

Joomla GSoC 2021 Project

Core Enhancements

Student

Yatharth Vyas

Mentors

Christiane Maier-Stadtherr

Niels Braczek

Timeline

Placing Modules in Menu	June 6 to June 27
Modules in the Articles-Edit-View	June 28 to July 9
Improve Menu List view (Menu Items)	July 10 to July 24
Merge Featured into Article	July 25 to August 15

Contents

1. Placing Modules in Menu	3
My Research:	3
Proposed Approach:	4
Summary of the flow:	4
Place Modules via Frontend (New Flow Upto Step 5)	4
Place Modules via Backend: (Same as old but Step 3 is a new enhancement)	5
Part I: Module Preview (Button in Frontend + Link from Backend)	6
Frontend	7
Backend	8
Part II: Global Configuration Setting in Modules to enable Frontend Placement	9
Proposed Timeline (3 weeks)	11
2. Improve the Incorporation of Modules in the Articles-Edit-View	12
Aim of the Task	12
Scope for Improvement:	12
Proposed Approach:	13
New Tab for Module(s) Imported	13
Inline Edit Module link for TinyMCE Editor	13
Extra Button to Add Module in Module Selection	15
Proposed Timeline (2.5 weeks)	16
3. Improve Menu List view (Menu Items)	17
Aim of the Task	17
Proposed Approach	17
Proposed Timeline (2 Weeks)	19
4. Improve Content: Merge Featured into Article	20
Proposed Approach	20
Proposed Timeline (3 weeks)	21

1. Placing Modules in Menu

My Research:

I looked up tutorials on placing modules in Joomla to understand where the users need a UX upgrade and which parts are confusing:

- ❖ First Search Result for Tutorial: [How to display Joomla! module on the frontend?](#)

This tutorial instructs the users to enable the 'View positions' settings and then visit the Frontend to understand where the positions are (preview) and then go back to the admin panel to place a module.

Scope for Improvement:

- Somehow reduce the number of times the user has to switch between backend and frontend to see the positions.
- There is no direct way for the website admin to know where the module will be placed and how it will look. I.e. A preview is missing

- ❖ Joomla Forum Discussion: [CREATE modules on front end](#)

This discussion appears in the top 5 search results for 'Joomla Add modules to frontend' Google search. Here many people have discussed the feature that allows them to directly place modules via frontend (without backend) and it seems that many users are interested in this feature.

Scope for Improvement:

- Ability to directly add modules via frontend. This will be similar to the add modules button in the Backend that opens the add module modal.

- ❖ Additional Notes:

- The re-ordering of Menu Items is a little confusing and if a feature was provided to easily re-order the components via Frontend directly or via a draggable list in the backend would be a really good upgrade. Although I feel this might have to be reconsidered for accessibility reasons, I think we should keep this for the end if we finish the other Project tasks before the schedule.
- While Placing modules via Backend, maybe a modal or some dummy image provided by the template owner ([for example](#), this image will be optional), showing a preview of all positions could be helpful. Or maybe just a Link that redirects to Frontend with the [Show Positions Setting](#) Enabled.

Proposed Approach:

Summary of the flow:

Place Modules via Frontend (New Flow Upto Step 5)

Think of Frontend Editing to understand this in a similar analogy:

1. Enable Frontend Module Placement from the com_modules configuration
2. Visit Frontend and log in as Administrator (or any role that has permissions to create and edit Modules)
3. You will find an 'Add Module' button at the bottom of the page's main content. Click on this and it will append ?pm=1 to the URL (ie. Switch to the place modules view)
4. In the Place Modules view, you will be shown all positions and a Button: 'Add Module here' in each of these positions. This will give the user a preview of where each position points to
5. On selecting any of the place modules buttons, a Modal/View would contain the 'Add Module' modal which is seen in Admin Panel.
6. Now in this modal, the admin user selects a module
7. Next, he reaches the Module's form page where the Position and Menu Assignment will be pre-filled and the user can add a Title/Note, etc, and submit to save the module.
8. And now the module is successfully placed.

Place Modules via Backend: (Same as old but Step 3 is a new enhancement)

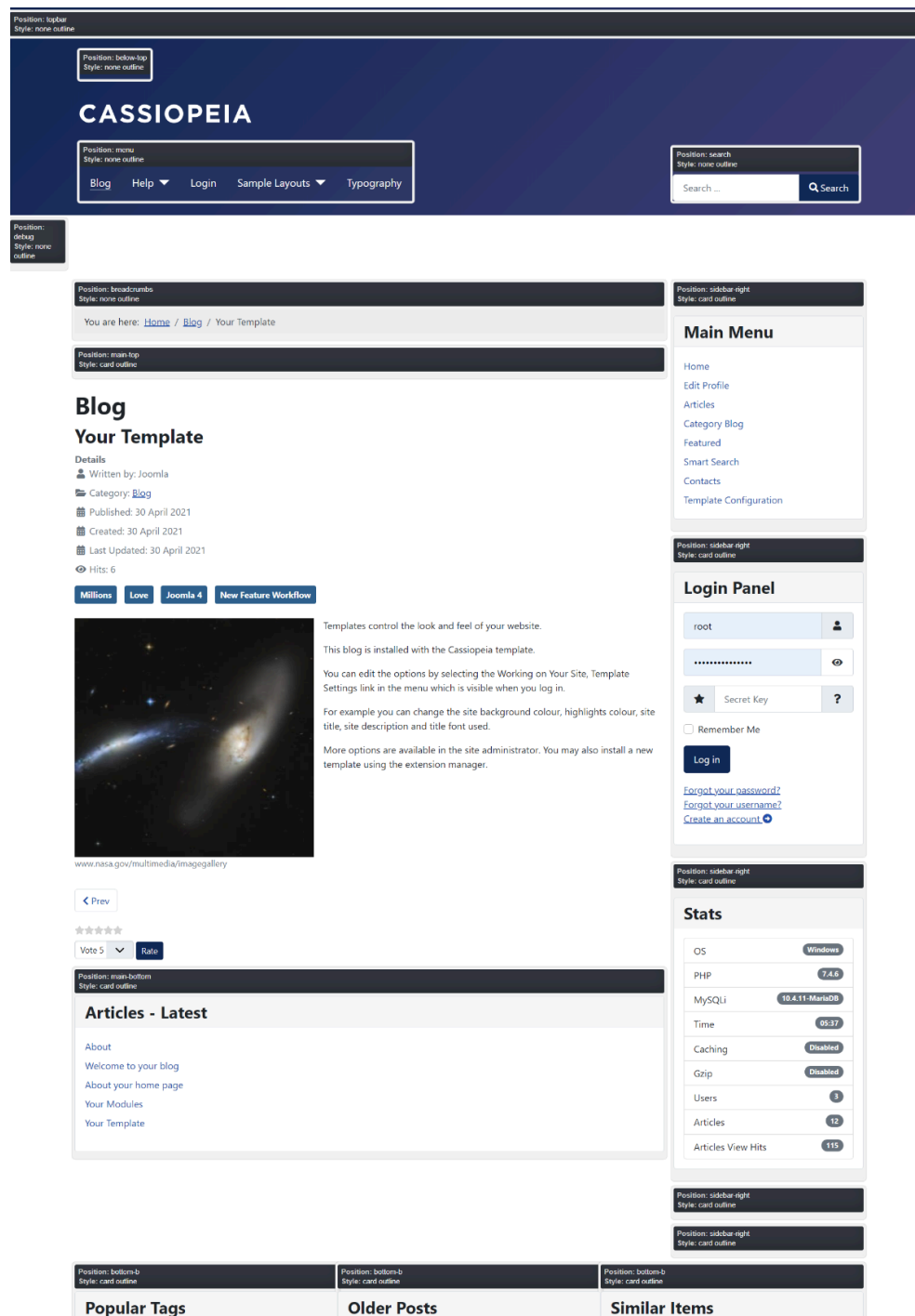
Except for Step 3, Everything is the same as it used to be, I didn't feel like anything could be improved in the existing Backend Module Add flow. Step 3 adds an extra feature for the users to preview each position in a single click rather than navigating to the frontend, adding ?tp=1 (template positions parameter), and then back to the backend while remembering the position name their minds.

1. Visit the Add Modules Page
2. Select the Module
3. In the Module form, you will see a link below the 'Position' dropdown. This link will show the frontend page with all positions in a Modal. Users can now select any position from the preview and it will update the dropdown's selection. This solves the issue of not knowing where each position place on the Frontend.
4. Fill the remaining form
5. Save and Close

This has been explained in more details in the next section →

Part I: Module Preview (Button in Frontend + Link from Backend)

When you Enable the Place Modules Option in Template's Global Configuration Options, a Preview, as shown above, is displayed to the user when **?tp=1** is passed to the page URL.

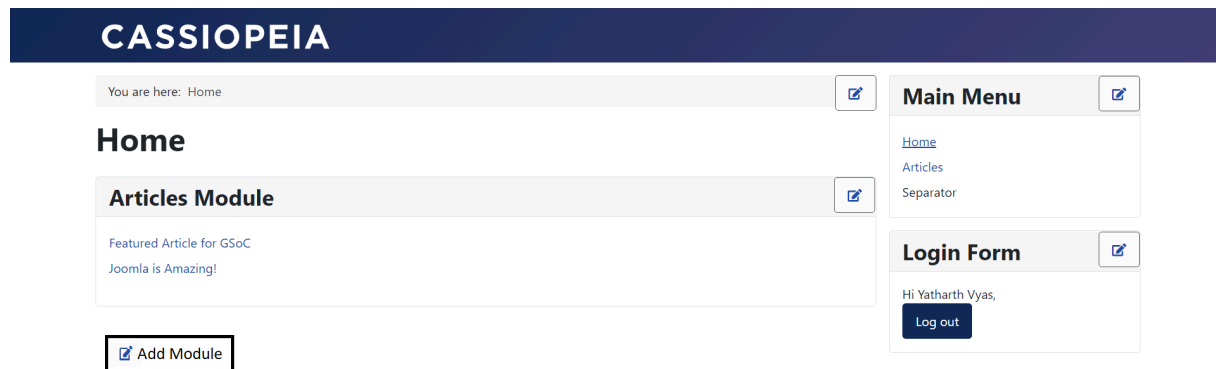


A button (link) for **Add Module** could be added to Frontend and a button (link) to **Select Module Position** could be added to Backend. When pressed, this would open up the add Module to

Menu Modal and this modal would automatically set the value of Position and Menu to the selection based on which button was pressed.

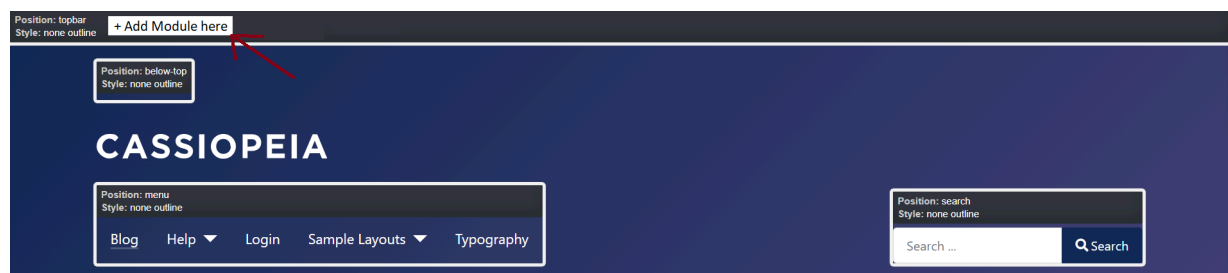
Frontend:

A button at the bottom of the page to Add Module (Similar to Atum's backend dashboard)



The Show Template Positions view is reached by passing **?tp=1** parameter of any page. Similar to this, we could add a parameter like say: **index.php?pm=1** (pm= place modules) and only if the user is logged in and has permissions, we will show the Positions along with the Add Module button as in the image below.

The Add Module button will then show the same page with **?pm=1** passed as a get parameter to the URL. Unlike the below image, This will show **all positions** with Add Module Here button



This **'Add Module here'** button (shown above) will open up the original Add Module Modal where the user will follow the steps in the old flow, ie. first select a Module type, and then the form will be displayed with Menu Assignment and Position inputs pre-filled based on selections.

The Module's Menu Assignment will be pre-filled as

Page on Frontend	Menu Assignment of the Module
index.php	All Pages
Any other view (specific menu)	the specific menu only

Backend:

For Module Position, add a link to select the position from preview so the user doesn't have to navigate to the Frontend and then back to the backend while remembering the position name:

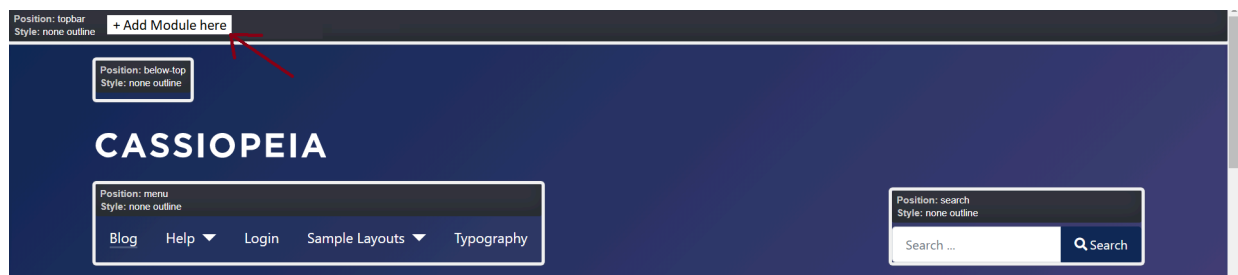
Before navigating to the frontend, a dropdown having list of all template names will have to be selected by the user and this value will be passed to the URL as

index.php?tp=1&template=custom-template-name so the preview is obtained of the custom-template-name.

The screenshot shows the Joomla! Module Manager configuration for the 'Articles - Newsflash' module. The interface includes tabs for 'Module', 'Menu Assignment', 'Advanced', and 'Permissions'. The 'Module' tab is active, showing the following settings:

- Category:** Joomla! (dropdown menu)
- Tags:** Type or select some tags (dropdown menu)
- Show Article Images:** ☒ Show (checkbox with label 'Show the images that are inside the text of the article.')
- Show Intro/Full Image:** Intro Image (dropdown menu)
- Title:** ☐ Hide (checkbox)
- Position:** top-a (dropdown menu with a link to 'Select a position from Preview')
- Status:** Published (dropdown menu)
- Start Publishing:** (calendar icon)

For Example, The link will redirect the user to the menu's frontend view (Maybe as a Modal?) and the user will have the option to select a position, ie. The button for 'Add Module Here' will be rephrased to 'Select Position'. **Note:** Ask User to select a template in case of Multiple



Implementation Ideas:

When you select a button

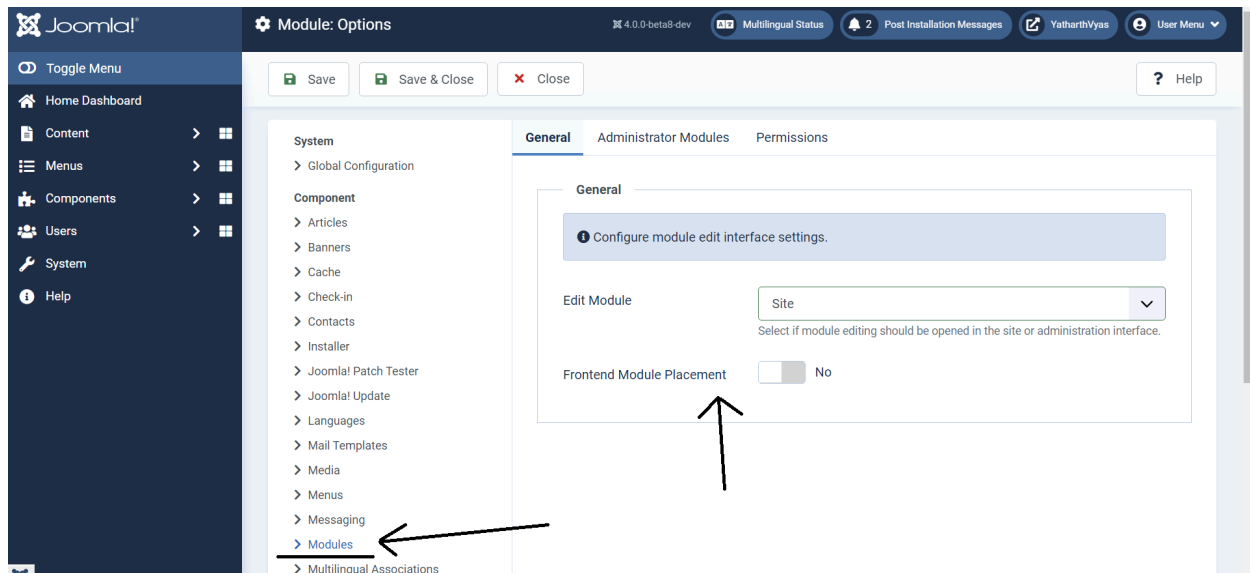
- Use data from Modal iframe to manipulate parent window's DOM: [example](#)
- <https://stackoverflow.com/a/9006960/11913141>
- We can listen to Modal close event to reload parent window if required: [Reference](#)

Part II: Global Configuration Setting in Modules Template to enable Frontend Placement

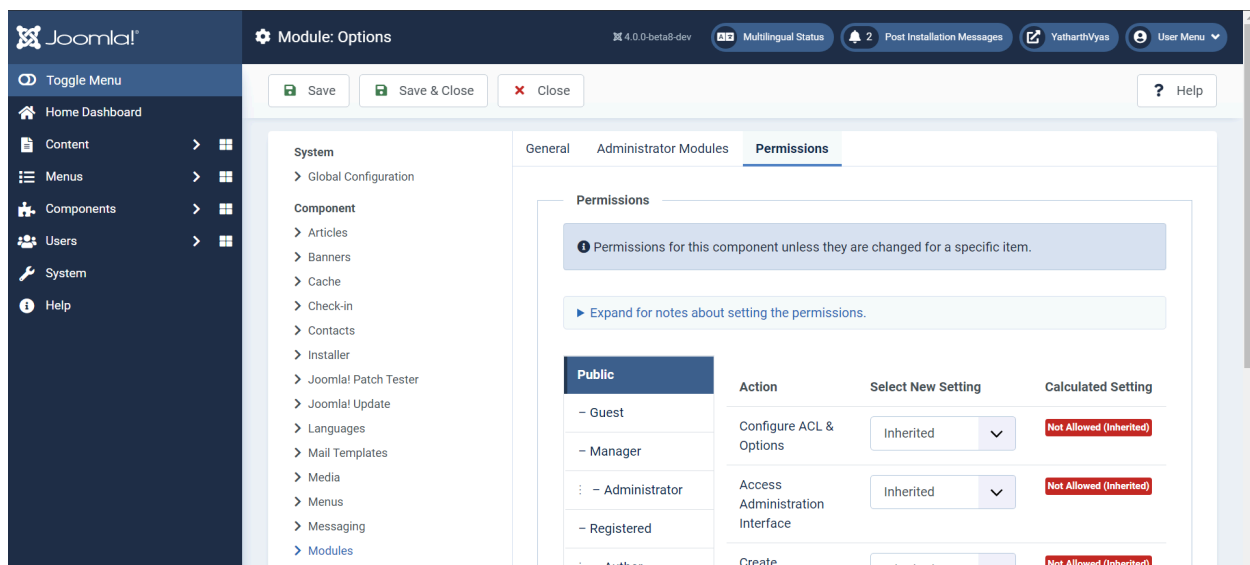
(similar to Frontend Inline Editing but because this is specific to Modules, it will be a part of the Modules Category's config.xml)

A configuration option to toggle Frontend Module Placement should be provided to the users.

A **switcher field** could be added to the Global Configuration page under com_modules. This boolean will be used to decide whether to show the 'Add Module' button in the frontend or not.



The permissions for the Module Placement will be taken from the Permissions for com_modules set in Global Configurations. So only those users who are logged in and belong to one of the permitted categories will be able to use the feature:



Also,

On selecting the Frontend Add Module button, the Add Module Modal should open up based on the selection of this dropdown below. Hence this will have to be rephrased.

Edit Module

Site

▼

Select if module editing should be opened in the site or administration interface.

The Label and Description for this will have to be rephrased to.

Module Modal

Site

▼

Select if module modal should be opened in site or administrator interface

because the modal is now common to add and edit.

Maybe the mentors could suggest a better rephrased statement here.

Proposed Timeline (3 weeks) (✓= DONE)

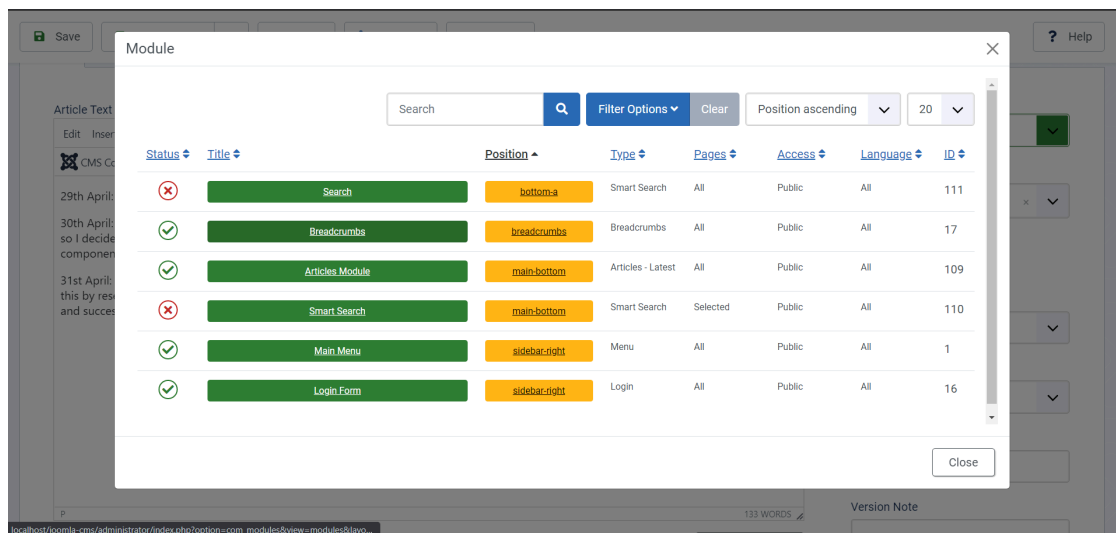
Adding Switcher Toggle to Global Configuration for Enabling Frontend Placements ✓	June 6	1 day
Rendering Add Module Button in Frontend conditionally based on the global configuration option ✓	June 7	1 day
Setting up ?pm=1 and rendering buttons and positions for it ✓	June 8 to June 9	2 days
Passing data of position and menu assignment (pages) from the button to Add Module page ✓	June 10 to June 17	1 week
Adding Preview link in backend's add/edit module page	June 18 to June 20	3 day
Setting up the Modal for Backend		
Setting up the frontend view for the Modal and returning the data	June 21 to June 22	2 days

I want to keep the next 5 days, June 22 to June 27 as a backup if I'm ever unable to do something on a workday or if I encounter some unexpected bug.

2. Improve the Incorporation of Modules in the Articles-Edit-View

Aim of the Task

The current way to add modules to articles is by opening the CMS dropdown and selecting modules to reach this screen:

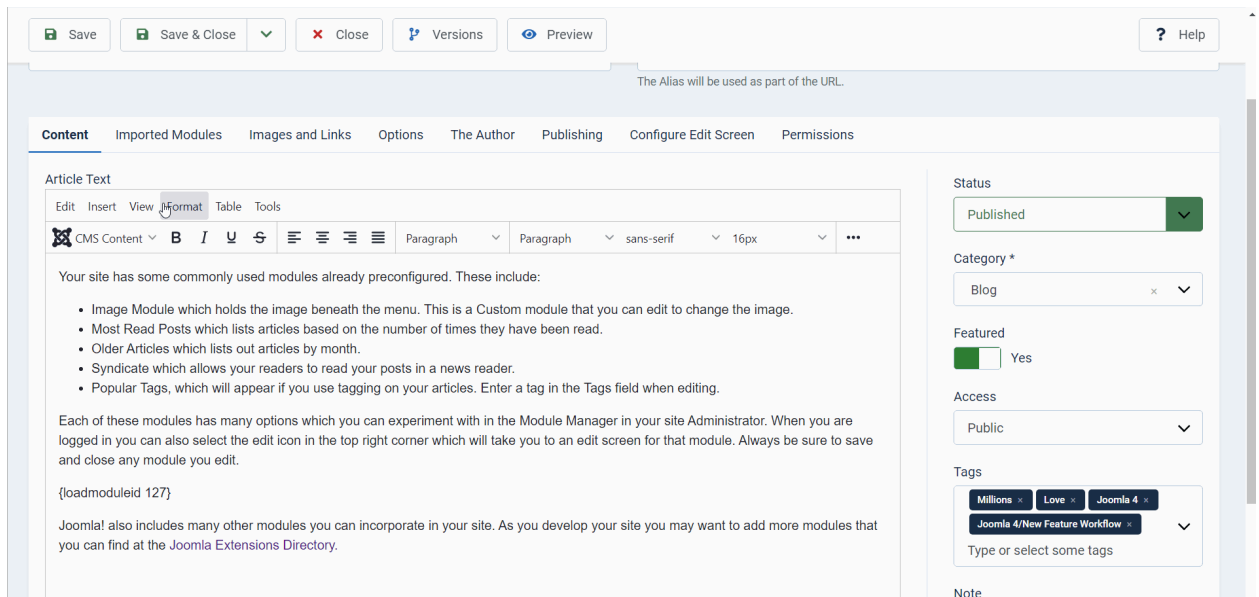


Scope for Improvement:

- There is no option to add a new module. Possible Enhancements are:
 - Creating a button that would either link to a new page where the user could add the new module
 - Opening it in the modal (I believe the first option is more preferable since it would make a UI less congested and allow reusability of code)
- Previously added modules have no edit option. Possible Enhancements are:
 - Place an edit button in the text box next to {loadmoduleid 127} that would open modal. This modal will reuse the edit.php view of that module.
 - Adding an 'edit' hyperlink in the textbox that would open that module's edit view in a new tab. It will also have the module id in the URL to map to the specific instant.
- Files to be modified: **administrator\components\com_content\tmpl\article**

Proposed Approach:

1. New Tab for Module(s) Imported

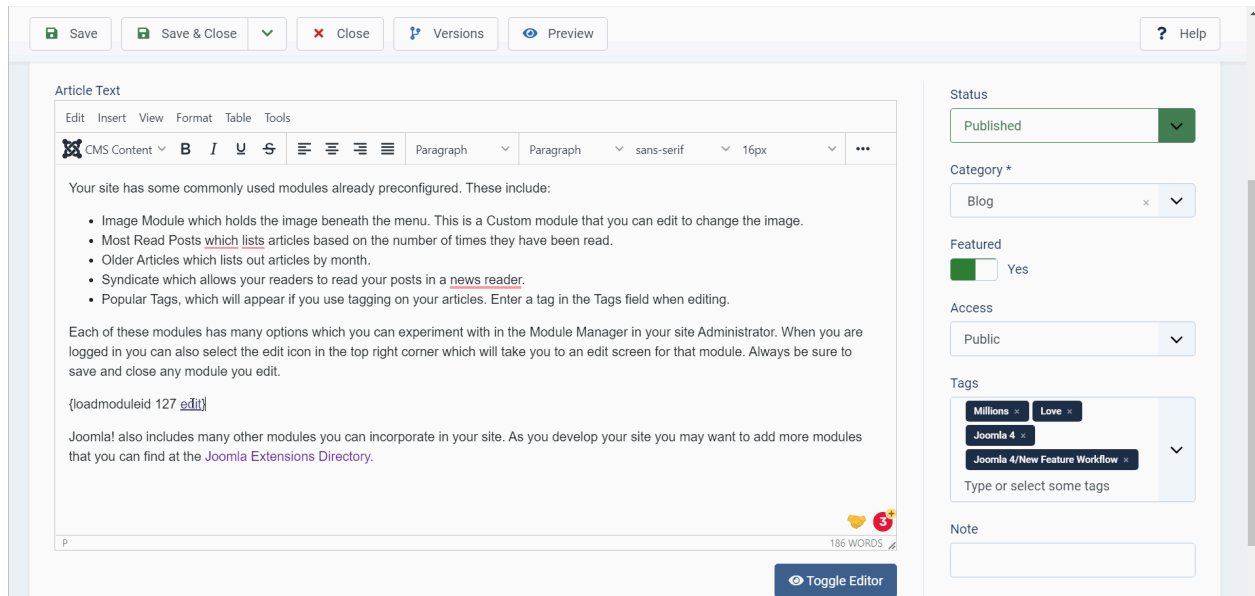


- A new tab for the Imported Modules.
 - This will also have a button to **Add New Module** (Opens Add Module Modal)
 - Each imported Module will be displayed as a table
 - The above GIF is not an actual representation of how it will look like. The table will be having only the following columns:
 - Module ID
 - Link to Edit Module
 - Module Name, etc details have been avoided here as they will require an extra SQL query. Maybe the extra query should be implemented here despite its cost.
- @Mentors** what's your opinion?

2. Inline Edit Module link for TinyMCE Editor: (Read Note below the GIF)

- Hyperlink inside of the editor for editing modules (Keeping textbox limitations in mind)
- This link would open the Edit Module view of the given module ID in a new tab.
- Users will have to right-click on the edit button and select the open link option in the dropdown for this.

- This link will not be stored in the backend anywhere but I will use regex and other string manipulation functions to add it conditionally for TinyMCE. That is, I could search for “{loadmodules /d+}” in regex and add the link to edit the module with ?id=:id before the closing “}”. This will be filtered out while saving the form.

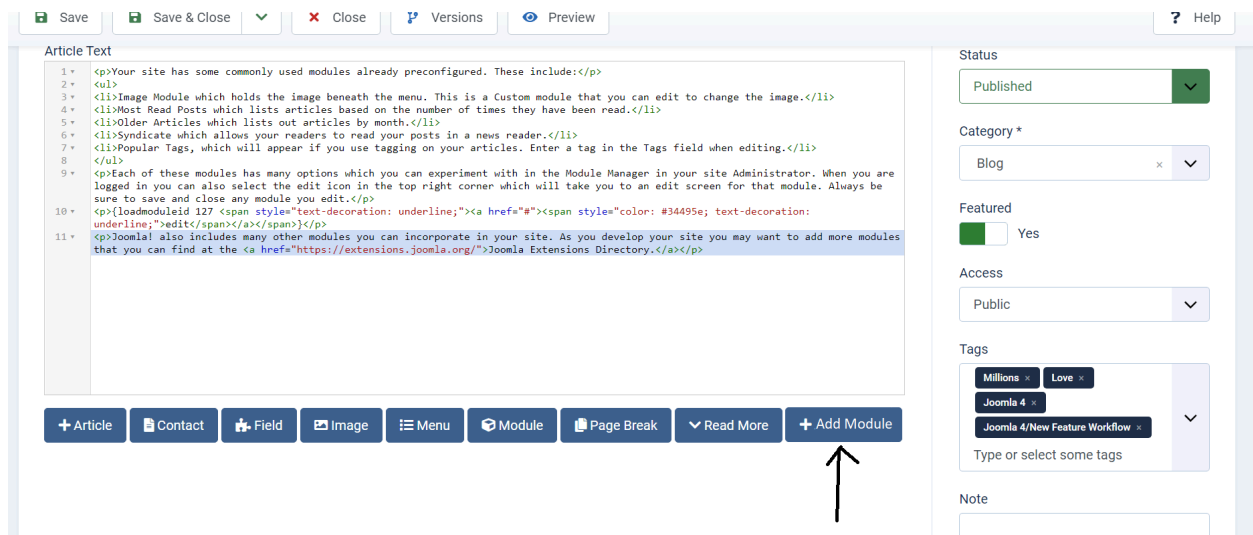


Note: This idea has a few flaws because of the limitations imposed by Editors and the biggest flaw is that it is only use-able for TinyMCE and other being that the link is indirectly visitable where the user will have to right-click on it first and then select the visit link option. Maybe this shouldn't be implemented (?)

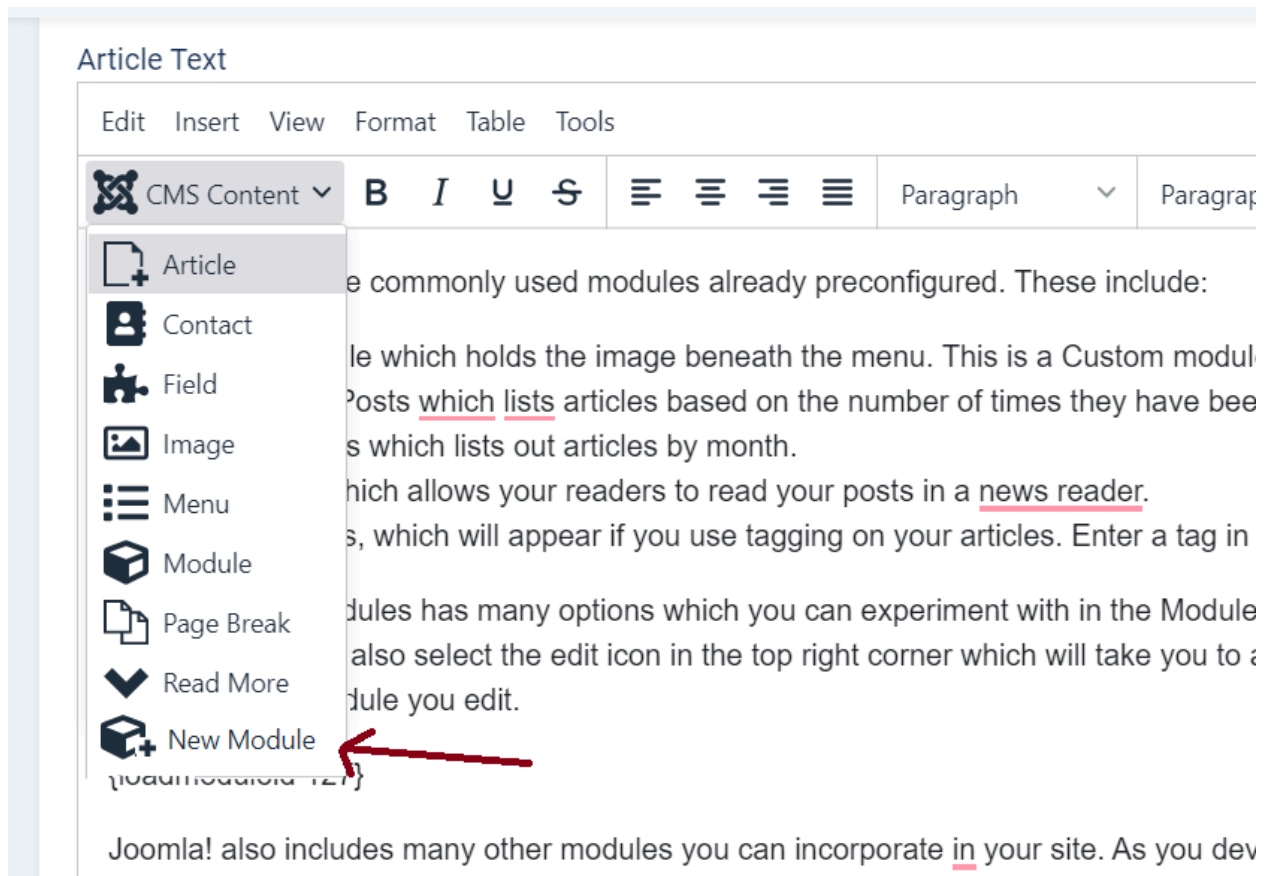
I believe there could be a better approach here but the Editor imposes many restrictions that cannot be overcome and a link like this seemed the only viable option.

3. Extra Button to Add Module in Module Selection

For CodeMirror



For TinyMCE



This button will be linked to the Add Module Page loaded via iframe in a Modal.

Proposed Timeline (2.5 weeks)

New Tab for Loaded Modules	June 28 to July 4	1 week
Add Module Button for TinyMCE	July 5	1 day
Add Module Button for CodeMirror	July 6	1 day

July 7 to July 9 (3 days) will be the backup days to cover up for holidays and unexpected bugs.

I haven't considered the Point **(2) Inline Edit button** for now as that idea seems incomplete to me. It could use some improvement. If I am able to improvise on it during Coding period then I will modify the timeline accordingly.

3. Improve Menu List view (Menu Items)

Aim of the Task

Add information of the selected menu item (like article title and id, category title and id) to the menu overview list and link it directly to the items. Categories receive a link to the category item as well as to the list of child items.

Proposed Approach

- New Column(s)

My initial plan was to take data from the same table by performing string slicing on the URL to find the Component Names but I figured out this was not the way to go as it doesn't consider the 3PD that might not have a well-formatted component/module pathname.

		menutype	title	alias	note	link	type	published	parent_id	level	
		The type of menu this item belongs to. FK to #_menu...	The display title of the menu item.	The SEF alias of the menu item.	The computed path of the menu item based on the al...	The actually link the menu item refers to.	The type of link. Component, URL, Alias, Separator	The published state of the menu link.	The parent menu item in the menu tree.	The relative level in the tree.	
☐		105	mainmenu	Separator	separator	separator	separator	0	1	1	
☐		106	mainmenu	Category List	category-list	category-list	index.php?option=com_content&view=category&id=2	1	1	1	
☐		107	main	COM_JEDCHECKER	com-jedchecker	com-jedchecker	index.php?option=com_jedchecker	1	1	1	
☐		108	search-page	Featured Article(s)	featured-article-s	featured-article-s	index.php?option=com_content&view=featured	1	1	1	
☐		109	search-page	GSoC Articles	gsoc-articles	gsoc-articles	index.php?option=com_content&view=category&id=8	1	1	1	

So, this approach stands void. There is a need to implement a **JOIN** query with the **component_id** and **#_extensions.id**. This might have some performance issues. To be honest, I spent a lot of time trying to figure this out but I was unsuccessful, The only other option was to add a new Column in the **#_menus** table but that would cause data redundancy which is not good practice. Plus the extra column would lead to more issues when it comes to updating and editing modules.

So, the **ONLY** possible solution I could think of is that we go ahead with a **JOIN** query and add the following new columns-

1. A **Hyperlinked Component/Module/Extension Name** of the Menu Item eg: Article. This ~~hyperlink will point to the administrator page for the component which will be the option=com_content&view=articles page for this case.~~

I'm not sure how I would determine the hyperlink here. I spent a lot of time thinking over it and browsing database tables but couldn't figure it out. I will update this later if some idea pops up in my mind.

~~This hyperlink will be generated by appending the #_menu's path column to JPATH_ADMINISTRATOR admin. This works because the frontend and backend links for such components are identical apart from the /administrator/ prefix~~ Edit: Doesn't work for every case, eg: Doesn't work for login, logout and edit profile, etc

	i	menutype	title	alias	note	path	link
		The type of menu this item belongs to. FK to #__me...	The display title of the menu item.	The SEF alias of the menu item.		The computed path of the menu item based on the al...	The actually link the menu
Group							
Smart-Search-Index		Smart Search/Smart-Search-Index				index.php?option=com_finder&view=index	component

Search

Q

Filter Options

Clear

Ordering ascending

20

☐		Status	Title	Menu	Home	Access	ID
☐			Featured Article(s) Alias: featured-article-s Articles > Featured Articles	Search Page		Public	108
☐			GSoC Articles Alias: gsoc-articles Articles > Category List > Category Name XYZ	Search Page		Public	109

Menu Item Type

Article Name / Category Name

On hovering over the Article Name / Category Name, More info on the same will be displayed in a popup

- The ID of the Item if it exists for example ID of Single Contact / Single Article / Category
- Name of the Item if it exists for example Category Name or Article Name etc

Search

Q

Filter Options

Clear

Ordering ascending

20

☐	▲	Status	Title	Item ID	Menu Item Details	Menu	Home	Access	ID
☐	⋮	✔	Featured Article(s) Alias: featured-article-s Articles	8	Featured Articles	Search Page		Public	108
☐	⋮	✔	GSoC Articles Alias: gsoc-articles Articles	9	Category Name XYZ Category List	Search Page		Public	109

- A link to its page view in Frontend

Menu Types where this can be used:

1. Article title and id
2. Category title and id
3. Contact name and id (for single contact)
4. Contact category and category id
5. Newsfeed title and id (for single news feed)
6. Newsfeed category and id
7. Tag title and id (for all items under single tag)
8. Menu item and id (For menu item alias)

Apart from the Name and ID for each of them, we could possibly add:

1. A link that redirects them to edit information about the menu type in the backend (for example a link to edit the single contact/category/article/news feed). Although I'm not sure how viable this would be and whether this would require an extra column in the DB
2. The name of the Component/Module that the Menu Type belongs to (This information is available in the same table as #_menu so it should be fairly easy to implement)
3. A link that navigates to the Frontend page of that menu item. This is also stored in the same #_menu table so it could be easily displayed in the table as a hyperlink without much performance impact.

Proposed Timeline (2 Weeks)

Column for Link to Frontend	July 10	1 day
Column for Name of Extension (News Feed/Contact/Article) using String Manipulation of the URL	July 11	1 day
Button to Modal / Show More Information (To Be Improvised)	July 12 to July 19	1 week

Since the last Idea is subject to improvisation, July 20 to July 24 (5 days) can be kept as backup for covering any delays.

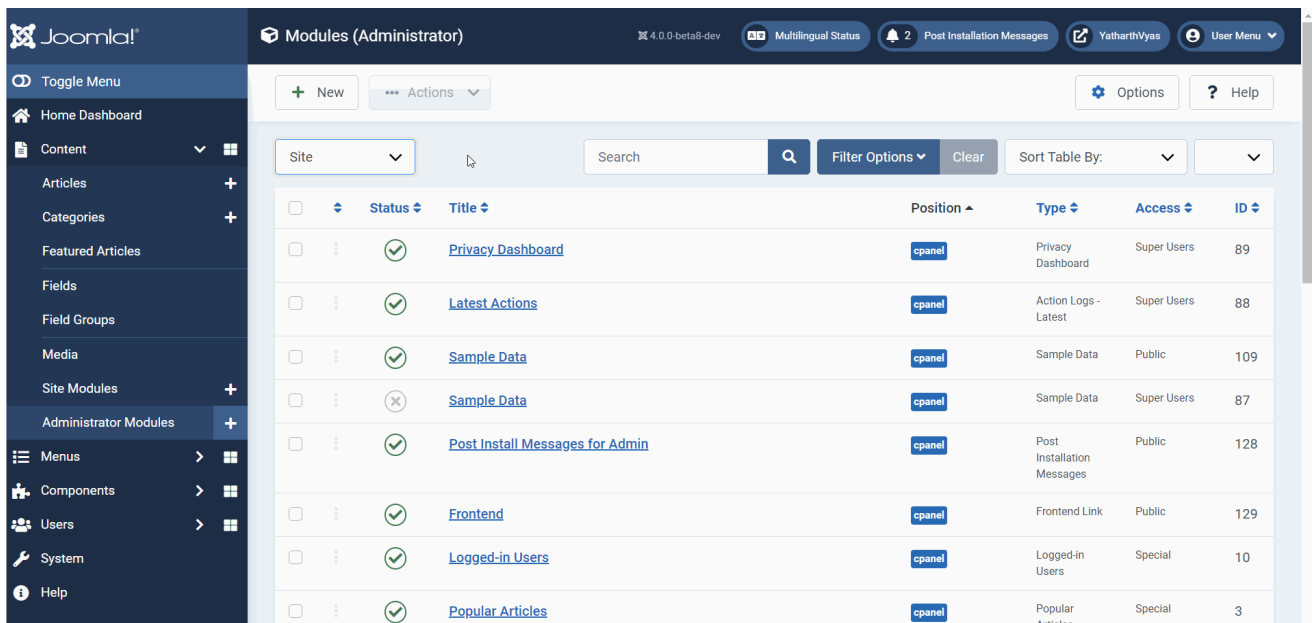
4. Improve Content: Merge Featured into Article

Proposed Approach

I propose implementing a dropdown outside the searchtools that defines the scope of the article's type. This will have 3 options

- Featured
- Unfeatured
- All Articles

To visualize, the output should look similar to the Add Modules Page. It has 2 unique menu items in the sidebar and they can be toggled using the dropdown. Both the selections (Administrator Modules and Site Modules) are displayed using a **single tmpl file** instead of two.



The screenshot shows the Joomla! administrator interface for the 'Modules' section. The left sidebar contains a 'Toggle Menu' button and a list of menu items: Home Dashboard, Content (with a dropdown arrow), Articles (with a plus icon), Categories (with a plus icon), Featured Articles, Fields, Field Groups, Media, Site Modules (with a plus icon), Administrator Modules (with a plus icon), Menus (with a right arrow), Components (with a right arrow), Users (with a right arrow), System, and Help. The main content area is titled 'Modules (Administrator)' and includes a toolbar with '+ New', 'Actions', 'Options', and 'Help'. Below the toolbar is a search bar and a 'Filter Options' dropdown. The table below has columns for 'Status', 'Title', 'Position', 'Type', 'Access', and 'ID'. The table contains 10 rows of modules, including 'Privacy Dashboard', 'Latest Actions', 'Sample Data', 'Post Install Messages for Admin', 'Frontend', 'Logged-in Users', and 'Popular Articles'.

	Status	Title	Position	Type	Access	ID
☐	✓	Privacy Dashboard	cpanel	Privacy Dashboard	Super Users	89
☐	✓	Latest Actions	cpanel	Action Logs - Latest	Super Users	88
☐	✓	Sample Data	cpanel	Sample Data	Public	109
☐	✗	Sample Data	cpanel	Sample Data	Super Users	87
☐	✓	Post Install Messages for Admin	cpanel	Post Installation Messages	Public	128
☐	✓	Frontend	cpanel	Frontend Link	Public	129
☐	✓	Logged-in Users	cpanel	Logged-in Users	Special	10
☐	✓	Popular Articles	cpanel	Popular Articles	Special	3

There is also a database table that only contains featured articles **#_ucm_content** so it is necessary to ensure that this DB table is used to display the featured articles if the dropdown is set to Featured for faster query performance.

For the toolbar, the options for the 'Actions' dropdown will have to be conditionally selected as

Type	Toolbar Actions
All	Publish, Unpublish, Feature, Unfeature, Archive, Check-in, Trash, Batch
Featured	Publish, Unpublish, Unfeature, Archive, Check-in, Trash, Batch
Unfeatured	Publish, Unpublish, Feature, Archive, Check-in, Trash, Batch

The following files for featured will have to be deleted and merged into the article:

Merge From	Merge Into
\administrator\components\com_content\tmpl\featured\	\administrator\components\com_content\tmpl\articles\
\administrator\components\com_content\src\Controller\FeaturedController.php	\administrator\components\com_content\src\Controller\ArticlesController.php
\administrator\components\com_content\src\Model\FeaturedModel.php	\administrator\components\com_content\src\Model\ArticlesModel.php
\administrator\components\com_content\src\Model\FeatureModel.php	\administrator\components\com_content\src\Model\ArticleModel.php
\administrator\components\com_content\src\View\Featured\HTMLView.php	\administrator\components\com_content\src\View\Articles\HTMLView.php
\administrator\components\com_content\src\Table\FeaturedTable.php	\administrator\components\com_content\src\Table\ArticleTable.php

Proposed Timeline (3 weeks)

Drop-down in frontend	July 25	1 day
Merge tmpl file	July 26 to July 27	2 days
Merge Model	July 28 to July 30	3 days
Merge View	August 1 to August 2	2 days
Merge Controller	August 3 to August 5	3 days
Merge Table	August 6 to August 8	3 days
Fix any links pointing to Featured	August 9	1 day

August 10 to August 15 can be kept as a backup to cover any backlogs.