
Joomla

Project III: Feature enhancement

April 2021



Google Summer of Code

REPRESENTATIVE

Mahmoud Ahmed Tawfik

Faculty of Computers and Artificial Intelligence, Cairo University

About me	1
Contact Information	1
My Background	1
Project Background & Design/Description	4
Cookies Management	4
Improve Menu List view (Menu Items) & Menu-Module Assignment	11
Improve Content	12
Improve The Incorporation Of Modules In The Articles-Edit-View	14
Benefit of my work to the Joomla Project users & developers	15
Users	15
Developers	15
Deliverables & Scheduling	16
Pre Coding Phase	16
Phase 1	16
Phase 1 Evaluation	17
Phase 2	17
Phase 2 Evaluation	17
Final Phase	17
Post GSoC	17
Other commitments	18
Community engagement	18
Appendix A: Programming Task	20

About me

Contact Information:

Name: Mahmoud Ahmed Tawfik
Country: Egypt
Email: mahmoudaahmedd@gmail.com
Skype Name: live:.cid.981c0592300f5219
University: Cairo University, Faculty of Computers and Artificial Intelligence
Contact Number: +201024970738
LinkedIn: [linkedin.com/in/mahmoudaahmedd](https://www.linkedin.com/in/mahmoudaahmedd)
Github: github.com/mahmoudahmedd
Website: mahmoud-ahmed.me

My Background:

I am a student in Computer Science/Information System departments, Faculty of Computers and Artificial Intelligence, Cairo University, Egypt.

I'm quite interested in web development. I am interested heavily in leveraging web technology and creating impactful products to help people to solve their problems. I've heard about Joomla from Joomla! Forum, and I really liked the community. I have made the [programming task](#), and I think this project is a really good opportunity for me to contribute to this community!

Previously I worked as a Software Engineer on payment gateway for a startup in Saudi Arabia. I developed many resources and improved the service response time for payments endpoint.

Previously I worked at Orange as Software Engineer Intern to build an internal web app to be used by the IT team in Egypt to manage 3000+ antennas including locations and electricity bill transactions by using PHP, HTML5/CSS3, Bootstrap, JS, Geolocation API, PHP-ExcelReader, MySQL, Git.

I also love Photography, Photography records history and preserves it for future generations to see. You can see some of the photos that i've taken [here](#).

Project Background & Design/Description

My project will be divided into **four** tasks

1. Cookies management system

The goal of the task

Webmasters need to integrate a cookie management system plugin to make their Joomla site compliant with many options that the Webmasters can control, customize the plugin from the administration Form. So the Webmaster can show his/her compliance status regarding the GDPR, CCPA and other privacy regulations law.

The problem

- Many options to make the manager able to control, customize the plugin from the administration Form for example, customize to put the cookie banner in either the header or the footer, handle cookie expiration period, configure cookie details and their description, handle different cookie categories.
- There are different needs for every website as there is a difference between GDPR, CCPA and other privacy regulations. As each region has its own consent. For example under the GDPR, The GDPR protects any individual located inside the EU, whereas the CCPA protects California residents.
- Auto block external services like Google Analytics. because It offers an EU Compliance addon, which it will be easy for them to meet GDPR requirements in just a few clicks.
- Some websites do not have a Cookie Policy page. Thus they need to generate their own Cookie Policy page.
- Cookie banner preview in administration Form for ease of viewing and usage.

Description & Design

I have developed a [draft task](#). It includes important folders such as language, media and sql. also I will develop a host of features some of which are listed below to solve the problems that have been mentioned.

The administration form (UI):

- General tab
- Various options to select the type of law needed for your site (GDPR, CCPA, GDPR & CCPA).
- Auto-hide option where you can accept an implicit consent from the user on delay or on scroll.

The screenshot shows a web-based administration interface for a plugin. At the top, there is a navigation bar with 'Plugin' and 'Customize cookie'. Below this, the main content area is titled 'plg_system_draft_task' and includes a breadcrumb trail 'system / drafttask'. The description reads 'Programing task plugin.' The form contains three configuration sections: 1. 'select the type of law needed for your site *' with radio buttons for 'GDPR', 'CCPA', and 'GDPR and CCPA'. 2. 'Auto hide on *' with radio buttons for 'delay' and 'scroll'. 3. 'Cookie lifetime *' with a text input field containing the value '60'.

- Customize cookie

All the customization options are embedded within the customize cookie tab

- Add a message header for the notice in the field provided.
- Add a message in the field provided.
- Cookie bar colour.
- Text colour.
- Font.
- Option to cookie notice is shown in the banner format, into pop-up.
- Position (the notice at the header or at the footer).
- Add link for Cookie Policy page.

The image shows a user interface for customizing a cookie notice. At the top, there are two tabs: 'Plugin' and 'Customize cookie', with the latter being selected. Below the tabs, the main content area is titled 'Customize cookie'. It contains several configuration options:

- Message Header:** A text input field with the placeholder text 'Enter a message header.'
- Message:** A text input field with the placeholder text 'Enter a message.'
- Cookie bar colour:** A color picker showing a light green color with the hex code 'rgba(238, 238, 238, 1)'. Below it is the label 'Cookie bar colour.'
- Text Colour:** A color picker showing a light green color with the hex code 'rgba(238, 238, 238, 1)'. Below it is the label 'Text Colour.'
- Cookie notice is shown in *:** Two radio button options: 'Banner' and 'Pop-up'.
- Position *:** Two radio button options: 'Header' and 'Footer'.

- Customize Buttons

For accept, Reject, Settings buttons

- Text colour.
- Font.

- Geolocation API

Integrate third party API for geolocation

- link address.
- location field name in the third party API.

- Generate Cookie Policy page

- Various options to select, then automatically generate the page.

- Cookie banner preview

- Display the cookie banner.

The administration form (Back-End):

- onAjaxDraftTask

This is one of the functions in the back end, When the user clicks on the OK button, this function will be executed, which will insert a transaction with some important information like (User-Agent, URL where the user consents) in the database. To facilitate this process we will use ajax. As Ajax allows users to make asynchronous calls to a web server. to avoid waiting for all data to arrive before allowing the user to act once more.

```
public function onAjaxDraftTask()
{
    // Get the user agent string
    $userAgent = $this->app->input->server->get('HTTP_USER_AGENT', '', 'string');
    // Get the request url string
    $requestUrl = $this->app->input->server->get('REQUEST_URI', '', 'string');
    // Generate a random UUID-4
    $uuid = $this->genUuid();
    // Generate a random hash to identify the action
    $ccuuid = md5($uuid . time());
    // Create the record
    $data = new stdClass();
    $data->uuid = $uuid;
    $data->user_agent = $userAgent;
    $data->consent_url = $requestUrl;
    $data->action_taken = "all_cookies";
    $data->message = $message;
    $data->ccuuid = $ccuuid;
    $data->created_at = date('Y-m-d H:i:s');

    try
    {
        $this->db->insertObject('cookie_consent', $data);
        // Set cookie with value
        $this->app->input->cookie->set(
            "uuid",
            $data->uuid,
            time() + $lifetime,
            $this->app->get('cookie_path', '/'),
            $this->app->get('cookie_domain', ''),
            $this->app->isHttpsForced(), true
        );
    }
    catch (Exception $e)
    {
        // Do nothing if the save fails
    }

    $data = array("ok");
    return $data;
}
```

Activ
Go to S

- genUuid

to Generate a universally unique identifier (UUID) used to identify visitors.

```
public function genUuid()
{
    return sprintf('%04x%04x-%04x-%04x-%04x-%04x%04x%04x',
        mt_rand(0, 0xffff), mt_rand(0, 0xffff),
        mt_rand(0, 0xffff),
        mt_rand(0, 0x0fff) | 0x4000,
        mt_rand(0, 0x3fff) | 0x8000,
        mt_rand(0, 0xffff), mt_rand(0, 0xffff), mt_rand(0, 0xffff)
    );
}
```

- onBeforeRender

When visiting the website, we must know if the visitor has previously given consent or not. Thus we need to set a cookie that will give us the consent date.

```
public function onBeforeRender()
{
    // Get the parameter values
    $lifetime = $this->params->get('cookie_lifetime', 60) * 24 * 60 * 60;
    $cookieName = $this->params->get("cookie_name", "banner_closed");
    $message = $this->params->get("message", "message");

    // Get cookie
    $cookieValue = $this->app->input->cookie->get($cookieName);

    // Check if a cookie is set.
    if(!$cookieValue)
    {
        // Set cookie with value
        $this->app->input->cookie->set(
            $cookieName,
            microtime(true),
            time() + $lifetime,
            $this->app->get('cookie_path', '/'),
            $this->app->get('cookie_domain', ''),
            $this->app->isHttpsForced(),
            true
        );
    }
}
```

The database table:

The database table has the following info.

```
--  
-- Table structure for table `cookie_consent`  
--  
CREATE TABLE `cookie_consent` (  
  `id` bigint(20) UNSIGNED NOT NULL,  
  `uuid` varchar(36) NOT NULL,  
  `user_agent` varchar(512) NOT NULL,  
  `consent_url` varchar(512) NOT NULL,  
  `action_taken` varchar(64) NOT NULL,  
  `message` text NOT NULL,  
  `ccuid` varchar(32) NOT NULL,  
  `created_at` datetime NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

I used Index with **uuid** column to retrieve data from the database more quickly than otherwise.

```
--  
-- Indexes for table `cookie_consent`  
--  
ALTER TABLE `cookie_consent`  
  ADD PRIMARY KEY (`id`),  
  ADD KEY `uuid` (`uuid`);  
--
```

2. Improve Menu List view (Menu Items) & Menu-Module Assignment

My goal for this task is to add information of the selected menu item and place the module menu assignment in the menu item. on another hand improvement user interface for Menu List view and Menu-Module Assignment.

The problem is that an overview is not presented to the user well, For example. user finds it difficult to assign modules in the menu item. As it is difficult to find the button easily.

Design and Description of my work

The solution to the problem is mainly in modifying the user interface (UI), Because we need to move the Module Assignment button to the menu item by adding an additional column to the menus view. as well as add information of the selected menu item (like article title and id, category title and id)

I will research to understand more about the Joomla component, It would be nice if my mentor be well acquainted with Joomla components development. But that will not prevent me from learning with myself from Joomla! Documentation

https://docs.joomla.org/J4.x:Developing_an_MVC_Component

3. Improve Content

The goal of the task

My goal in this task is to improve content to make it not confused and not duplicated menu items. Which will facilitate viewing the articles for the Webmasters.

The problem

The problem will be in two subtasks

- Displaying featured articles as the first articles in view.
- Remove the view featured.

Design and Description of my work

For displaying featured articles as the first articles in view articles, There are several ways to display featured articles as first articles.

Among those ways in `com_content/src/Model/ArticlesModel.php` file, By the way it is responsible for supporting a list of article records. this file contains function `populateState($ordering = 'a.featured', $direction = 'DESC')`, Which has two parameters (`$ordering`, `$direction`).

By changing the value of the first parameter to be 'a.featured' instead of 'a.id' and keep the second parameter as it is 'desc'. because we need to display featured articles in descending order.

As it is shown in the picture:

```
/**
 * Method to auto-populate the model state.
 *
 * Note. Calling getState in this method will result in recursion.
 *
 * @param string $ordering An optional ordering field.
 * @param string $direction An optional direction (asc|desc).
 *
 * @return void
 *
 * @since 1.6
 */
protected function populateState($ordering = 'a.featured', $direction = 'desc')
{
```

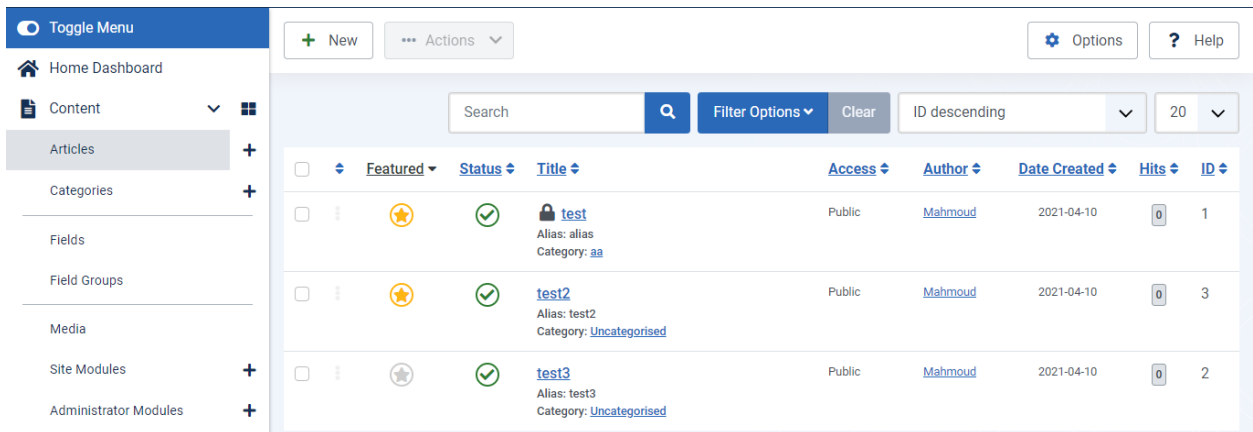
to remove the view featured, we are going to update the content component... first in presets **content.xml** We will delete this part to hide it from Toggle Menu.

```
<menuitem
  title="COM_CONTENT_MENUS_CATEGORY_MANAGER"
  type="component"
  element="com_categories"
  link="index.php?option=com_categories&view=categories&extension=com_content"
  quicktask="index.php?option=com_categories&extension=com_content&task=category.add"
  quicktask-title="COM_CONTENT_MENUS_NEW_CATEGORY"
/>
```

Then we delete **FeatureModel** in **com_content/src/Model** folder, **Featured** folder in **com_content/src/View**, featured folder in **com_content/tmp**

Screenshot

This is a cropped screenshot of the outcome for displaying featured articles written with Joomla 4.0.



4. Improve the incorporation of modules in the Articles-Edit-View

My goal is to improve the incorporation of modules in the Articles-Edit-View.

Design and Description of my work

By enhancing the user interface (UI) the function to insert modules in an article will be supplemented with the function to create and edit.

I will research to understand more about the Articles Joomla component, It would be nice if my mentor be well acquainted with Articles Joomla components development.



Benefit of my work to the Joomla Project users & developers

For Users:

- All of the above tasks are suitable candidates after Joomla supports the cookie management system plugin and enhanced control, performance, etc. for users.
- Cookie management system plugin feature can be installed individually as a separate plugin without the need to make changes to the Joomla administrator page code.

For Developers:

- I will be a new active contributor to the community and I will make an effort to accomplish the tasks required of me.
- I will keep my code clean and have comments to help those who want to contribute to develop those tasks after me.
- After the GSoC program is over, I would like to follow up more projects. I'll be happy to contribute to it. I would also like to propose new features that can help the growth of the Joomla CMS. As I believe that the three-month period with the Joomla community will increase my knowledge and make me able to do so.

Deliverables & Scheduling

The Request for Proposal timeline is as follows:

- Pre Coding Phase (April 13 - May 17)

April 13 to May 17

Although this is not a part of GSoC program, I will exploit this period to self study and get ready in case I get accepted to start coding without any latency. I will be giving minimum of 30 hours a week for this project during this period to accomplish these tasks:

- Familiarize myself more with how Joomla works.
- Get more familiar with the Joomla plugins/components development to create it.
- Trying to close the rest of the opened issues.

May 17 to June 7 (community bonding period)

- Discuss the strategies and timeline with the mentors for any suggested modification.
- Remain active regularly to efficiently discuss those strategies.
- Finalize the implementation strategy of tasks' implementation in addition to any modifications suggested by the mentors.

- Phase 1 (June 7 - July 12)

- Implement cookie management system plugin.
- Add unit/integration tests for the cookie management system plugin.

From June 7 to June 10

- Create the administration form UI (Manifest file)
- Create banner UI (HTML5/CSS3, JS)

From June 10 to June 12

- Create pop-up UI (HTML5/CSS3, JS)

From June 12 to June 30

- Back-End development for the plugin

From June 30 to July 5

- Support multiple language (.ini & sys.ini files)

From July 5 to July 12

- Add unit test for the plugin
- Add integration test for the plugin

- Phase 1 Evaluation (July 12 – July 16)

- Discuss the drawbacks of Phase 1 with the mentor and ask for any improvements.
- Ask the mentor for any additional info required for Phase 2.
- Gain any missing information related to Phase 2

- Phase 2 (July 16 – July 31)

- Improve Menu List view (Menu Items) with tests.
- Improve Menu-Module Assignment with tests.
- Improve Content with tests.
- Improve the incorporation of modules in the Articles-Edit-View with tests.

- Phase 2 Evaluation (July 31 – Aug 4)

- Discuss the drawbacks of Phase 2 with the mentor and ask for any improvements.
- Ask the mentor for any additional info required for the final Phase.
- Gain any missing information related to the final Phase.

- Final Phase (Aug 4 – Aug 16)

- Complete the implementation and work on the documentation with code examples for cookie management system plugin.
- Submit final code and project summaries.

- Post GSoC

I would like to follow up on this project even after the GSoC program is over. I'll be happy to contribute to the Joomla community after the GSoC.

Other commitments

I will be involved with my midterm exams from May 15 to May 21. Also with my final exams from July 1 to July 15. So, I will interact less. However, I can give 10 hrs per week to my project/community during this time period. I assigned small tasks in this period.

Community Engagement

A plugin to Joomla users

Brief

I have developed a plugin to Joomla 3 users to help them to rank higher in search engines. SEOPlugin is a simple plugin, designed to help visitors and search engines to get the most out of your website.

Some of which are listed below

- . Automated technical SEO to generate meta tags, like description.
- . Meta description, for better branding and consistent snippets in the search results.
- . Set different title schema.

The plugin code

Download the plugin to your local machine as a zip file package.

The plugin code and working zip file are available on [GitHub](#).

User Interfaces

The Plugin Administration Form

Plugin

Title Tag Settings

SEO Plugin

system / SEOPugin

This Joomla plugin helps you to stay ahead.

Meta Description Length:

Front Page Meta Desc:

- Use Config Setting
 Generate From Content
-

Category Meta Desc:

- Use Category Description
 Generate From Content
-

Plugin

Title Tag Settings

Page Title Layout:



Front Page Title Layout:



Custom Title:

Separator:



Appendix A: Programming Task

Background

ProgrammingTask plugin generates a nice popup with some informative Text. The plugin code and working zip file are available on GitHub:

<https://github.com/mahmoudahmedd/programming-task-plg>

The zip file structure

The installation zip file is structured like this:

```
programmingtask.zip
  language
    en-GB
      en-GB.plg_system_programmingtask.ini
      en-GB.plg_system_programmingtask.sys.ini
  media
    css
      main.css
    js
      app.js
  programmingtask.php
  programmingtask.xml
```

The manifest file

[The manifest file](#) contains extension data and it tells the install code what goes where. the plugin files will be installed in `site_root/plugins/system/programmingtask`.

The languages destination folder is `site_root/administrator/language` and the media destination folder is `site_root/media/plg_system_programming_task`.

The language files

The sys.ini file is used by the system during installation and when listing plugins.

The en-GB.plg_system_programmingtask.sys.ini file:

```
1  #@copyright 2021 Mahmoud Ahmed
2  #@author Mahmoud Ahmed <mahmoud_Ahmed@stud.fci-cu.edu.eg>
3
4  PLG_PROGRAMING_TASK_DESCRIPTION = "Programing task plugin."
```

The .ini file is used when displaying the plugin parameters form.

The en-GB.plg_system_programmingtask.ini file:

```
1  #@copyright 2021 Mahmoud Ahmed
2  #@author Mahmoud Ahmed <mahmoud_Ahmed@stud.fci-cu.edu.eg>
3
4  PLG_SYSTEM_PROGRAMING_TASK_COOKIE_NAME_LABEL="Your Cookie-Name:"
5  PLG_SYSTEM_PROGRAMING_TASK_COOKIE_LIFETIME_LABEL="Cookie Lifetime (days):"
6
7  PLG_SYSTEM_PROGRAMING_TASK_ERROR_NO_COOKIE_NAME="No Cookie Name."
8
9  PLG_SYSTEM_PROGRAMING_TASK_COOKIE_BANNER_HEADER="Hello"
10 PLG_SYSTEM_PROGRAMING_TASK_COOKIE_BANNER_DESCRIPTION="The first time you visited us:"
11
```

The media files

The main.css file is a cascading style sheet (CSS) file used to format the contents of a webpage.

The app.js file is a document that contains myFunction() of JavaScript code to hide the form when pressing the "OK" button.

The app.js file:

```
1  function myFunction() {
2      document.getElementById("myDIV").style.display = "none";
3  }
```

The Plugin Administration Form

This form is available via the Administrator interface.

The screenshot shows a web interface for plugin administration. At the top, a dark blue header contains a logo, a plug icon, the text "Plugins: plg_system_programming...", and a version indicator "4.0.0-beta8-dev". Below the header are three buttons: "Save", "Save & Close", and "Close". The main content area is titled "Plugin" and displays the details for "plg_system_programming_task". It includes a breadcrumb "system / programmingtask" and a description "Programing task plugin.". Two form fields are visible: "Your Cookie-Name: *" with the value "is_visited_before" and "Cookie Lifetime (days): *" with the value "60".

Plugins: plg_system_programming... 4.0.0-beta8-dev

Save Save & Close Close

Plugin

plg_system_programming_task

system / programmingtask

Programing task plugin.

Your Cookie-Name: * is_visited_before

Cookie Lifetime (days): * 60

The Plugin Code

See the code on [GitHub](#) for the full list.

The entry point from programmingtask.php:

```
<?php
/**
 * @author  Mahmoud Ahmed
 * @version 1.0.0
 */
defined('_JEXEC') or die;

use Joomla\CMS\Plugin\CMSPlugin;
use Joomla\CMS\Language\Text;

/**
 * ProgrammingTask Plugin.
 */
class PlgSystemProgrammingTask extends CMSPlugin
{
    /**
     * Load the language file on instantiation.
     *
     * @var    boolean
     * @since 3.9.0
     */
    protected $autoloadLanguage = true;

    /**
     * Application object.
     *
     * @var    \Joomla\CMS\Application\CMSApplication
     * @since 3.8.0
     */
    protected $app;
```

Screenshot

This is a cropped screenshot of the outcome for a programming task written with Joomla 4.0.

