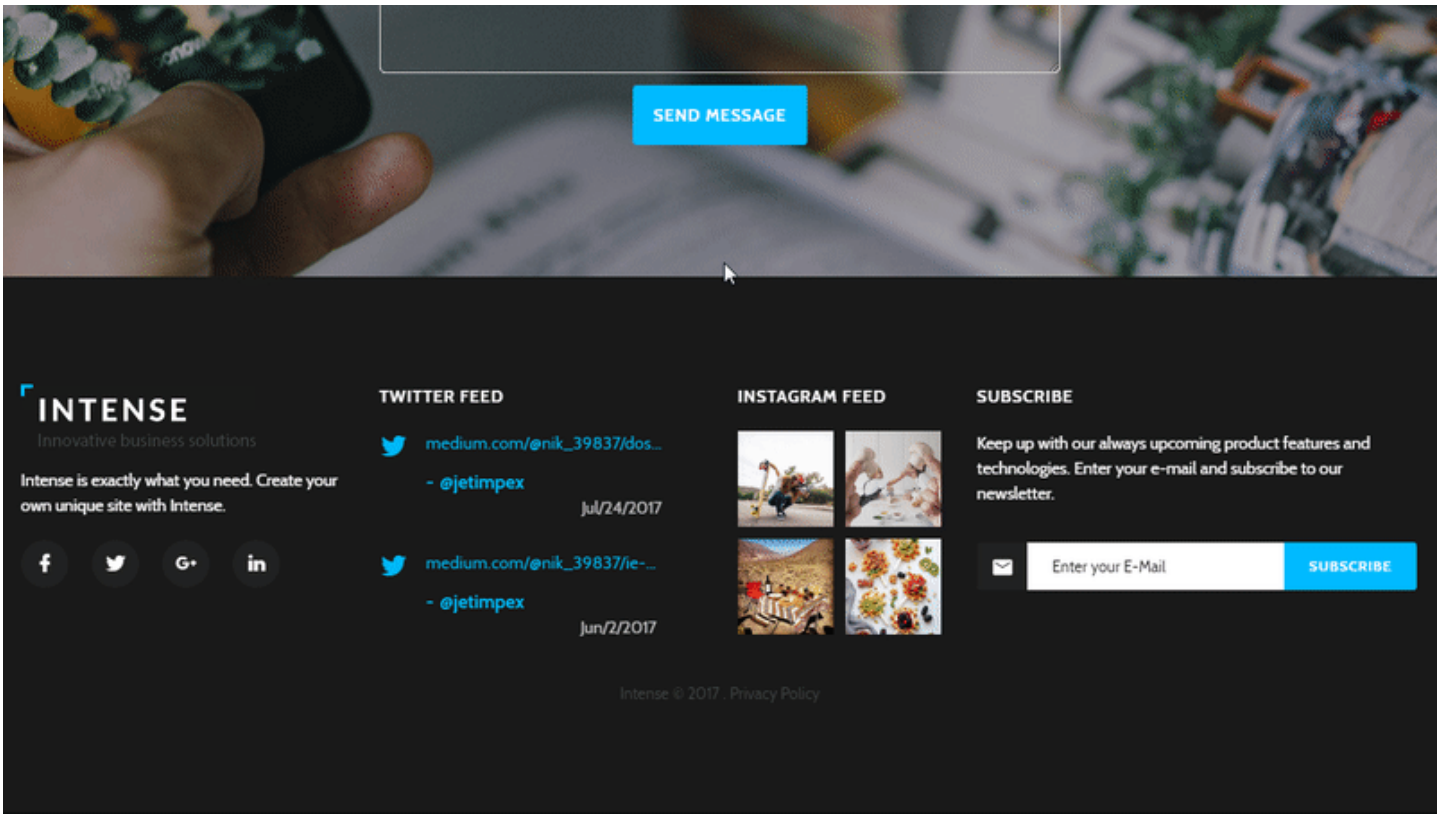
"Account" option allows to define the account ID to display the posts of.

"Tag" option allows you to specify a tag to display the posts for.

The detailed plugin documentation is available at the following link. (https://github.com/NoviBuilder/novi-plugin-rd-instafeed)

RD Twitter Feed Plugin

Allows you to configure the RD Twitter Feed twitter plugin using the visual editing tools.



Plugin setup:

querySelector (by default is set to ".novi-twitter") - allows you to define the item processed by the plugin.

Editor panel:

- Trigger for the plugin display

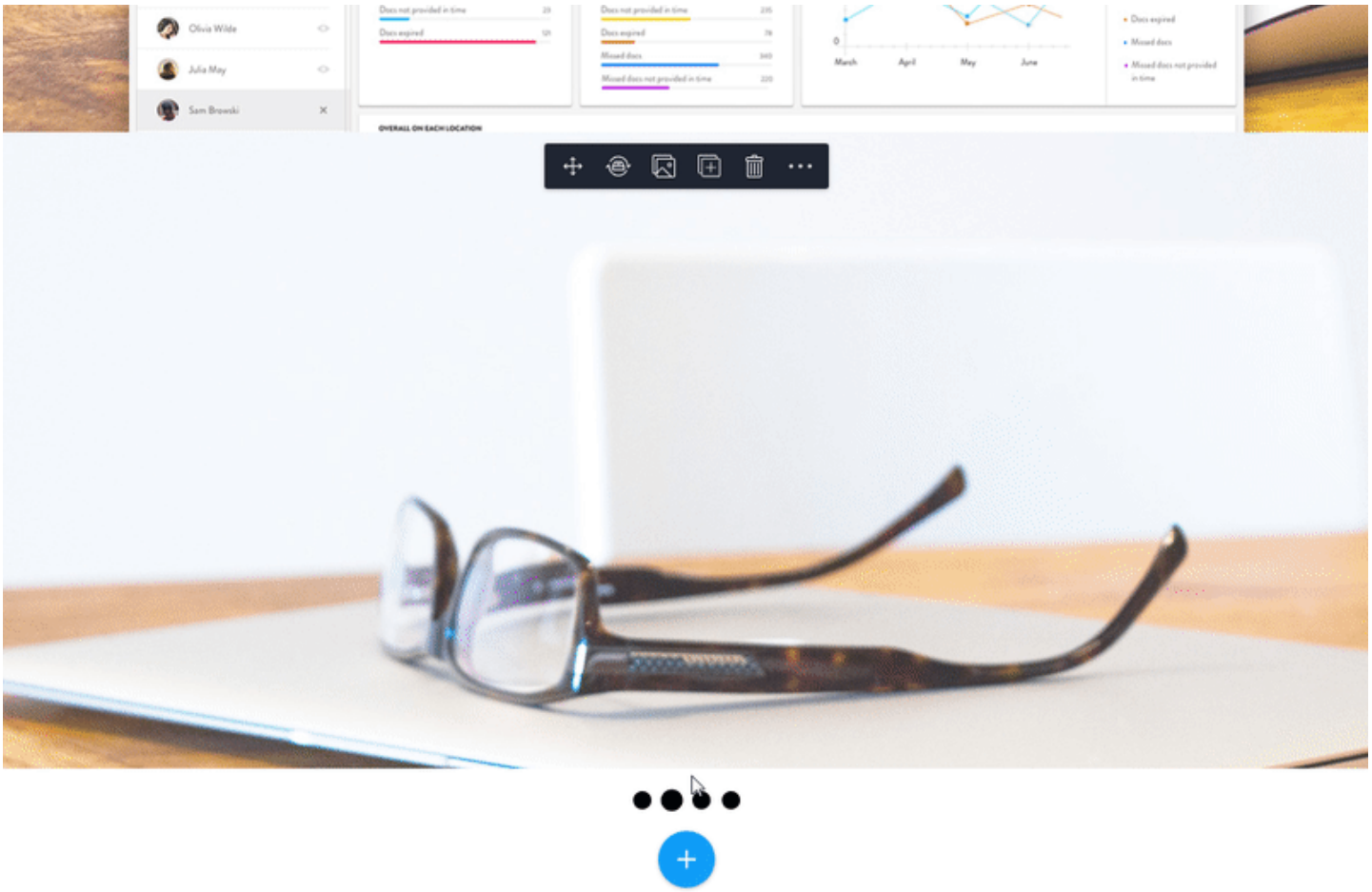
Plugin configuration area

"Twitter User Name" field specifies the author username, whose posts are displayed.

The detailed plugin documentation is available at the following link. (https://github.com/NoviBuilder/novi-plugin-rd-twitterfeed)

Camera Slider Plugin

Allows you to configure the jQuery plugin Camera slider with the help of visual editing tools.



Plugin setup:

querySelector (by default is set to ".novi-camera") - allows you to define the item processed by the plugin.

Editor panel:

- Trigger for the slide image replace option display

- Additional options (add/remove)

Image selection area (Media Gallery)

The image to be added is selected by double-clicking the left mouse button on the target image.

You can learn more about media gallery capabilities under "Media gallery" item (link) in the "Features" section.

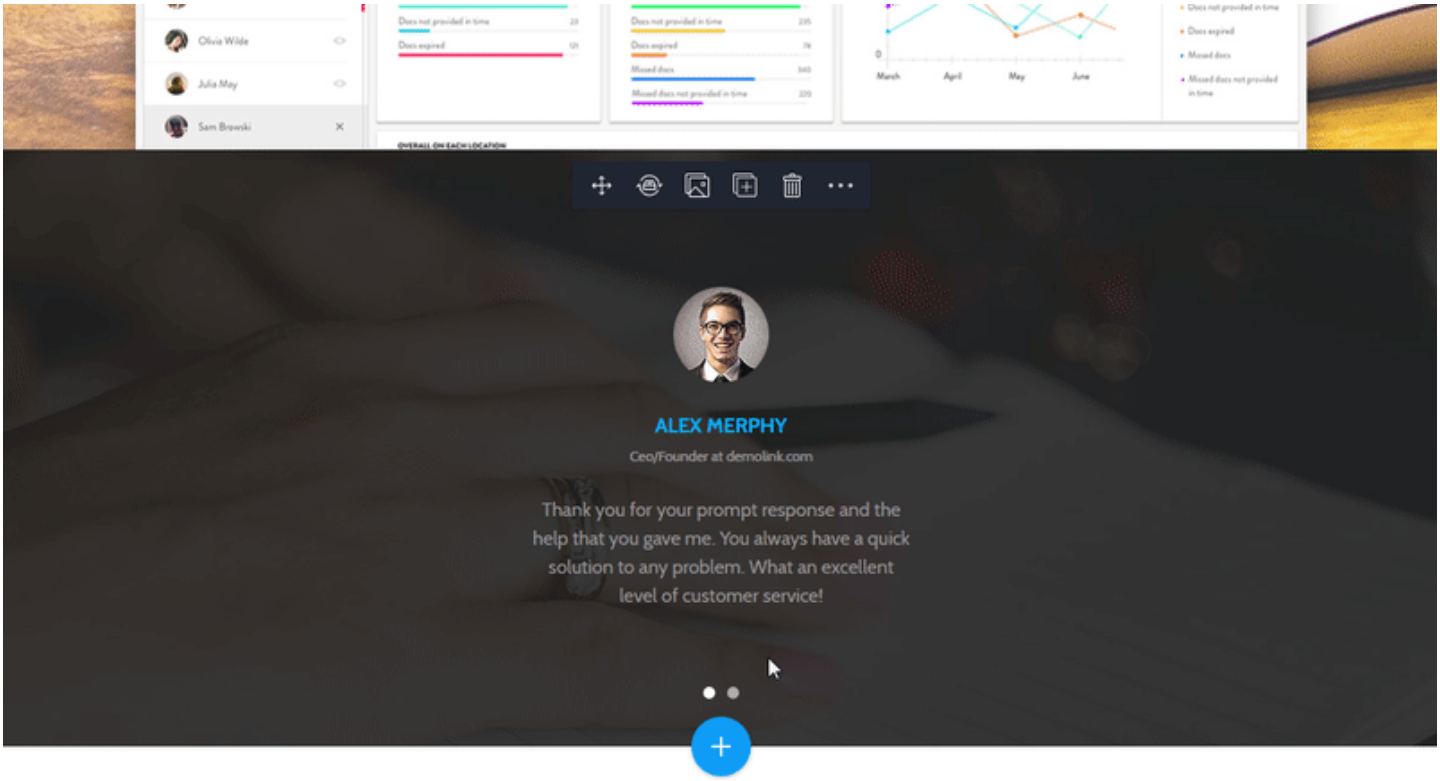
Image crop area:

"Keep original aspect ratio" setting is used for preserving the aspect ratio of the original image.

The detailed plugin documentation is available at the following link (https://github.com/NoviBuilder/novi-plugin-camera-slider)

Swiper Slider Plugin

Allows you to configure the Swiper Slider using the visual editing tools.



The Benefits of Intense

Plugin setup:

querySelector (by default is set to ".swiper-container") - allows you to define the item processed by the plugin.

effects("slide", "fade") - allows you to choose the set of effects for the animation on switching between the slides.

Editor panel:

- Trigger for the plugin display

- Additional options (add/remove slide, slider configuration)

Image upload area

"Choose file" button that allows to select an image

"Upload background image" button that allows to upload the selected image

Image selection area (Media Gallery)

The image to be added is selected by double-clicking the left mouse button on the target image.

You can learn more about media gallery capabilities under "Media gallery" item (link) in the "Features" section.

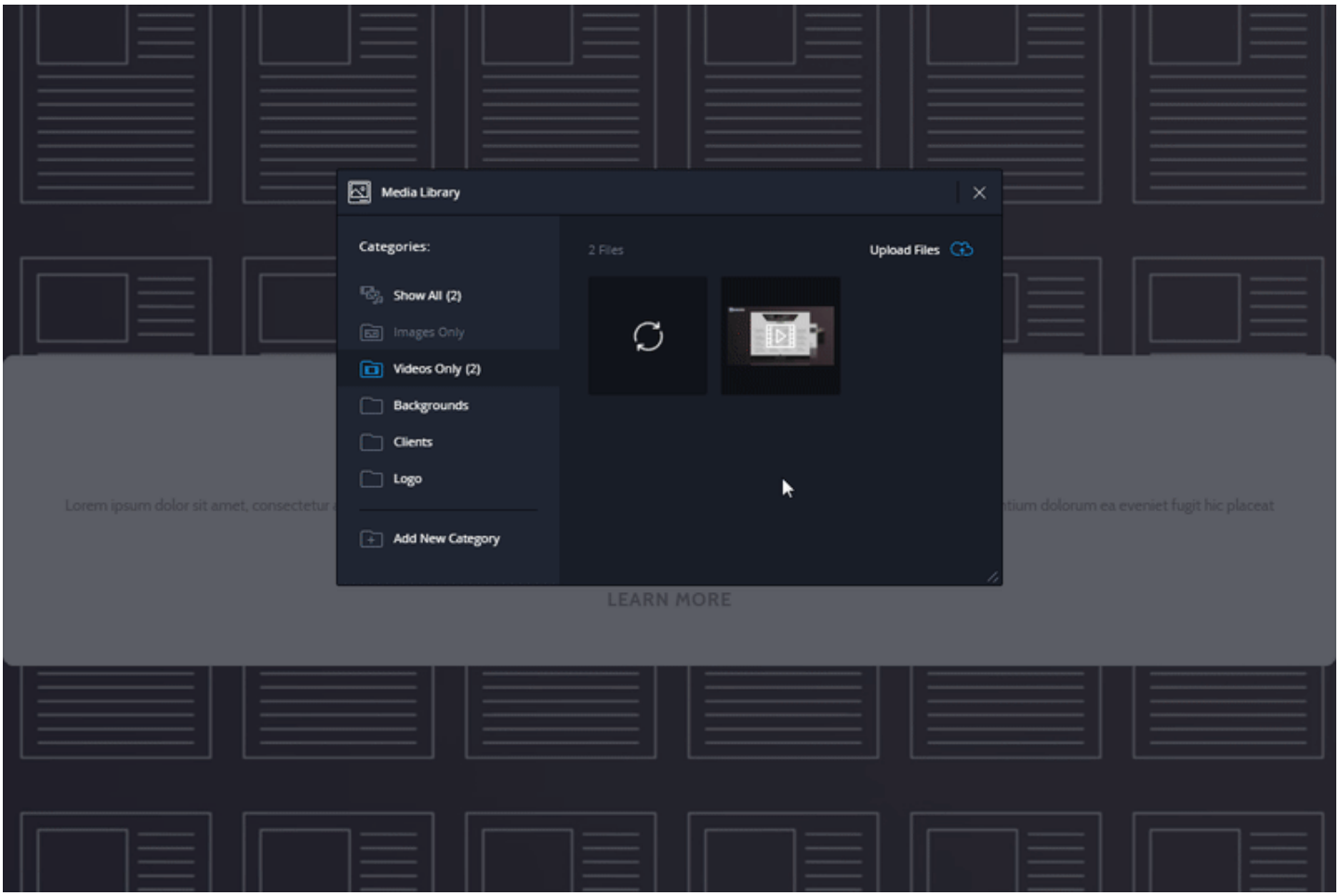
Image crop area:

"Keep original aspect ratio" setting is used for preserving the aspect ratio of the original image.

The detailed plugin documentation is available at the following link.
(https://github.com/NoviBuilder/novi-plugin-swiper-slider)

Vide Plugin

Lets you set up jQuery plugin Vide using visual tools.



Plugin setup:

querySelector (".novi-vide" by default) lets you define the element, to which the plugin functionality will be applied.

Editor panel:

- trigger for displaying a substitutional poster (displayed when the video can not be loaded).

- trigger for displaying a substitutional video.

Media selection area (Media Gallery)

The media file to be added is selected by double-clicking the left mouse button on the target image.

You can learn more about media gallery capabilities under "Media gallery" item (link) in the "Features" section.

Image crop area:

"Keep original aspect ratio" setting is used for preserving the aspect ratio of the original image.

More detailed plugin documentation is available under the followinglink. (https://github.com/NoviBuilder/novi-plugin-vide)

Import Template (Step by Step)

This section provides a step-by-step guidance on importing a third-party HTML template to Novi Builder and its subsequent set up.

The template import process consists of the following stages:

- Builder setup;
- Template import preparation;
- Preparing scripts for working with builder;
- Preparing styles for working with builder;
- Preparing HTML markup for working with builder;
- Project setup;
- System setup;
- Project preparation for use.

Builder Setup

Before you start working with builder, you should perform the initial setup. To do this, navigate to the builder tab, then go to the config directory.

Open config.json file in any text editor.

To continue, configure the following settings:

- jets - "true"
- demoMode - "false"

Template Import Preparation

Before importing the template, you should create an appropriate .zip file for importing it into builder.

In our case, we've got a template with the following structure:

site - template directory;

- **bat**- directory that contains all the needed php files;
- **css** - directory that contains template css styles;
- **fonts** - iconic fonts, integrated to the template;
- **images** - directory that contains template imagery;
- **js** - template js code;
- about.html - template HTML pages;
- blog.html
- blog-single-post.html
- contacts.html
- index.html
- news.html
- portfolio_grid.html
- portfolio_justified.html
- portfolio_list.html
- portfolio_masonry.html
- privacy.html

To import the template, prepare .zip archive file that contains all the contains of the 'site' directory.

Attention! There should be no additional parent directories in the .zip file.

Wrong:

site.zip

- site

bat

...

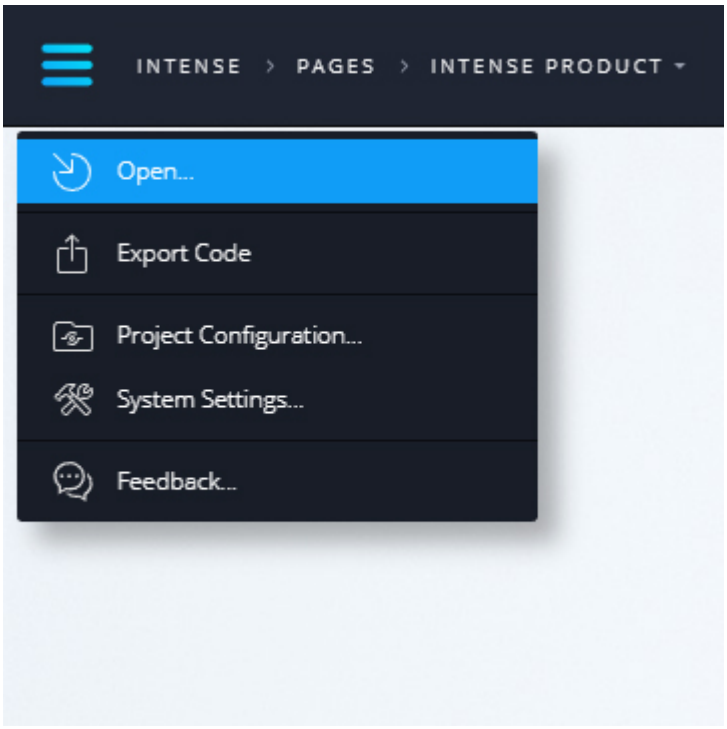
privacy.html

Correct:

site.zip

- bat
- ...
- privacy.html

So, you .zip file is ready and the builder is set up. Now, open the builder on the server, go to the main menu and choose "Open"



Choose the .zip file and upload it. Then, you'll be prompted to re-load the system. Accept the offer.

That's it! Your project is successfully imported to the builder.

Preparing Scripts for Working with Builder

Novi Builder is compatible with all js scripts, but some of them may require additional setup in Novi environment.

To set them up, use the xMode attribute, which is added to the 'window' object after the document loading.

The working logic or style display of your scripts may differ if the check is active:

```
if ( window.xMode ) {
  // Your JS code executed in Design Mode and Edit Mode of Novi Builder.
} else {
  // Your JS code that will not work in Design Mode and Edit Mode of Novi Builder.
}
```

Please, pay attention that the xMode attribute of window object will be available after the document is loaded, in this concern, the above-mentioned code listing will look as follows:

```
$document.ready(function () {
  var isNoviBuilder = window.xMode;
  if ( isNoviBuilder ) }
    // Your JS code executed in Design Mode and Edit Mode of Novi Builder.
  } else {
    // Your JS code that will not work in Design Mode and Edit Mode of Novi Builder.
  }
});
```

Cases, when the additional setup is necessary:

- Your script redefines the native
- Your script allows moving elements that contain text with connection events, such as mousedown, mousemove, mouseup. In this case, text editing is impossible. For example, slide switching with the help of mouse drag.
- Your script creates an unsynchronized copy of the element. Under the unsynchronized copy we mean an element, which does not change the initial element during editing and vice versa. In this case, the correct saving of the template is not guaranteed. For example, duplicating the navigation to implement a sticky navigation.
- Your script animates an element using the means of transform group. In this case, the element selection during and before the animation will be located in the resulting animation point. For instance, wow animation.
- Your script adds submit, click and other ajax logics while being processed. When editing this element (e.g., when editing button text) extra queries are sent. For instance, form submission without page reload (ajax).

Consider an example based on our template.

Our template contains the following scripts:

- RD Navbar (https://github.com/TemplatemonsterPlugins/rd-navbar)
- Swiper Slider (http://idangero.us/swiper/)
- Materialize Parallax (http://materializecss.com/parallax.html)
- RD Google Map
- RD Mailform (https://github.com/TemplatemonsterPlugins/rd-mailform)
- Isotope Filter (https://isotope.metafizzy.co/filtering.html)

RD Navbar (https://github.com/TemplatemonsterPlugins/rd-navbar) Setup

First, let's define the xMode attribute for using it inside the builder:

```
$document.ready(function () {
    var isNoviBuilder = window.xMode;
    ...
});
```

Now, let's turn off the duplication of the navbar sticky menu, due to the reasons, specified in the 3rd point of "Cases, when the additional setup is necessary":

```
 if (plugins.rdNavbar.length) {
    plugins.rdNavbar.RDNavbar({
        stickUpClone: (plugins.rdNavbar.attr("data-stick-up-clone") && !isNoviBuilder) ? plugins.rd
    })
}
```

Let's perform an additional setup to use the navbar inside the builder with convenience, namely:

- Turn off the sticky navigation;
- Remove following on anchor links;
- Turn off submenu/MegaMenu display on hover.

```
if (plugins.rdNavbar.length) {
    plugins.rdNavbar.RDNavbar({
        anchorNav: !isNoviBuilder, // turning off anchor navigation to make menu editing more conve
        stickUpClone: (plugins.rdNavbar.attr("data-stick-up-clone") && !isNoviBuilder) ? plugins.rd
        responsive: { // turning off sticky navigation to make editing the page on page scroll more
            0: {
                stickUp: (!isNoviBuilder) ? plugins.rdNavbar.attr("data-stick-up") === 'true' : fal
                },
            768: {
                stickUp: (!isNoviBuilder) ? plugins.rdNavbar.attr("data-sm-stick-up") === 'true' :
            },
            992: {
                stickUp: (!isNoviBuilder) ? plugins.rdNavbar.attr("data-md-stick-up") === 'true' :
            },
            1200: {
                stickUp: (!isNoviBuilder) ? plugins.rdNavbar.attr("data-lg-stick-up") === 'true' :
            }
        },
        callbacks: {
          onDropdownOver: function(){ // turning off the submenu/megamenu display on hover to enabl
            return !isNoviBuilder;
          },
    })
}
```

To be able to open submenus, add an element with "rd-navbar-submenu-toggle" class to the HTML markup to the level of the submenu element. In our case this is:

```
<ul class="rd-navbar-nav"> // main menu markup
    <li class="active">
        <a href="./">Home</a>
    </li>
    ...
    <li class="rd-navbar--has-dropdown rd-navbar-submenu"> // item that has submenus
        <span class="rd-navbar-submenu-toggle"></span> // our toogle for opening a submenu
        <a href="#" tabindex="-1">Portfolio</a>
        <ul class="rd-navbar-dropdown rd-navbar-open-left">  // submenu
            <li>
                <a href="portfolio_masonry.html" tabindex="-1">Masonry style</a>
            </li>
            ...
        </ul>
    </li>
</ul>
```

Swiper Slider (http://idangero.us/swiper/) Setup

First, you should turn off slide switching using mouse in accordance with the 2nd point of "Cases, when the additional setup is necessary":

```
if (plugins.swiper.length) {
    plugins.swiper.swiper({
        simulateTouch: s.attr('data-simulate-touch') && !isNoviBuilder ? s.attr('data-simulate-touc
        ...
    )}
}
```

Moreover, to edit the slide content more conveniently, let's turn off the slider autoplay.

```
if (plugins.swiper.length) {
    plugins.swiper.swiper({
        ...
        autoplay: isNoviBuilder ? false : s.attr('data-autoplay') ? s.attr('data-autoplay') === "fa
    )}
}
```

Materialize Parallax (http://materializecss.com/parallax.html) Setup

Let's turn off Parallax inside the builder for optimization:

```
if (plugins.materialParallax.length) {
    var i;
    if (!isNoviBuilder) { // Parallax initialization outside the builder
        plugins.materialParallax.parallax();
    }
    else { // adding background to a section in case we're in Novi
        for (i = 0; i < plugins.materialParallax.length; i++) {
            var parallax = $(plugins.materialParallax)
        ),
        imgPath = parallax.data("parallax-img");
        parallax.css({
            "background-image": 'url(' + imgPath + ')',
            "background-attachment": "fixed",
            "background-size": "cover"
        });
    }
}
```

RD Mailform (https://github.com/TemplatemonsterPlugins/rd-mailform) Setup

According to the 5th point of "Cases, when the additional setup is necessary", you should set up the form submission in the form script:

```
if (plugins.rdMailForm.length) {
    var i, j, k,
    ...
    for (i = 0; i < plugins.rdMailForm.length; i++) {
        var $form = $(plugins.rdMailForm[i]),
        formHasCaptcha = false;
        $form.attr('novalidate', 'novalidate').ajaxForm({
            data: {
            "form-type": $form.attr("data-form-type") || "contact",
            "counter": i
        },
        beforeSubmit: function (arr, $form, options) {
            if (isNoviBuilder) return false; // disabling form submission in builder
            ...
        }
    ...
}
```

Other scripts require no additional setup.

Preparing Styles for Working with Builder

There are cases, when you need to style an element inside the builder differently, e.g. show the block content, which is displayed on hover, to enable its editing.

In such cases, use data-attribute data-x-mode="true", added to tag.

```
html[data-x-mode="true"] body{ // selector that determines styles inside the builder
    ...
    font-size: 20px; // style set for the selector
    ...
}
```

One more important aspect of adding styles is that your project inside the builder resides in an iframe.

To avoid the incorrect calculation of the element height with the height:100% property, you should specify:

```
html[data-x-mode]{
    height: 100%;
}
```

Moreover, to provide for the space to insert a container, add the following styles to your "Page container" (check out the ‘Project Setup’ section):

```
.page{
    height: 100vh;
}
```

Preparing HTML Markup for Working with Builder

There are no restrictions or rules in terms of markup creation. Below, you'll see the recommendations, which will make the import process and subsequent template use easier for you.

It's important to understand that your markup can be changed by the visual editor, so pay attention to the following list of recommendations:

- Expect that the the amount of content in your blocks will change. So, center and justify the content if necessary.

that columns and sections may change places, be added or deleted. For li instance, centering the columns will 'save' you in case the user removes a column.

- Add a container for page content, inserting all your sections into it, and having footers and headers on the same level. This will help you when you'll be creating layers (read about them in the subsequent sections).
- The blocks, in which content will be dragged, should be wrapped in one parent container to make creating layers more hassle-free.
- Create the system of offsets and positioning that is as much universal as possible.

In our template, let's start with the page container. Currently, it looks the following way:

```
<body>
  <div class="page text-center">
    <header class="page-header">...</header>
    <main class="page-content">
      <section>...</section>
      ...
      <section>...</section>
    </main>
    <footer>...</footer>
  </div>
</body>
```

Following the guidelines, let's make it look the following way:

```
<body>
  <div class="page text-center">
    <header class="page-header">...</header>
    <section>...</section>
    ...
    <section>...</section>
    <footer>...</footer>
  </div>
</body>
```

Project Setup

Project setup is available in the main builder menu and includes:

- Project Name
- Publishment Path
- Image Directory Path
- Video Directory Path
- Page Container
- New Page Template
- Project Layers

Now, let’s take a more in-depth look into each of these points.

Project Name

Takes care of the project name, which is displayed on the main Novi panel.

Publishment Path

Specify the publishing path on the server containing builder. The path should be relative, the root directory is our directory with the builder.

For example, if you want it to be published to the directory next to the builder, specify the following path:

../publish-folder

Image Directory Path

Define the path to the images of your project. This path is necessary for media gallery functioning. After adding a media file to your project, this file is moved to this directory.

The path should be relevant. The root folder is your project folder.

In our case, according to the project structure, this path is:

images/

Video Directory Path

This path is set up in the same way as "Image Directory Path", except for the fact that here you specify the path for video files:

video/

Page Container

This setting allows to specify the css selector that defines the main page container.

The primary container is the wrapper for the page container, by default this is the <body> tag.

In our case, .page is the container.

New Page Template

This section takes care of how a new page looks in the builder.

It’s very important to understand that, in case a Page Container is empty when a new page is created, the user is prompted to select a preset to start working with the page, otherwise the page is immediately displayed.

Let’s take the index page markup as a basis:

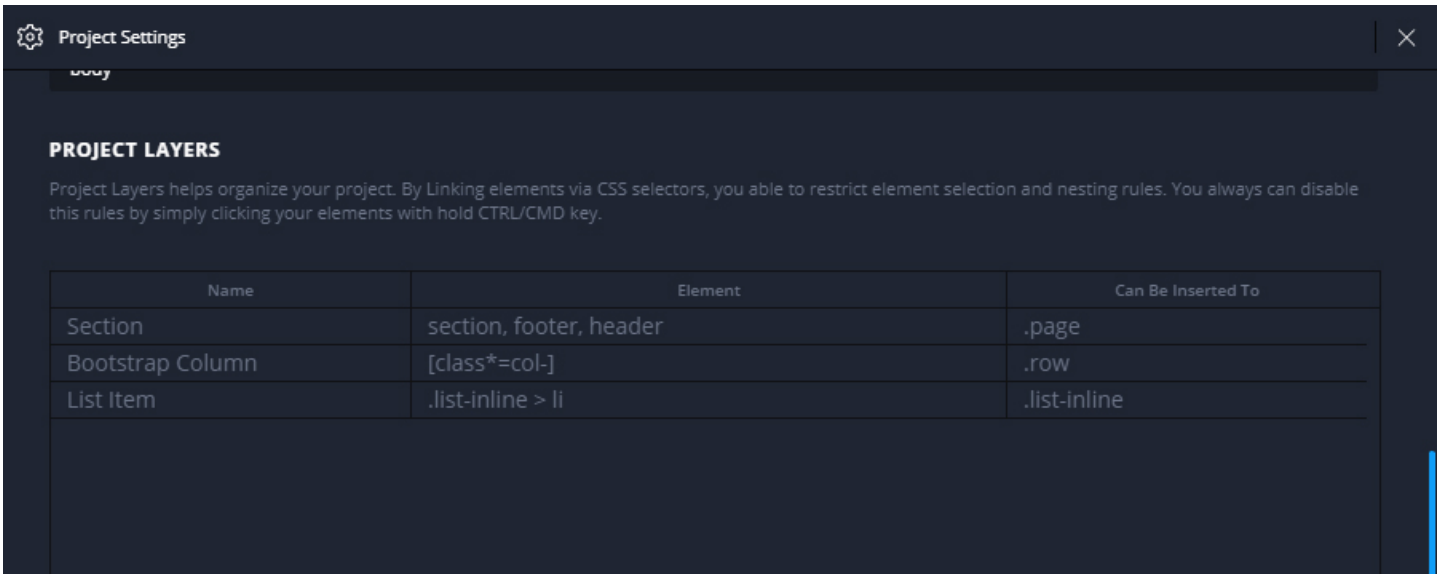
```
<!DOCTYPE html>
<html lang="en" class="wide smoothscroll wow-animation">
<head>
    <title>Home</title>
    <meta name="format-detection" content="telephone=no">
    <meta name="viewport" content="width=device-width, height=device-height, initial-scale=1.0, ma
    <meta http-equiv="X-UA-Compatible" content="IE=Edge">
    <meta charset="utf-8">
    <link rel="icon" href="images/favicon.ico" type="image/x-icon">
    <link rel="stylesheet" type="text/css" href="//fonts.googleapis.com/css?family=Roboto:300,400,5
    <link rel="stylesheet" href="css/style.css">
    <!--[if lt IE 10]><div style="background: #212121; padding: 10px 0; box-shadow: 3px 3px 5px 0 r
</head>
<body>
    <div class="page text-center">
    </div>
    <script src="js/core.min.js"></script>
    <script src="js/script.js"></script>
</body>
</html>
```

Project Layers

Layers are a very important for template import and influence its subsequent use in the Novi Builder.

Prior to describing layers, it’s recommended to take a look at the Novi layer system, which is described in the "Project settings" section.

Let’s add a couple of rules:



The first rule will help you manage sections.

The second manages Bootstrap columns.

The last one takes care of inline list elements.

The number of rules is not limited. In case a couple of rules concern the same element, the last rule in the table will be applied.

System Setup

System setup is available in the main builder menu and includes:

- Startup/Shutdown
- Text Editor

Startup / Shutdown

The section of system settings allows enabling interactive presentation on first builder launch and turning on checking for builder updates.

In our case, "Show tips on startup" and "Check for updates" are enabled.

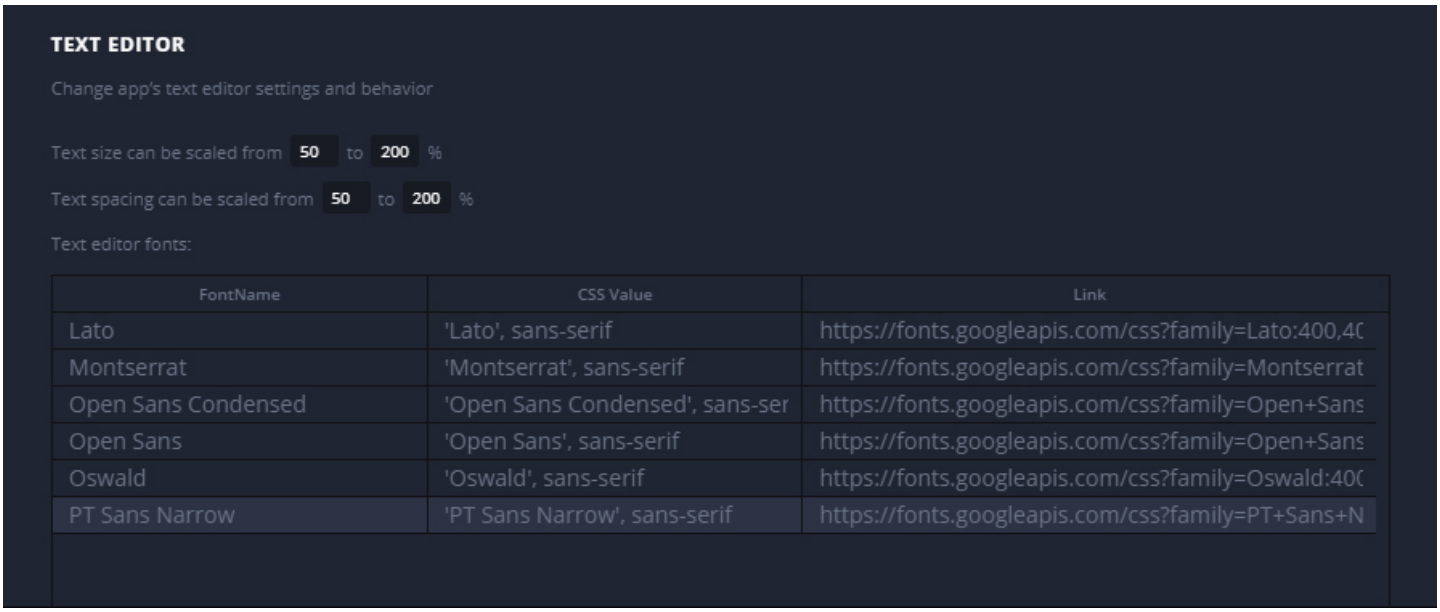
Text Editor

In this section, the setting of a text editor are available.

You can set the range of text size and line spacing by means of a text editor.

Also, it is possible to connect the font from a remote service, e.g., Google Fonts.

In our case, the fonts used in the template are already specified in the list of default fonts. Let’s set the following ranges:

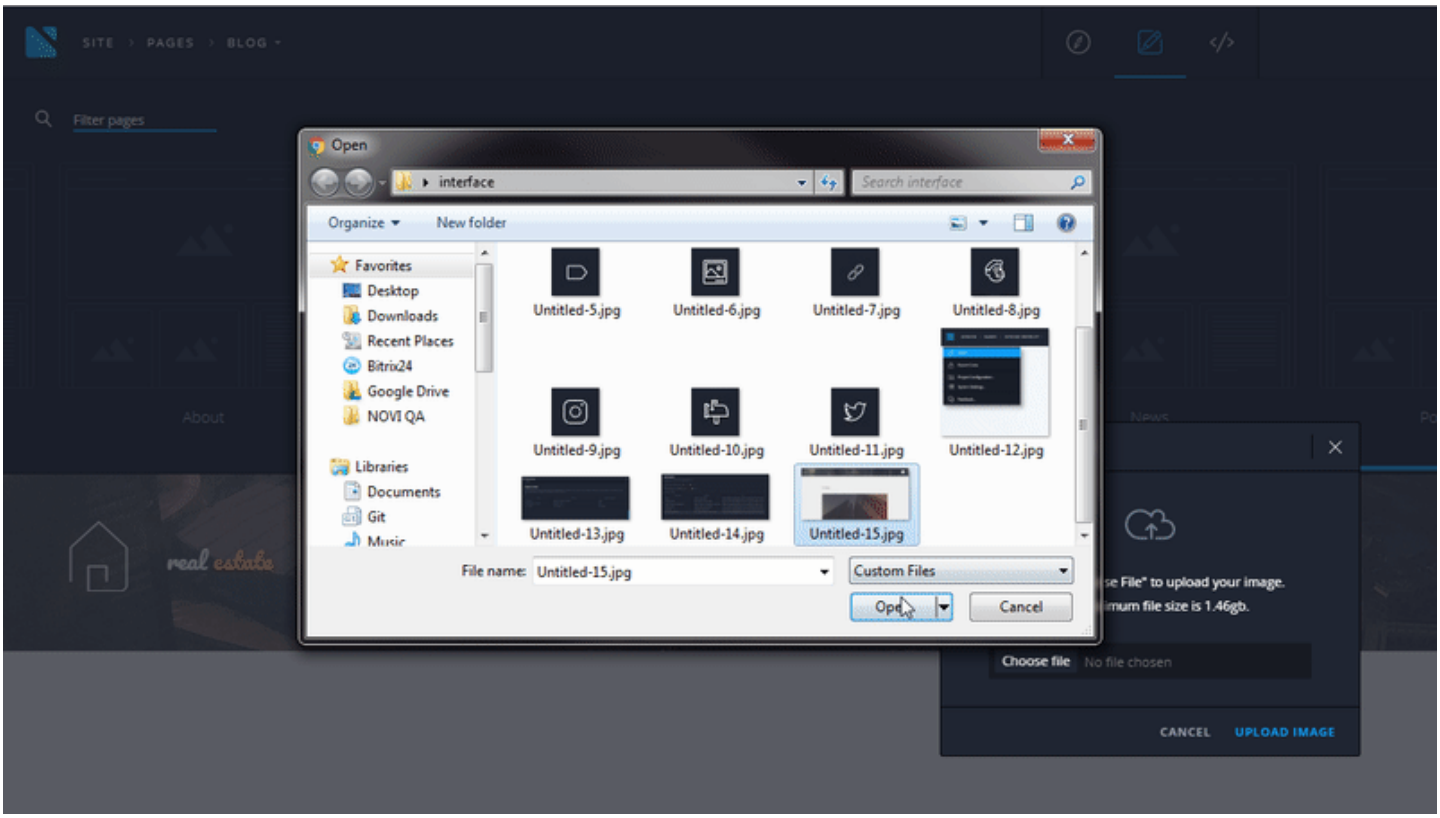


Project Preparation for Use

We've set up the project, specified all the necessary system settings, prepared the markup, js scripts and css.

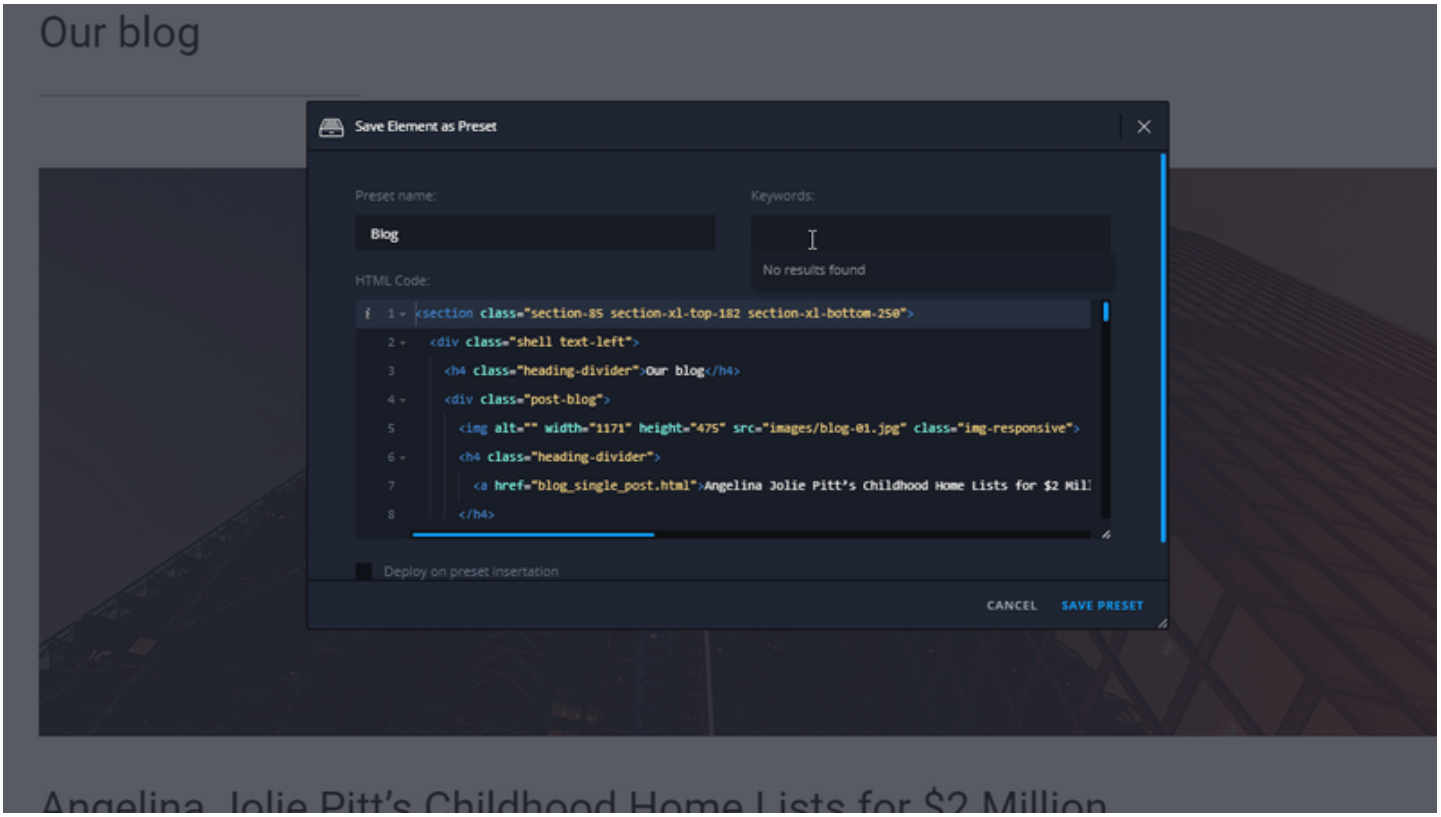
The next step is adding content to the project: we'll prepare presets, set up pages, upload necessary previews.

First, let's add previews for our pages. That's not hard. Let's use a page manager:



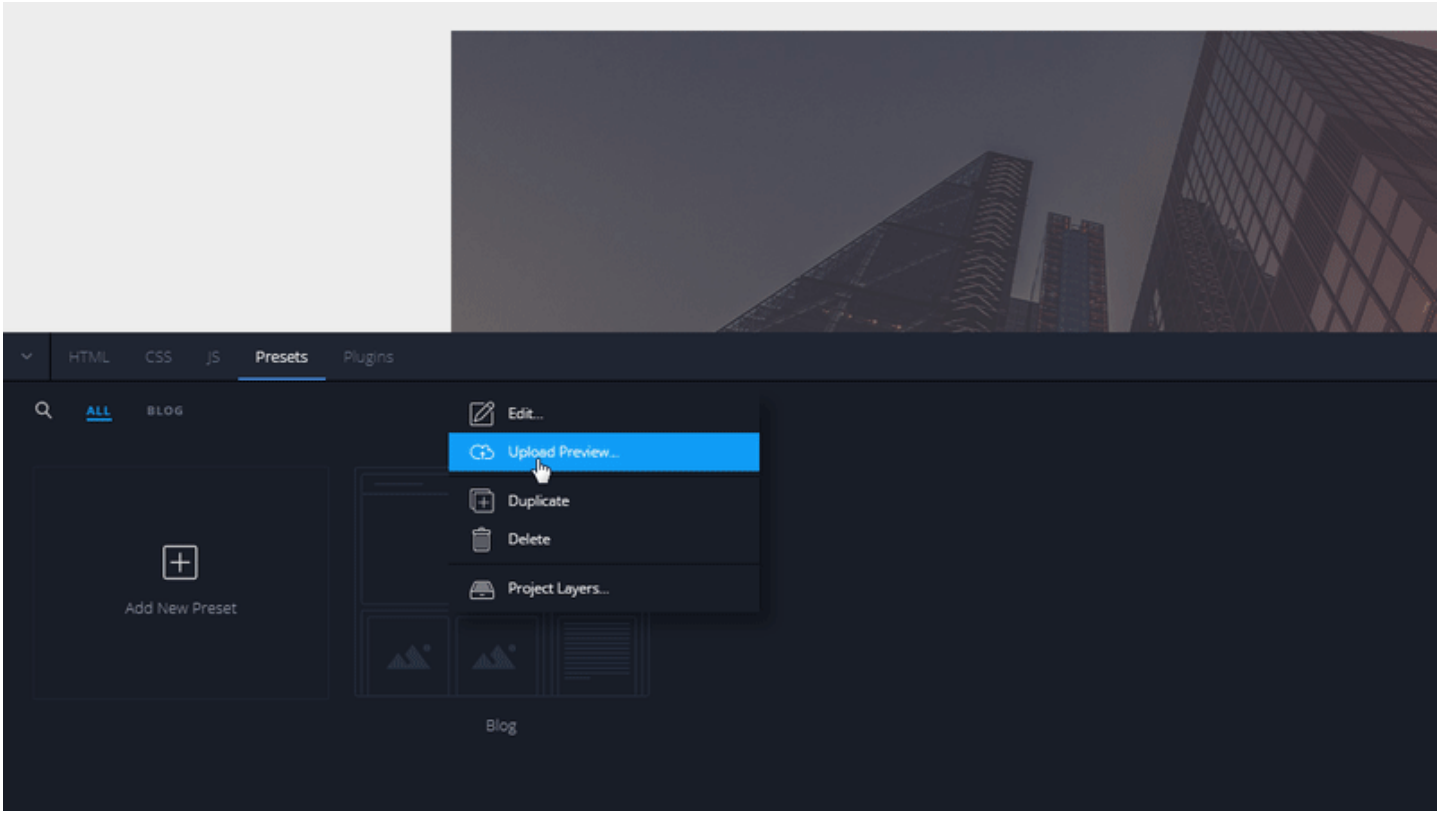
Now, let's add presets to our project: you can do this using a context menu of the visual editor or using a preset manager, which is located on the developer panel.

Here is an example of adding a blog preset:



You can add as many presets as you wish, creating filtering categories in the meanwhile.

The last step is adding previews to your presets. Here is an example, based on the created blog:



We've performed all the necessary steps for importing a third-party template.