Let Us Build a WordPress Custom Post Type (CPT)

Sudip Chakraborty [1] & P. S. Aithal [2]
[1] D.Sc. Researcher, Institute of Computer Science and Information Sciences, Srinivas University, Mangalore-575 001, India,
OrcidID: 0000-0002-1088-663X; E-mail: sudip.pdf@srinivasuniversity.edu.in
[2] Professor, Institute of Management & Commerce, Srinivas University, Mangalore, India,
OrcidID: 0000-0002-4691-8736; E-Mail: psaithal@gmail.com

Subject Area: Computer Science.
Type of the Paper: Experimental Research.
Type of Review: Peer Reviewed as per |C|O|P|E| guidance.
Indexed In: OpenAIRE.
DOI: https://doi.org/10.5281/zenodo.10440842
Google Scholar Citation: IJAEML

How to Cite this Paper:
Chakraborty, S. & Aithal, P. S. (2023). Let Us Build a WordPress Custom Post Type (CPT). *International Journal of Applied Engineering and Management Letters (IJAEML)*, *7*(4), 259-266. DOI: https://doi.org/10.5281/zenodo.10440842

International Journal of Applied Engineering and Management Letters (IJAEML)
A Refereed International Journal of Srinivas University, India.

Crossref DOI: https://doi.org/10.47992/IJAEML.2581.7000.0202

Received on: 03/12/2023
Published on: 29/12/2023

Let Us Build a WordPress Custom Post Type (CPT)

Sudip Chakraborty [1] & P. S. Aithal [2]
[1] D.Sc. Researcher, Institute of Computer Science and Information Sciences, Srinivas University, Mangalore-575 001, India,
OrcidID: 0000-0002-1088-663X; E-mail: sudip.pdf@srinivasuniversity.edu.in
[2] Professor, Institute of Management & Commerce, Srinivas University, Mangalore, India,
OrcidID: 0000-0002-4691-8736; E-Mail: psaithal@gmail.com

ABSTRACT

Purpose: *WordPress is a popular content management system. Almost half of the internet websites are built using WordPress. It stores data as a post inside its database. It has a built-in post type where a programmer or developer can quickly build a post. But sometimes it does not fulfill our requirements. We need to customize the available posts by adding additional fields. Another way is to create a custom post type from scratch. Sometimes, we are not encouraged to implement custom post types due to the lack of practical or lengthy documentation. Here, we provide practical-oriented documentation so researchers can quickly build custom post types into their projects. The CPT module code is freely available to download and use.*
Design/Methodology/Approach: *We install WordPress websites locally using a "local" application. Inside the plugin folder, we create a folder named "custom-post-type." Inside the folder, we make a file "index.php." We add our code inside the file so that when the website refreshes, it establishes a plugin automatically.*
Findings/Result: *WordPress's custom post type is an outstanding feature. Where the default post type cannot provide proper data structure, the custom post type allows us to fulfill our requirements. We installed a WordPress website using local software and created a custom post. After that, we add data to it. It worked without any issues.*
Originality/Value: *Many documents are available on WordPress's custom post type. Most of those are lengthy and time-consuming to execute. Sometimes, our researcher cannot afford more time to implement the code into their project. So, consuming several hours, we provide the complete practical process in a very concise manner.*
Paper Type: *Experimental-based Research.*
Keywords: WordPress Custom Post Type, WordPress Posts, WordPress CPT, WordPress Additional field in Post type.

1. INTRODUCTION :

Nowadays, most industrial or home appliances have several sensors. Almost all gadgets have sensors. Nowadays, fridges and ACs are IoT-enabled. The company can detect the defective component without visiting the customer's house. They are well equipped when they will service the faulty gadgets, cutting down the service time. Users also benefit from this technology. They can know the running status of the AC. If they forget to turn off the AC, they can quickly turn it off from the office or anywhere outside the home. The administration can view the production process from their office using the digital twin technology. In all the above examples, one thing is common: the data flow between two nodes. The communication may be direct or indirect. Direct communication is synchronized in nature. The indirect communication is Asynchronized in nature. This communication has an intermediate buffer or temporary data storage place where one end updates the data with a specific interval. The other node fetches the data from the temporary buffer. In this scenario, the IoT market is growing fast. The big IT company created its IoT infrastructure. We can build our custom IoT infrastructure using their platform within a couple of hours. But the service is not free. We need to subscribe to at least one plan. With the budget constraints in their project, the researcher can face a little challenge.

In this scenario, we planned a way to keep our data in the cloud storage. We host a website. Every website uses a database to store the content and configuration. We have already spent enormous amounts of money hiring cloud space to run the website. We can hold our research data in the cloud space. In WordPress, we keep data as a post type. But post type sometimes does not meet our requirement, or data transaction is not optimized. For that, we need a custom post type or CPT. Here, we provide practical examples of how we can create CPT easily. The new researcher can benefit from this work.

2. RELATED WORKS :

Jones, K. M., et. Their book discussed the content management system's best practices [1]. Fragulis, G. F. et. Al. designed an online dynamic examination system using a CMS WordPress plugin [2]. In their other book, Jones, K. M. et al. discuss WordPress as a library content management system [3]. In their paper, Leary et al. discuss Custom Post Types, Taxonomies, and Fields [4]. Lakshmi, S. M. et. Al. built an online dynamic outpatient queue system for automated hospital token generation [5]. Jones, K. M. et al. show how to start WordPress [6]. Dimenstein, I. B. et. Al. developed a laboratory niche website [7]. Duong, B. Q. et al. worked on implementation guides to support the clinical adoption of pharmacogenomics [8]. In their book, Weller B. et al. illustrated how to Create versatile and powerful marketing and advertising campaigns [9]. In their paper, Chakraborty S. et al. demonstrate CRUD operation on WordPress [10][11]. Another paper described implementing Git technology in any project [12]. Chakraborty S. et al. told how to implement an automation framework for data transfer and GUI [13][14][15].

3. OBJECTIVES :

We implement a custom post type if WordPress's default post type is unsuitable. The CPT or custom post type is a little bit tricky to implement. The objective of the research is to provide some practical information regarding CPT. The researchers can easily integrate the CPT into their project following a step-by-step procedure. We offered the used code so that they could rapidly incorporate it.

4. APPROACH AND METHODOLOGY :

Figure 1 depicts the project block diagram. At first, we installed WordPress locally using the Local application. Inside the plugin folder, we create a custom plugin. Inside the folder, we created a file.

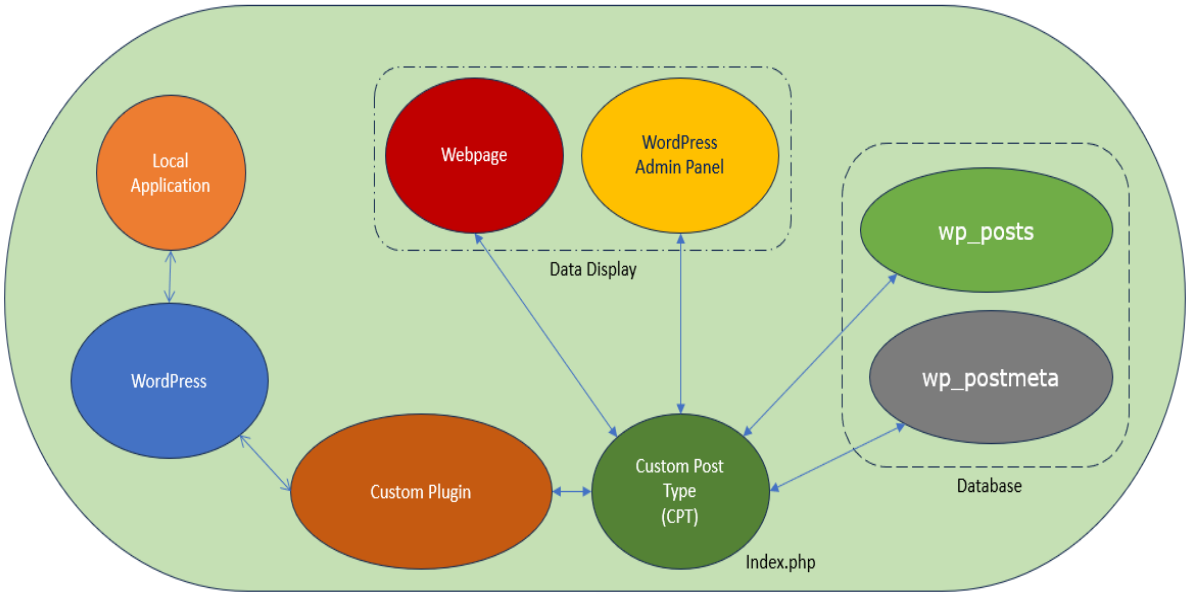

Fig. 1: Project Block Diagram

Name “index.php”. Inside the file, we add a custom post-handler code. This module is connected to two database tables: “**wp_posts**” and “**wp_postmeta**” The primary data is stored at “**wp_posts**”, and the additional fields are stored inside the “**wp_postmeta**”.

5. EXPERIMENT:

5.1 Create Custom Post Type from scratch:

1) Create a project in GitHub and clone it into our system. The paper [12] can be a good reference to create GitHub repository.
2) Install "**Local**" software. Inside the application, install the WordPress website. The paper [16][17] might guide us in establishing the local website.
3) Under "**..\app\public\wp-content\plugins**" , create_folder "**custom-post-type**".
4) Inside, create the file "**index.php**." Write the code below, which is depicted in figure 2.

index.php ● Settings

plugins > custom-post-type > index.php

```php
<?php
/* Plugin Name: Custom Post Type Plugin
* Description: Create custom post type
* Version: 1.0.0
* Author: Dr. Sudip Chakraborty
*/
function register_custom_post_type() {
    $labels = array
    (
        'name'                  => _x( 'Events', 'Post type general name'),
        'singular_name'         => _x( 'Event', 'Post type singular name'),
        'menu_name'             => _x( 'Events', 'Admin Menu text'),
        'name_admin_bar'        => _x( 'Event', 'Add New on Toolbar'),
        'add_new'               => __( 'Add New'),
        'add_new_item'          => __( 'Add New Event'),
        'new_item'              => __( 'New Event'),
        'edit_item'             => __( 'Edit Event'),
        'view_item'             => __( 'View Event'),
        'all_items'             => __( 'All Events'),
        'search_items'          => __( 'Search Events'),
        'parent_item_colon'     => __( 'Parent Events:'),
        'not_found'             => __( 'No Events found.'),
        'not_found_in_trash'    => __( 'No Events found in Trash.'),
        'featured_image'        => _x( 'Events Cover Image', 'Overrides the “Featured Image” phrase for th
        'set_featured_image'    => _x( 'Set cover image', 'Overrides the “Set featured image” phrase for t
        'remove_featured_image' => _x( 'Remove cover image', 'Overrides the “Remove featured image” phrase
        'use_featured_image'    => _x( 'Use as cover image', 'Overrides the “Use as featured image” phrase
        'archives'              => _x( 'Events archives', 'The post type archive label used in nav menus.
        'insert_into_item'      => _x( 'Insert into Events', 'Overrides the “Insert into post”/”Insert int
        'uploaded_to_this_item' => _x( 'Uploaded to this Events', 'Overrides the “Uploaded to this post”/”
        'filter_items_list'     => _x( 'Filter Events list', 'Screen reader text for the filter links head
        'items_list_navigation' => _x( 'Events list navigation', 'Screen reader text for the pagination he
        'items_list'            => _x( 'Events list', 'Screen reader text for the items list heading on th
    );

    $args = array(
        'labels'             => $labels,
        'public'             => true,
        'publicly_queryable' => true,
        'show_ui'            => true,
        'show_in_menu'       => true,
        'query_var'          => true,
        'rewrite'            => array( 'slug' => 'event' ),
        'capability_type'    => 'post',
        'has_archive'        => true,
        'hierarchical'       => false,
        'menu_position'      => null,
        'menu_icon'          => 'dashicons-calendar',
        'supports'           => array( 'title', 'editor', 'author', 'thumbnail', 'excerpt', 'comments'),
    );

    register_post_type( 'event', $args );
}
add_action( 'init', 'register_custom_post_type' );
////////////////////////////////////////////////////////////////////////////
```

Fig. 2: CPT code inside index.php file

5) Activate the plugin.
6) Open the WordPress admin panel. At the left side of the panel, observe one newly added menu, '**Events**.'
7) From the left side menu bar of the WordPress admin panel, go to settings > Permalinks > click on "**Save changes**."
8) Now, custom CPT creation is Completed.

5.2 Add data to the custom post:

1) From the left side Admin menu bar, click on "**Events**."
2) On the top left, click on "**Add New**"
3) Fill title "Event-1", description, etc.
4) Click the "**Publish**" button from the right side.
5) Now, click " **Events** " from the left side." Observe that our newly created event is available.

5.3 Add meta box (additional fields):

To add a different field in the custom post type, we need to follow the below steps:

1) At the end of the "**index.php**," add the lines depicted in Figure 3.
2) Refresh the browser where the Admin panel is opened or open the WordPress admin panel from the local interface.

```
//////////////< Create Custom Meta Box>////////////////////////////////
function da_custom_meta_boxes()
{
 add_meta_box('da_cpt_id','Event Option', 'da_cpt_callback_func', 'event','normal','low');
}

add_action('add_meta_boxes','da_custom_meta_boxes');

function da_cpt_callback_func()
{
   wp_nonce_field(basename(__FILE__),'wp_da_cpt_nonce');
   $event_loction=get_post_meta(get_the_ID(),'event_location', true);
?>
    <div>
        <laber for="event_location">Event Location:</laber>
        <input type="text" id="event_location" name="event_location" value="<?php echo $event_loction ?>" >
    </div>
<?php
}

add_action('save_post','custom_cpt_save_meta_box',10,2);

function custom_cpt_save_meta_box($post_id,$post)
{
 if(!isset($_POST['wp_da_cpt_nonce']) || ! wp_verify_nonce($_POST['wp_da_cpt_nonce'],basename(__FILE__)))
    return;

    if('event' != $post->post_type)
    return;

    if(isset($_POST['event_location']))
    {
        $event_location=sanitize_text_field($_POST['event_location']);
        update_post_meta($post_id,'event_location',$event_location);
    }
}
////////////////////////////////////////////////////////////////////////
```

Fig. 3: the code of additional field addition

3) click on the "**Events**" menu from the left menu bar. click on "**Add New**." At the end, we will observe another field available, depicted in Figure 4.

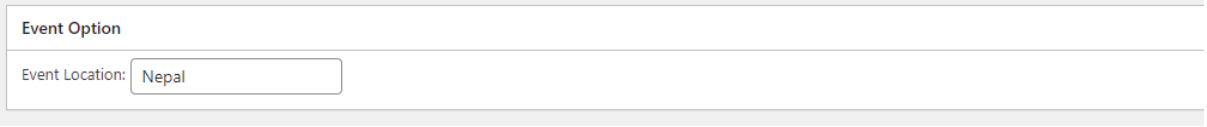

Fig. 4: additional field display

4) Fill the input box at the Event location input like "Nepal." Then click on "**Publish**".
5) Click on "**Events**" from the left side. The event location will not be observed. It is inside the table "**wp_postmeta**." To view this additional field from the local interface, click "**Open Adminer**."
6) From "**wp_posts**" table, get the "**post_id**" which we just added. We can sort "**post_date**" to find the latest event.
7) From the left, click on "**select**," which is beside "**wp_postmeta**." In the search box, select "**post_id**" ; in the third box, type "post_id" value and press enter. The "**meta_key**" "**event_location**" shows the "**meta_value**" "Nepal." Figure 4 depicts the event_location data.

```
101  ///////////////< Display Additional field inside the CPT interface>/////
102  add_action('manage_event_posts_columns','da_custom_cpt_column');
103
104  function da_custom_cpt_column($columns)
105  {
106      $custom_columns =[
107       'cb' => '<input type="checkbox"/>',
108       'title'=> 'Event Title',
109       'event_location'=> 'Event Location',
110       'date' => 'Date',
111      ];
112      return $custom_columns;
113  }
114
115  add_action('manage_event_posts_custom_column','da_cpt_custom_column_data',10,2);
116
117  function da_cpt_custom_column_data($columns,$post_id)
118  {
119      switch($columns)
120      {
121          case "event_location":
122              echo $event_location=get_post_meta($post_id,'event_location',true);
123              break;
124      }
125  }
126
127  add_filter('manage_edit-event_sortable_columns', 'da_cpt_sortable_columns');
128
129  function da_cpt_sortable_columns($columns)
130  {
131      $columns['event_location']='event_location';
132      return $columns;
133  }
134  ////////////////////////////////////////////////////////////////////////////
```

Fig. 5: Additional field data is displayed inside the "**wp_postmeta**" table.

5.4 Display Additional Field in the Custom Post Interface:

We need to follow the following steps-

1) Add the codes at the end of the "**index.php**." depicted in Figure 6.
2) Refresh the browser, and from the left side, open the "**Event**" post type and observe the "**event location**" displayed in a column.

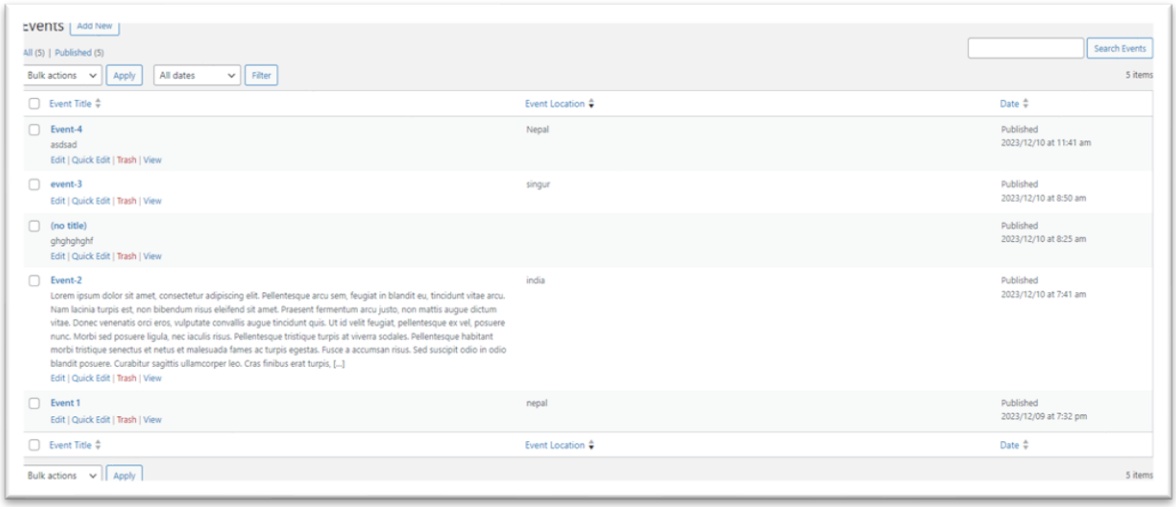

Fig. 6: Additional field data display inside the CPT Interface

We can sort content by clicking the column header "**Event Location**". It will be sorted in either ascending or descending order.

6. RECOMMENDATIONS :

- The project code is available from https://github.com/sudipchakraborty/Let-Us-Build-a-WordPress-Custom-Post-Type-CPT-.git
- There are more features available in WordPress. A good tutorial on WordPress custom post type: https://www.youtube.com/watch?v=6QS25lx8E-Q&list=PLwyFcQ0GXhmCywnYtqOWaXJMMQcOCZpQy
- Add exception handling code in the code if it is deployed in the production environment.
- One good course on WordPress development on udemy.com is "Become a WordPress Developer: Unlocking Power with Code" by Brad Schiff.

7. CONCLUSION :

WordPress is one of the most popular content management systems in the internet world. The custom post type has excellent features. We can customize our requirements using this CPT. Here, step by step, we demonstrate how we can create CPT. The researchers trying to implement CPT into their project can get valuable reference information from this work.

REFERENCES :

[1] Jones, K. M., & Farrington, P. A. (2013). Learning from WordPress libraries: Content-management system best practices and case studies. American Library Association. Google Scholar↗

[2] Fragulis, G. F., Lazaridis, L., Papatsimouli, M., & Skordas, I. A. (2018, September). ODES: an online dynamic examination system based on a CMS Wordpress plugin. In 2018 South-Eastern European Design Automation, Computer Engineering, Computer Networks and Society Media Conference (SEEDA_CECNSM) (pp. 1-8). IEEE. Google Scholar↗

[3] Jones, K. M., & Farrington, P. A. (2011). Using WordPress as a library content management system. American Library Association. Google Scholar↗

[4] Leary, S., & Leary, S. (2013). Custom Post Types, Taxonomies, and Fields. WordPress for Web Developers: An Introduction for Web Professionals, 291-324. Google Scholar↗

[5] Lakshmi, S. M., Mogili, U., Eluri, S., & Rao, D. R. Online Dynamic Out Patient Queue System for Automated Token Generation in Hospitals. Google Scholar↗

[6] Jones, K. M., & Alida-Farrington, P. (2011). Getting started with wordpress. *Library Technology Reports*, *47*(3), 8-15. Google Scholar↗

[7] Dimenstein, I. B., & Dimenstein, S. I. (2013). Development of a laboratory niche web site. *Annals of Diagnostic Pathology*, *17*(5), 448-456. Google Scholar↗

[8] Duong, B. Q., Arwood, M. J., Hicks, J. K., Beitelshees, A. L., Franchi, F., Houder, J. T., ... & IGNITE Network. (2020). Development of customizable implementation guides to support clinical adoption of pharmacogenomics: Experiences of the implementing genomics in practice (ignite) network. *Pharmacogenomics and Personalized Medicine*, 217-226. Google Scholar↗

[9] Weller, B., & Calcott, L. (2012). The definitive guide to Google AdWords: Create versatile and powerful marketing and advertising campaigns. Apress. Google Scholar↗

[10] Chakraborty, S., & Aithal, P. S. (2023). CRUD Operation on WordPress Database Using C# SQL Client. *International Journal of Case Studies in Business, IT, and Education (IJCSBE)*, *7*(4), 138-149. DOI: https://doi.org/10.5281/zenodo.10162719

[11] Chakraborty, S., & Aithal, P. S., (2023). CRUD Operation On WordPress Database Using C# And REST API. *International Journal of Applied Engineering and Management Letters (IJAEML)*, *7*(4), 130-138. DOI: https://doi.org/10.5281/zenodo.10197134

[12] Chakraborty, S., & Aithal, P. S., (2022). A Practical Approach To GIT Using Bitbucket, GitHub and SourceTree. *International Journal of Applied Engineering and Management Letters (IJAEML), 6*(2), 254-263. DOI: https://doi.org/10.5281/zenodo.7262771

[13] Chakraborty, S., & Aithal, P. S. (2023). Industrial Automation Debug Message Display Over Modbus RTU Using C#. *International Journal of Management, Technology, and Social Sciences (IJMTS), 8*(2), 305-313. DOI: https://doi.org/10.5281/zenodo.8139709

[14] Chakraborty, S., & Aithal, P. S. (2023). Modbus Data Provider for Automation Researcher Using C#. *International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7*(3), 1-7. DOI: https://doi.org/10.5281/zenodo.8162680

[15] Chakraborty, S., & Aithal, P. S., (2023). MVVM Demonstration Using C# WPF. *International Journal of Applied Engineering and Management Letters (IJAEML), 7*(1), 1-14. DOI: https://doi.org/10.5281/zenodo.7538711

[16] Chakraborty, S., & Aithal, P. S. (2023). CRUD Operation on WordPress Database Using C# SQL Client. *International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7*(4), 138-149. DOI: https://doi.org/10.5281/zenodo.10162719

[17] Chakraborty, S., & Aithal, P. S., (2023). CRUD Operation On WordPress Database Using C# And REST API. *International Journal of Applied Engineering and Management Letters (IJAEML), 7*(4), 130-138. DOI: https://doi.org/10.5281/zenodo.10197134

****