

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

на тему: **«РОЗРОБКА ВІДЖЕТУ ДЛЯ  
ІНТЕГРАЦІЇ OPENSTREETMAP В CMS  
DRUPAL**

Виконав(ла): студент(ка) 2 курсу, групи 8.1219-з

спеціальності 121 інженерія програмного забезпечення  
(шифр і назва спеціальності)

освітньої програми інженерія програмного забезпечення  
(назва освітньої програми)

О.І. Малих

(ініціали та прізвище)

Керівник доцент кафедри фундаментальної  
математики, к.ф.-м.н. Панасенко Є.В.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент доцент кафедри комп'ютерних наук,  
доцент, к.т.н. Решевська К.С.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти магістр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма інженерія програмного забезпечення

**ЗАТВЕРДЖУЮ**

Завідувач кафедри програмної  
інженерії, к.ф.-м.н., доцент  
Лісняк А.О.

(підпис)

«        »

2020 р.

**З А В Д А Н Н Я**

**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ(СТУДЕНТЦІ)**

Малих Олександрю Ігоровичу

(прізвище, ім'я та по-батькові)

1. Тема роботи (проекту) Розробка віджету для інтеграції OpenStreetMap в CMS Drupal

керівник роботи (проекту) Панасенко Євген Валерійович, к.ф.-м.н., доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затвержені наказом ЗНУ від « 20 » травня 2020 року № 577-с

2. Строк подання студентом роботи 25.11.2020

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Провести порівняльний аналіз поширених CMS, проаналізувати основні програмні засоби для створення модуля карт у Drupal, спроектувати та розробити модуль для інтеграції карт OpenStreetMap до CMS Drupal.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Панасенко Є.В., к.ф.-м.н., доцент		
2	Панасенко Є.В., к.ф.-м.н., доцент		
3	Панасенко Є.В., к.ф.-м.н., доцент		

7. Дата видачі завдання \_\_\_\_\_ 12.08.2020 \_\_\_\_\_

**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	26.08.2020	Виконано
2.	Збір вихідних даних.	09.09.2020	Виконано
3.	Обробка методичних та теоретичних джерел	16.09.2020	Виконано
4.	Розробка першого розділу.	07.10.2020	Виконано
5.	Розробка другого розділу.	28.10.2020	Виконано
6.	Розробка третього розділу.	18.11.2020	Виконано
7.	Оформлення та нормоконтроль кваліфікаційної роботи.	25.11.2020	Виконано
8.	Захист кваліфікаційної роботи.	09.12.2020	Виконано

Студент \_\_\_\_\_  
(підпис)

О.І. Малих \_\_\_\_\_  
(ініціали та прізвище)

Керівник роботи \_\_\_\_\_  
(підпис)

Є.В. Панасенко \_\_\_\_\_  
(ініціали та прізвище)

**Нормоконтроль пройдено**

Нормоконтролер \_\_\_\_\_  
(підпис)

О.В. Кудін \_\_\_\_\_  
(ініціали та прізвище)

## РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка віджету для інтеграції OpenStreetMap в CMS Drupal»: 52 с., 22 рис., 5 табл., 19 джерел.

СИСТЕМА УПРАВЛІННЯ КОНТЕНТОМ, OPENSTREETMAP, МОДУЛЬ, СТРУКТУРА, ПЛАГІН, ВІДЖЕТ, ВЕБ-СЕРВЕР, САЙТ, ШАБЛОН.

Об'єкт дослідження – етапи розробки плагіна карт для CMS Drupal.

Предмет дослідження – плагін для інтеграції OpenStreetMap в CMS Drupal.

Мета роботи – розробка віджету для інтеграції OpenStreetMap в CMS Drupal.

Метод дослідження – аналітичний, пошуковий, порівняльний.

У кваліфікаційній роботі розглянуто особливості розробки віджету для інтеграції карт OpenStreetMap і розроблено модуль з плагіном для CMS Drupal. Розглянуто основні поняття – система управління контентом, модуль, сайт, структура, віджет. Розглянуто найпоширеніші системи управлінням контентом, модулі карт які вже розроблені у CMS Drupal. Розроблений модуль з віджетом можна використовувати як ресурс відображення необхідної інформації на карті власного сайту.

## SUMMARY

Master's Qualification Thesis «Development of a Widget for Integration OpenStreetMap to Drupal CMS»: 52 pages, 22 figures, 5 tables, 19 references.

CONTENT MANAGEMENT SYSTEM, OPENSTREETMAP, MODULE, STRUCTURE, PLUGIN, WIDGET, WEB SERVER, WEBSITE, TEMPLATE.

The object of research is the stages of creating a map plugin for CMS Drupal.

The research subject is a plugin for integrating OpenStreetMap into CMS Drupal.

The aim of the study is to develop a widget for integrating OpenStreetMap into the Drupal CMS.

The methods of research are analytical, comparative, analysis and searching of information.

The qualification work considers the features of the development of a widget for the integration of OpenStreetMap maps and developed a module with a plugin for CMS Drupal. The basic concepts are considered - content management system, module, site, structure, widget. The most common content management systems, map modules that have already been developed in the Drupal CMS are considered. The developed module with a widget can be used as a resource for displaying the necessary information on the map of your own site.

## ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат.....	4
Summary.....	5
Вступ.....	7
1 Загальний огляд поширених CMS, локальних серверів та проекту OpenStreetMap.....	9
1.1 Поняття CMS.....	9
1.2 Загальний огляд поширених систем управлінням контенту та їх порівняння.....	10
1.3 Загальний огляд OpenStreetMap.....	19
1.4 Загальний огляд та порівняння локальних серверів.....	20
2 Проектування модуля відображення матеріалів на карті.....	24
2.1 Архітектура CMS Drupal.....	24
2.2 Протікання робочого процесу у Drupal.....	26
2.3 Діаграма компонентів.....	28
2.4 Діаграма варіантів використання (прецедентів).....	30
3 Етапи розробки плагіна інтеграції OpenStreetMap.....	35
3.1 Базові налаштування модуля Drupal.....	35
3.2 Реалізація хуків.....	37
3.3 Налаштування відображення матеріалу.....	39
Висновки.....	49
Перелік посилань.....	51

## ВСТУП

У сучасних умовах важко уявити навколишній світ без інформаційних технологій та Інтернету, який все більше стає невід'ємною частиною нашого життя. В Інтернет переноситься бізнес, телефонія, керування системами безпеки та системами власних будинків.

Веб-сфера має дуже стрімкий розвиток у наші дні. Так приблизно за тридцять років від створення першого сайту ми бачимо наскільки вони стали яскравішими та різноманітнішими за призначеннями: торгові, інформаційні, розважальні, соціальні мережі.

Проблемою перших сайтів певний час було те, що вони були статичними, потому вносити будь які зміни до їх змісту потребувало зусиль фахівців розробників, котрі знали архітектуру сайту. Однак зараз швидкість розвитку веб-сервісів потребує миттєвої реакції та готовності вносити зміни до порталів, чим і скористались розробники створивши CMS (Content Management System) – система яка дозволяє створювати та редагувати зміст сайту. Більшість CMS використовують принцип модульності у якому ядро сайту/CMS залишається незмінним, а за потреби додаткових можливостей досвідчений користувач або розробник може їх здійснити завдяки додаткових модулів.

Drupal – CMS з відкритим вихідним кодом, на базі якої можна створювати будь-які сайти, проте найчастіше його використовують у великих проектах, де необхідна велика продуктивність.

У порівнянні з іншими поширеними CMS, Drupal більш потужний та гнучкий завдяки більшим можливостям та налаштуванням. Разом з тим, налаштування сайту на Drupal буде більш складним, проте у нього є багато переваг:

- 1) Drupal відрізняється великою захищеністю та продуктивністю на базовому рівні;

2) можна адаптувати для простих сайтів або для складних веб-додатків, система дуже гнучка;

3) велика кількість функцій: порівнюючи з іншими системами управління контентом.

Дані у контенті сайту зберігаються в одному або декількох полях, які прив'язані до типу сутності та/або підтипу. При настройці форми редагування контенту для кожного підтипу контенту можна обрати віджет. Віджет визначає метод який буде використовуватись для вводу даних у поле.

У даній роботі поставлена задача розробити віджет для інтеграції OpenStreetMap в CMS Drupal. Отриманий у результаті віджет допоможе у створенні міток, ліній і областей за допомогою OpenStreetMap з можливістю вказувати свої параметри для кожного об'єкту.



# 1 ЗАГАЛЬНИЙ ОГЛЯД ПОШИРЕНИХ CMS, ЛОКАЛЬНИХ СЕРВЕРІВ ТА ПРОЕКТУ OPEN STREET MAP

## 1.1 Поняття CMS

На сьогоднішній день більш ніж половина усіх сайтів у мережі Інтернет використовує ту або іншу систему управління контентом (англ., Content Management System, CMS) [1]. Однак термін CMS, так і не отримав чіткого визначення. Він може мати декілька значень в залежності від цілей людини або проекту.

Некомерційна міжнародна організація АІМ (англ., Association for Information and Image Management – Асоціація по питанням Управління Інформацією та Зображеннями) інтегрувала поняття ЕСМ (англ., Enterprise Content Management, ECM) та WCM (англ., Web Content Management, WCM – система управління веб-контентом) як дві складові частини CMS [2].

У цьому випадку під ЕСМ мається на увазі програмний комплекс, який забезпечує документообіг, роботу внутрішньої бази знань організуючої у загальному випадку набір бізнес-процесів підприємства. Як одну із функцій ЕСМ може включати у себе і роботу із веб-контентом (наприклад платформа Microsoft SharePoint). WCM стало включати у себе деякий комплекс набору інструментів, які дозволяють керувати веб-сайтом та його контентом.

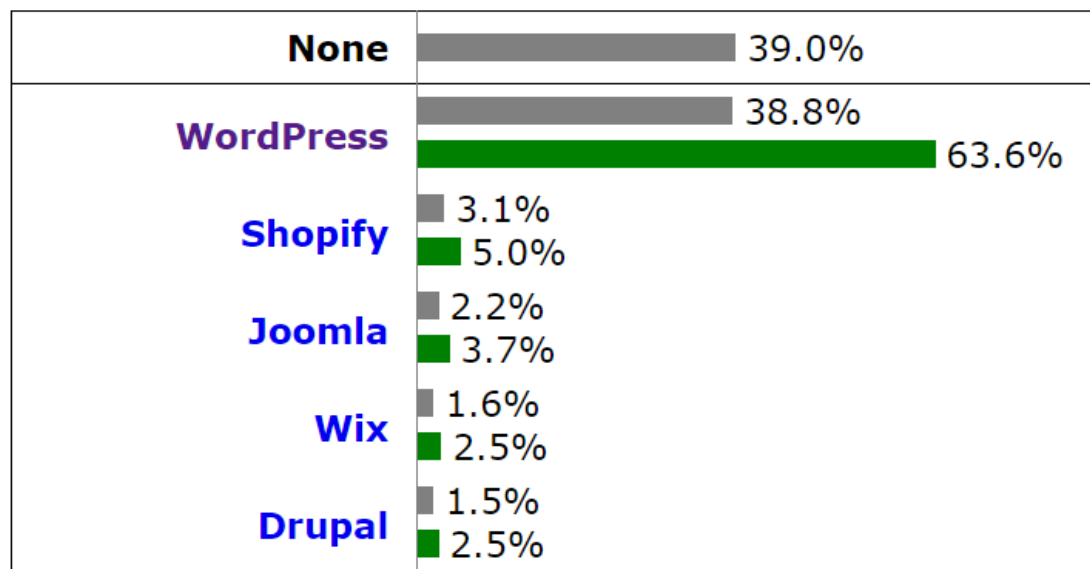
Поширене визначення CMS яку ще називають «двигуном» сайту, мається на увазі деяка програмна система, яка використовується для створення, редагування та керуванням змістом (контентом) сайту і інсталяції на власний хостинг [1]. Це визначення дуже схоже на визначення WCM, яке ми і визначимо у нашій роботі як одне дійсне.

## 1.2 Загальний огляд поширених систем управління контентом та їх порівняння

Великий вибір різноманітних систем управління контентом на ринку пояснюється відмінностями у типах сайтів, для яких вони відповідатимуть як найкраще (будь то форум, інтернет магазин або сайт-візитка).

З іншого боку не дивлячись на деяку спеціалізацію кожної CMS під відповідну задачу, на сьогоднішній день існує велика кількість різноманітних доповнень (плагінів, модулів) для поширених CMS, які суттєво розширюють їх функціональні можливості. Однак потрібно розуміти, що збільшення кількості додаткових модулів може відобразитися на швидкості роботи, безпеці та внутрішній злагодженій роботі механізмів сайту тобто можливих конфліктів цих модулів.

Більш ніж 61% усіх Web-сайтів у мережі Інтернет керуються тими чи іншими CMS [3].



 - відсоток від загальної кількості сайтів з CMS

 - відсоток від загальної кількості сайтів

Рисунок 1.1 – Діаграма використання систем управління контентом

Як бачимо з рисунку 1.1 більш ніж 38% усіх сайтів у Інтернеті працюють на двигуні WordPress. Фактично це більше 63% від сайтів, які керуються тими чи іншими CMS. Наступні за поширеністю системи CMS: Joomla – 3.7% та Drupal – 2.5%.

Shopify та Wix які набрали досить багато відсотків у нашому рейтингу пропонують послуги Saas (англ., software as a service, Saas – це послуга доступу до програмного забезпечення). Провайдери послуги Saas пропонують для своїх клієнтів власні платформи, із власним хостингом та власними індивідуальними можливостями. В даній роботі такий тип послуг не буде розглянуто як окрема система CMS.

За даними вже іншого порталу WhatCms перше місце за кількістю сайтів поміж поширених CMS належить WordPress – 29.49%. Далі йдуть Joomla – 8.3% та Drupal – 1.6% [4]. Blogger як і декілька компаній з попередньої діаграми – це Saas платформи (див. рис. 1.2).

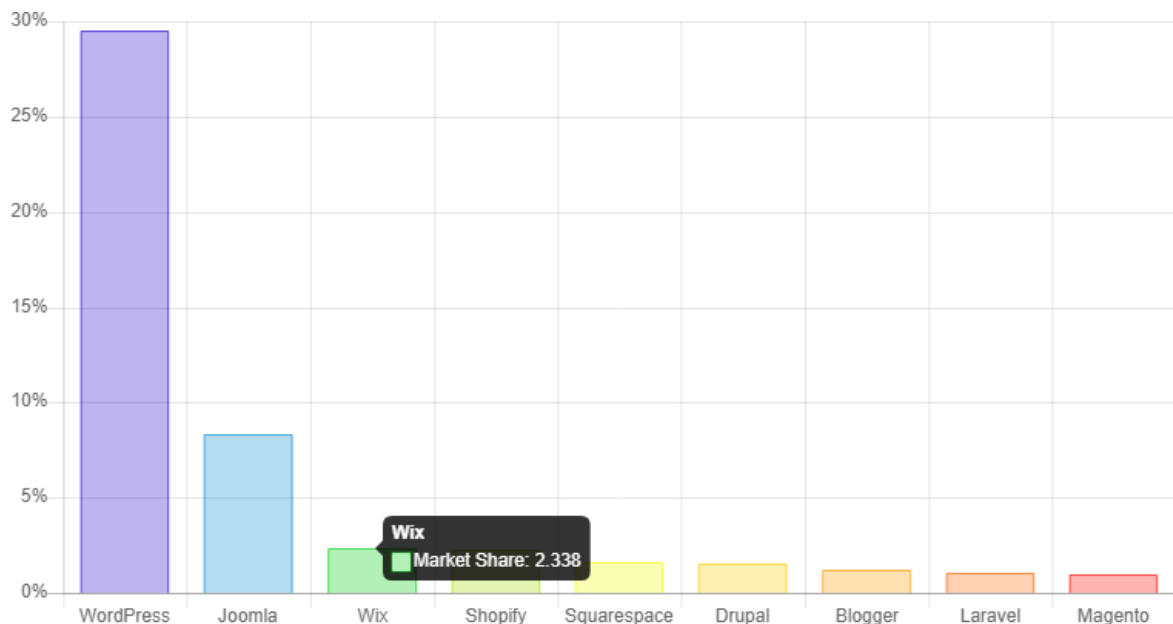


Рисунок 1.2 – Діаграма використання поширених CMS порталу WhatCms

Проаналізувавши вказану статистику, у роботі обрано наступні три найпоширеніші CMS: WordPress, Joomla та Drupal. Проведемо їх огляд для подальшого аналізу.

WordPress уперше був випущений у 2003 році і доволі швидко здобула популярність як у доволі досвідчених розробників, так і серед звичайних користувачів. Завдяки простому налаштуванню, не найвищим вимогам до ресурсів хостингу та великій кількості розширень ця CMS вже не один рік займає перше місце. На сьогоднішній день WordPress виступає кращою CMS для блогу. WordPress вдало підходить для доволі нескладних веб-сайтів, таких як щоденні блоги і сайти новин, а також для тих хто шукає для себе нескладну CMS. Доповнення дозволяють легко розширити функціональність сайту. Треба відмітити і досить велику кількість SaaS рішень, який використовує на своїй платформі цей двигун. Частина успіху у діаграмах представлених на рисунках 1.1 та 1.2, без сумніву відноситься до SaaS рішень.

Особливості WordPress:

- а) остання версія – 5.5.3;
- б) написано на PHP;
- в) версії старші за 5.0 офіційно оголошені «небезпечними»;
- г) мінімальні вимоги до хостингу, підтримку яких обіцяє розробник:
  - PHP 7.3;
  - MySQL 5.6 або MariaDB 10.0;
  - HTTPS;
  - Apache або Nginx [5].

Переваги WordPress:

- а) безкоштовна CMS поширюється з відкритим вихідним кодом;
- б) велика кількість як платних, так і безкоштовних шаблонів та плагінів;
- в) зручна панель адміністрування;
- г) проста CMS для користувача;
- д) велика спільнота;
- е) доволі висока продуктивність;

ж) доступні платні плагіни з перевіреною якістю.

Недоліки WordPress:

а) відносно не маленька вимога до ресурсів, особливо при інсталяції значної кількості плагінів;

б) відсутність технічної підтримки у не Saas варіантах;

в) багато плагінів написано неякісно, тим самим створюючи проблеми у роботі та пробіли у безпеці;

г) сайти на WordPress зламують найчастіше.

Для яких сайтів використовують CMS WordPress у процентному відношенні показано на рисунку 1.3 [6]:

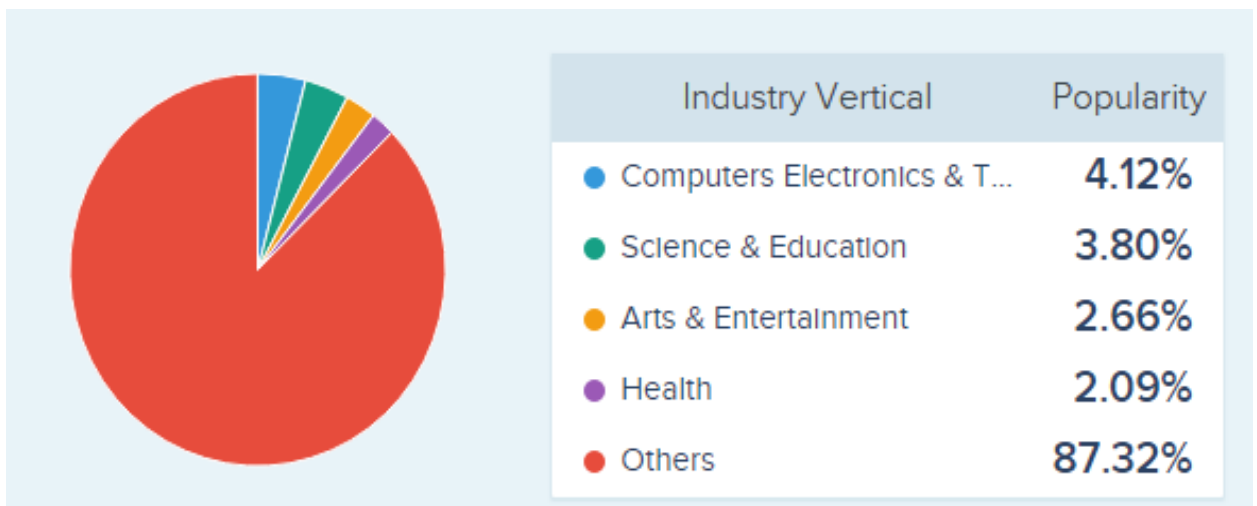


Рисунок 1.3 – Діаграма галузевої вертикалі використання CMS WordPress

Проаналізувавши діаграму (див. рис. 1.4) яка включає у собі всі сайти у мережі Інтернет, починаючи з 2011 року популярність CMS WordPress впевнено зростає порівнюючи з кожним попереднім роком. Пов'язати цей зріст можна з тим, що дана система має зручний інтерфейс та досить легка в опануванні [7].

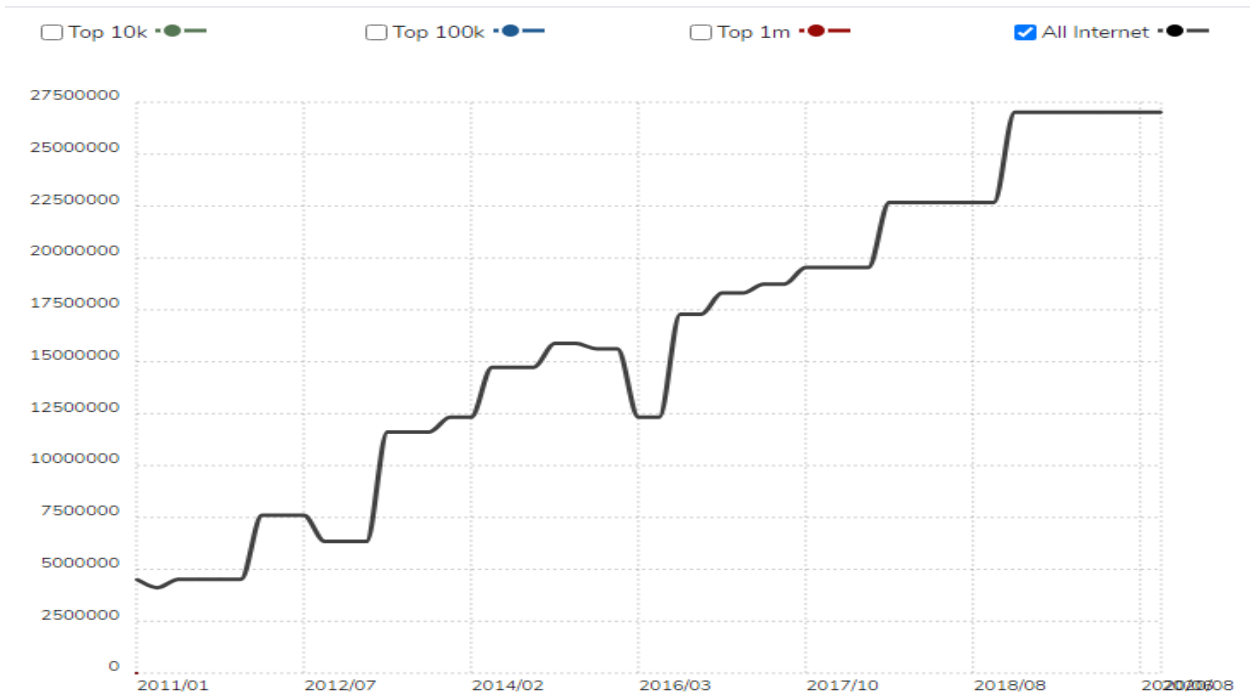


Рисунок 1.4 – Діаграма використання WordPress

CMS Joomla вийшла у 2005 році. Віддзеркалюючи філософію цього двигуна, його назвали словом, що звучить на суахілі як «все разом». Фактично розробляється як CMS для порталів, Joomla дозволяє створювати сайти з більшою гнучкістю контенту і внутрішньої структури, ніж WordPress, але при цьому з досить простим і інтуїтивно зрозумілим інтерфейсом. Ця CMS підтримує електронну комерцію, соціальні мережі та багато іншого. Використовуючи цей двигун розробники створюють сайти-візитки, інтернет-магазини, фотогалереї та інші сайти.

Особливості двигуна Joomla:

- а) остання версія 3.9.22 [8];
- б) написана на PHP та JavaScript;
- в) мінімальні системні вимоги:
  - PHP 5.3.20;
  - MySQL 5.1 або SQL Server 10.50.1600.1 або PostgreSQL 8.3.18;
  - Apache 2.0 або Nginx 1.0 або Microsoft IIS 7.

Переваги Joomla:

- а) безкоштовне розповсюдження з відкритим вихідним кодом;
- б) часте надання оновлень двигуна;
- в) велике співтовариство користувачів і розробників;
- г) велика кількість доступних платних і безкоштовних тем та плагінів;
- д) відносно невисокий рівень знань розробника та користувача.

Недоліки Joomla:

- а) відсутність технічної підтримки;
- б) друга CMS по кількості зламів.

У яких сферах використовується Joomla наведено на рисунку 1.5 [6]:

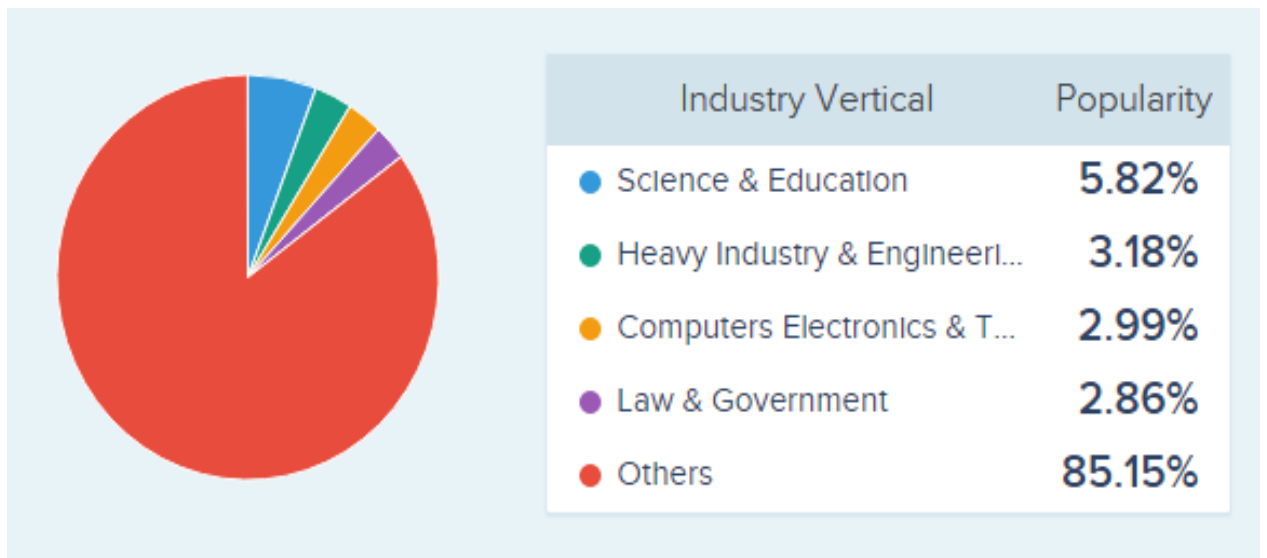


Рисунок 1.5 – Діаграма галузевої вертикалі використання CMS Joomla

У даної CMS спостерігається певна тенденція щодо зниження популярності наведена на діаграмі (див. рис. 1.6) [7].

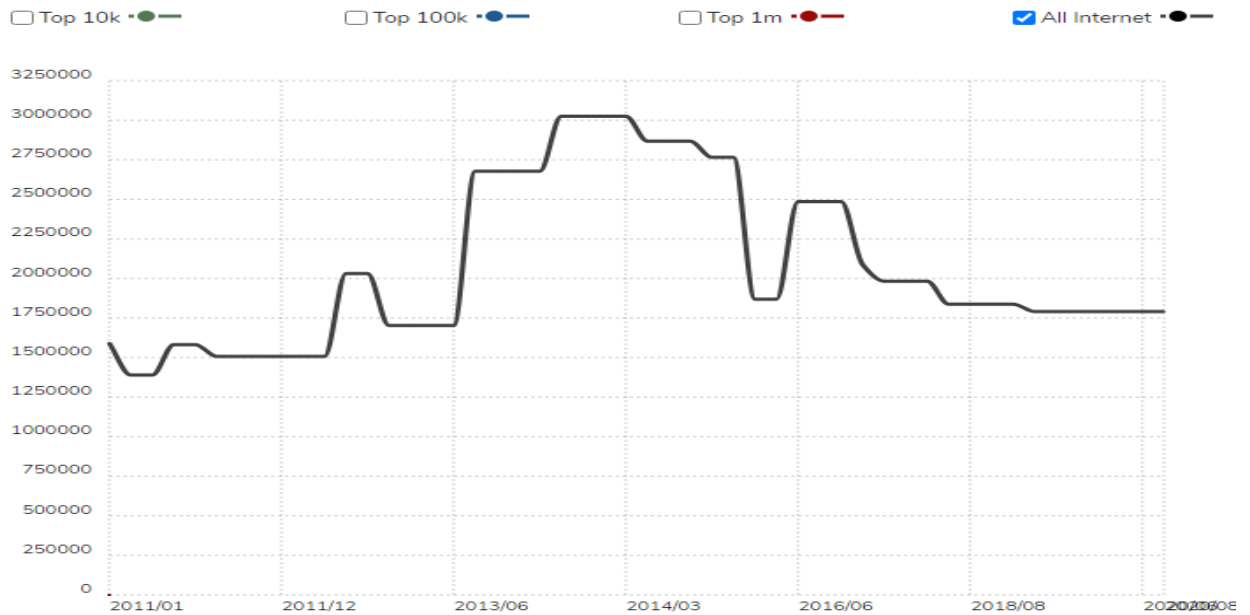


Рисунок 1.6 – Діаграма використання Joomla

CMS Drupal вперше вийшла у 2000 році і є потужним, зручним для розробників інструментом для створення складних сайтів. Як і більшість потужних інструментів, Drupal вимагає певних знань і досвіду для роботи. На основі Drupal часто створюють портали, новинні сайти, форуми, інтернет-магазини – одні з найбільш просунутих сайтів.

Проте Drupal є найскладнішим для користувача движків з трійки лідерів. Хоча його використання з кожним випуском і стає все простіше.

Особливості Drupal CMS:

- а) остання версія 9.0.7 [9];
- б) ядро надає тільки мінімальний функціонал, потрібний для роботи CMS, більше можливостей додається завдяки додатковим модулям;
- в) інсталяція модулів виконується у зв'язці. Якщо для реалізації функціоналу якогось модуля потрібні інші – то вони встановлюються автоматично з першим;
- г) мінімальні вимоги хостингу для CMS Drupal 8:
  - PHP 5.x/7.x для x86 та PHP 5.x для x64;
  - MySQL 5.5.3 або MariaDB 5.5.20, або Percona Server 5.5.8, або PostgreSQL 9.1.2., або SQLite 3.6.8 [15];



- Microsoft SQL та Server MongoDB підтримуються завдяки окремим модулям;

- Apache 2.x (використовується у якості Web-сервера для Drupal найчастіше) або Nginx (0.7x, 0.8x, 1.0x, 1.2x), стабільна версія 1.8x або 1.9x.

Переваги Drupal:

- а) безкоштовна CMS з відкритим вихідним кодом;
- б) стабільна робота ядра двигуна;
- в) велика кількість безкоштовних тем, і різних розширень;
- г) досить розвинене співтовариство розробників;
- д) для вирішення типових задач є готові набори плагінів;
- е) Drupal відомий своєю потужною таксономією та здатністю відмічати, класифікувати та організовувати складний контент;
- ж) вбудована система хешування, що забезпечує прискорення завантаження сторінки.

Недоліки Drupal:

- а) складність використання для початківців користувачів;
- б) менша кількість доступних безкоштовних плагінів ніж у попередніх;
- в) відзначають велику вимогливість до хостингу за рахунок більш частих звернень двигуна до бази даних, ніж у інших двигунів.

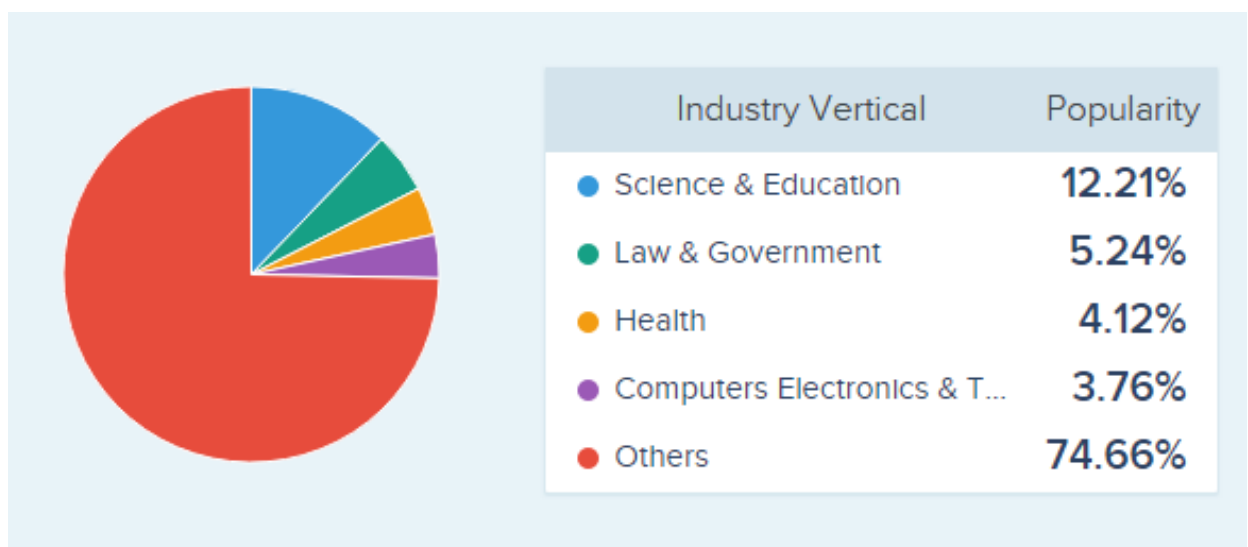


Рисунок 1.7 – Діаграма галузевої вертикалі використання CMS Drupal

У яких сферах використовується Drupal наведено на рисунку 1.7 [6].

У даної CMS спостерігається певна тенденція щодо збереження популярності наведена на діаграмі (див. рис. 1.8) [7], це пояснюється тим, що інтерфейс Drupal постійно вдосконалюється з кожним випуском оновлень. Починаючи з 8.x версії система перейшла на об'єктно орієнтовне програмування.

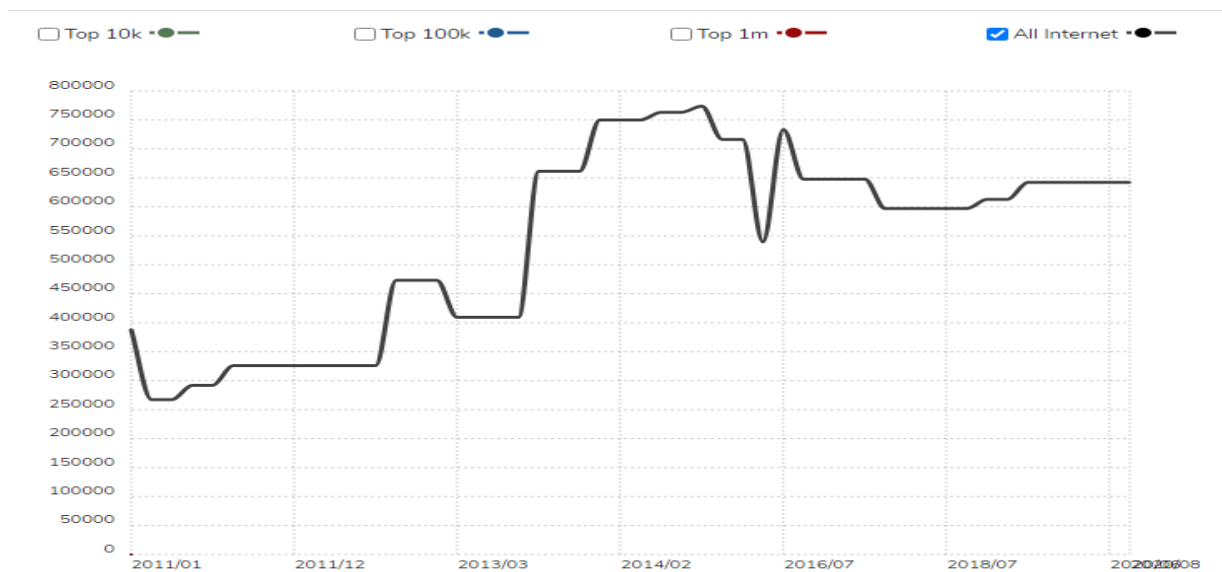


Рисунок 1.8 – Діаграма використання Drupal

У розділі ми розглянули та порівняли три найпоширеніші у світі CMS такі як WordPress, Joomla та Drupal. Кожна з них має свою специфіку та особливості. Через різноманітні можливі сфери застосування складно обрати кращу систему керування сайтом. Вибираючи Drupal, необхідно оцінити його функціональність яку можна отримати завдяки додатковим плагінам і складність освоєння. Двигун ефективний, коли потрібно створити великий проект, який буде відрізнятися стабільністю і високим рівнем безпеки. Зупинимо свій вибір на CMS Drupal.

### 1.3 Загальний огляд OpenStreetMap

OpenStreetMap – міжнародний проєкт, головна мета якого є розробка вільної, відкритої карти світу (див. рис. 1.9) [12]. У загальний принцип розробки карт на порталі OSM (англ., OpenStreetMap, OSM) було закладено принципи вікі. Який полягає у тому, що кожен охочий після створення акаунту на даному веб-сервісі зможе додати та видалити дані до карти, тобто займатись її редагуванням.

При порівнянні цього сервісу з сервісами від Google та Yandex то у містах останні більш деталізовані, проте на територіях де часто проходять туристи або сільська місцевість OSM дає кращі данні завдяки GPS трекам.

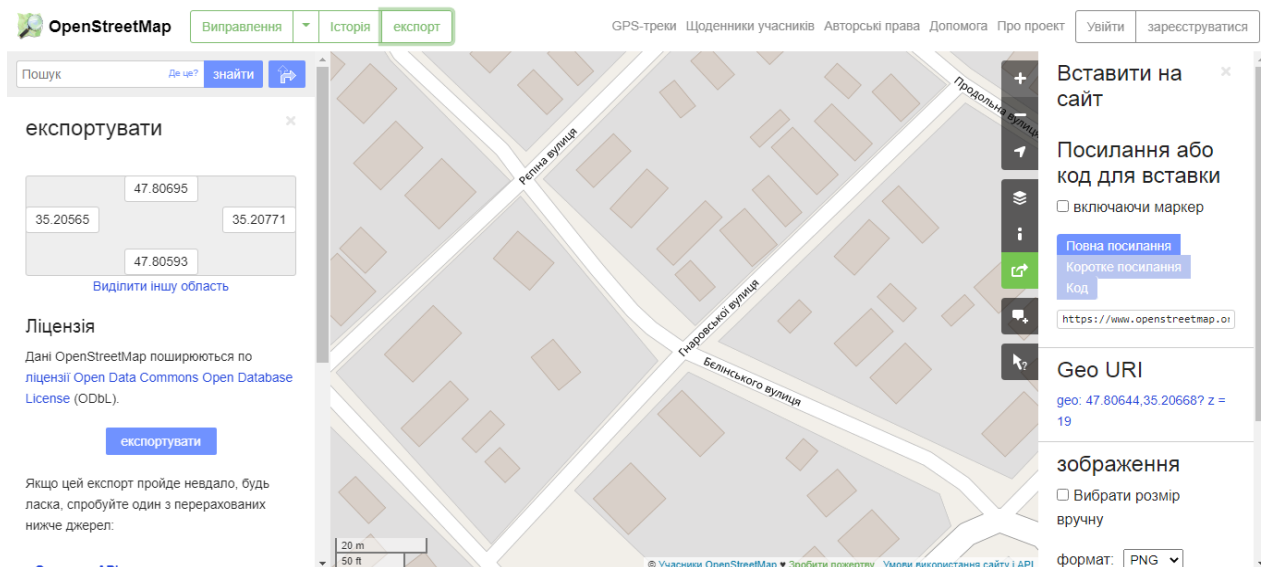


Рисунок 1.9 – Головна сторінка проєкту OpenStreetMap

Фундація OpenStreetMap – організація, що здійснює збір коштів для підтримки роботи серверів, на яких працює проєкт OpenStreetMap. Її основне завдання – підтримка проєкту, вона не керує проєктом та «не володіє» даними OSM.

Переваги:

а) швидкість оновлення даних стосовно редагувань які були внесені до карти;

б) велика деталізація у певних місцях стосовно інших порталів які надають свої карти;

в) різноманітність шарів таких як: велосипедні доріжки, транспортні та гуманітарні карти;

г) відкрита можливість до редагування даних;

д) безкоштовне використання;

е) велика кількість сторонніх сервісів які надають різні плитки на дані OpenStreetMap.

Недоліки:

а) неможливість надати API як у інших сервісах;

б) відкритість до недосвідчених користувачів;

в) відсутність якісних супутникових знімків.

#### **1.4 Загальний огляд та порівняння локальних серверів**

Локальний сервер – це програмне забезпечення, яке дозволяє створити аналог хостингу на власному комп'ютері, тим самим дає можливість розробляти та тестувати динамічні сайти з використанням серверних мов програмування та баз даних.

Існує доволі велика кількість локальних серверів, таких як: OpenServer, Denwer, WampServer, AMPPS та інші. У роботі зупинимось і розглянемо один локальний сервер такий як OpenServer.

OpenServer – це портативна WAMP (англ., Windows, Apache, MySQL і PHP, WAMP) платформа, потужний програмний комплекс серверного програмного забезпечення. Головним призначенням даного проекту є розробка, тестування та налагоджування веб-проектів [10].

Для початку роботи з додатком необхідно запустити виконуючий файл Open Server.exe, після цього у системній панелі з'явиться значок у вигляді прапорця (див. рис. 1.10). Запуск, зупинка, перезавантаження і налаштування

серверу, а також його усіх модулів або додаткових програм відбувається після натискання по цьому значку.

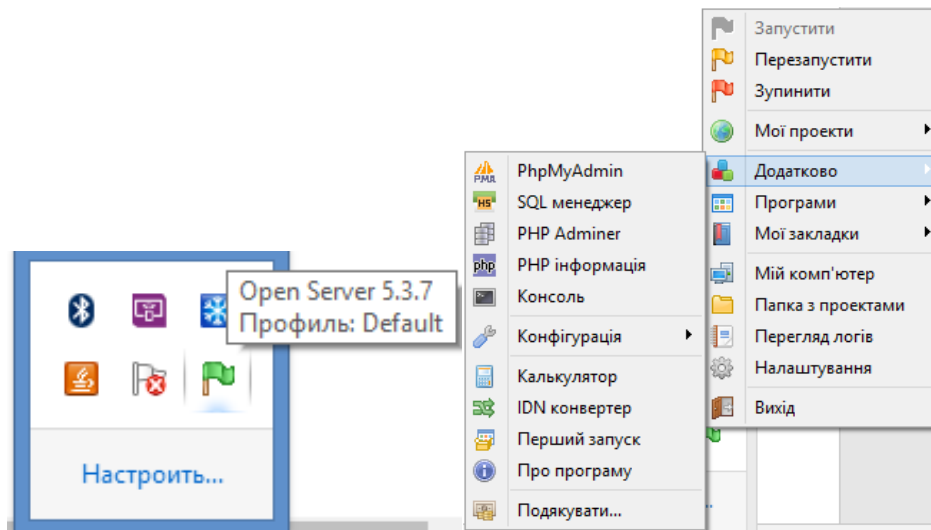


Рисунок 1.10 Позначення Open Server у системній панелі

Панель керування Open Server складається з 12 основних команд (див. рис. 1.10). Верхні три пункти відповідають за запуск, перезавантаження та зупинку серверу відповідно. Пункт «Мої проекти» відкриває доступ до проектів завдяки вбудованому браузеру. Задля збереження сайтів, скриптів та інших веб-додатків використовується папка «domains». Пункт «Додатково» містить команди виклику різноманітних серверних додатків – РНРMyAdmin, MySQL менеджера, консолі і інші. Наступні пункти відкривають доступ до різноманітних призначених для користувача додатків. Пункт «Налаштування», вікно якого містить 13 вкладок (див. рис. 1.11), кожна з яких відповідає відповідні параметри. Наприклад, у вкладці «Сервер» можливо створити та налаштувати віртуальні диски, змінити налаштування портів, шлях до проектів. Після внесення змін у налаштуваннях необхідно виконати перезавантаження серверу.

Open Server – доволі перспективний локальний веб-сервер який об'єднує у собі максимальну простоту та досить високу функціональність. Завдяки таким якостям додаток може бути використано як початківцями, так і

професійними веб-розробниками. Програмне забезпечення постійно вдосконалюється та оновлюється, додатком до цього буде й те, що Open Server цілком безкоштовне.

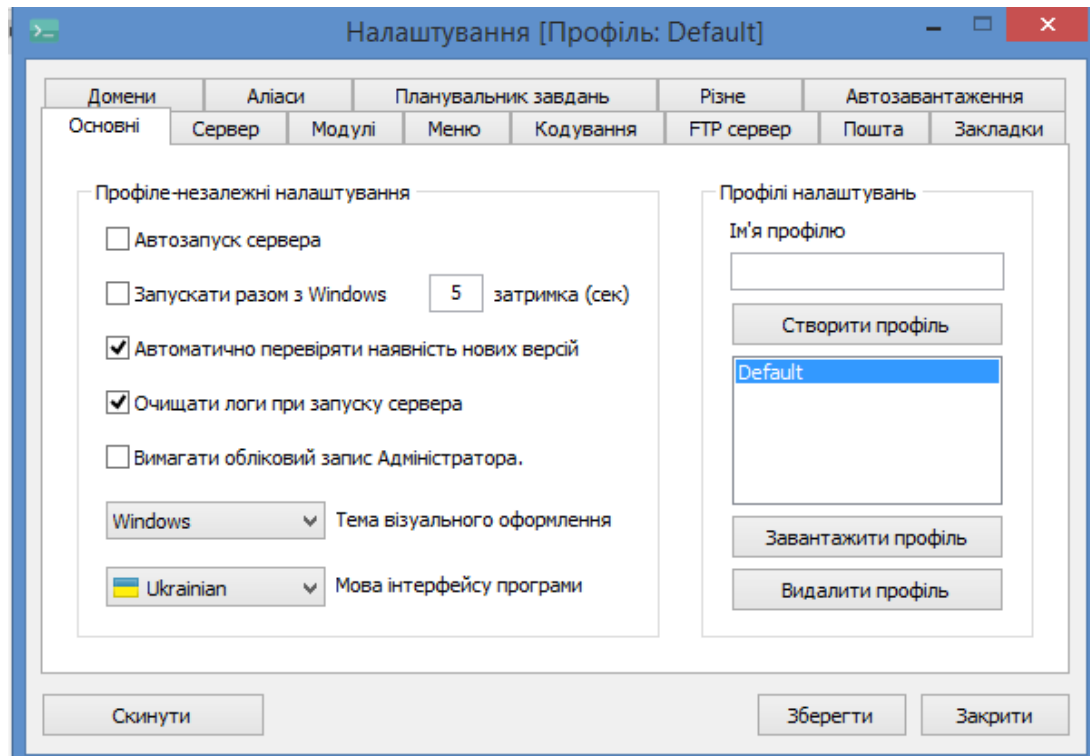


Рисунок 1.11 – Вкладка «Налаштування» у Open Server

Підсумовуючи матеріал викладений у першому розділі роботи, можна зробити такі висновки, що більш ніж половина сайтів у мережі Інтернет використовують ту чи іншу систему управління контентом. Також визначились що Drupal входить до трійки найпоширеніших CMS, таких як WordPress та Joomla, має певну тенденцію щодо збереження своєї популярності. CMS Drupal буде встановлено на локальний сервер – Open Server на якому буде проводитись розробка, тестування та налагоджування сайту з модулем карт.

Розглянуто міжнародний проєкт такий як OpenStreetMap визначені його недоліки та переваги поміж яких слід зазначити відкритість даних, постійне їх оновлення та можливість власне додати власні корегування до карти у певному районі.

До переваг Drupal при розробці віджета карт можна віднести:

- а) безкоштовна CMS з відкритим вихідним кодом;
- б) високий рівень безпеки;
- в) доволі високий рівень функціональності, що дозволяє поєднати майбутній модуль з існуючими плагінами.

## 2 ПРОЄКТУВАННЯ МОДУЛЯ ВІДОБРАЖАННЯ МАТЕРІАЛІВ НА КАРТІ

### 2.1 Архітектура CMS Drupal

Drupal працює на будь якій операційній системі будь-то: Windows, Linux, MAC OS. Drupal, як і інші системи управління контенту по суті являє собою графічний інтерфейс між операційною системою і користувачем [11].

Структура Drupal, як і будь якого динамічного сайту слід розуміти ні з самої системи управління контенту, а з точки зору операційної системи на яку його встановлено. За для функціонування Drupal необхідно: операційна система, сервер бази даних (MySQL), веб-сервер (Apache) та PHP сервер.

Операційна система, на яку встановлюються сервера, необхідні для роботи Drupal, вона організує мережеві з'єднання з сайтом Drupal. Доступ до сайту за протоколами передачі даних HTTP здійснюється за допомоги веб-серверу чи HTTP серверу такого як Apache. Уся інформація сайту така як: текстовий зміст, різні параметри конфігурацій Drupal та початкові налаштування сайту зберігаються на сервері баз даних. Взаємодія між веб-сервером і сервером баз даних здійснює PHP сервер, його задача динамічне створення сторінки сайту, а саме переміщувати дані з веб-сервері до баз даних і назад.

CMS Drupal складається із декількох рівнів або шарів. Найнижчий шар системи відповідає за підсистему безпеки Drupal, систему відображення шаблонів, механізмом керування доступу та обробку сеансів користувачів. Над базовим рівнем розміщується шар додаткових модулів які доповнюють базові можливості додатковим функціоналом [17]. Останній рівень системи, уявляє зручний інтерфейс для користувачів тобто система тем оформлення Drupal. Сторінки сайту на Drupal для графічного оформлення використовують



каскадні таблиці стилів CSS, а для додавання динамічності HTML елементам мову JavaScript з бібліотекою JQuery.

Загальна структура Drupal наведена на рисунку 2.1:



Рисунок 2.1 – Загальна структура Drupal

Майже вся зовнішня частина системи Drupal базується на понятті модуля. Модуль Drupal – це файл або декілька файлів, які написано на мові PHP з можливістю під'єднання JavaScript файлів, та реалізуючи ту чи іншу функцію, з якою вміє працювати CMS Drupal. Усі можливості CMS які доступні як адміністратору так і користувачу реалізовані завдяки модулів Drupal. За допомоги модулів реалізується такі можливості як: вхід до системи, додавання нового змісту до сайту, створення власних карт і інше. У Drupal існує два типу модулів: базові модулі, які встановлюються разом з CMS Drupal, вони задають той мінімалістичний функціонал сайту та зовнішні, які можливо під'єднати у процесі розробки сайту на Drupal, їх використовують за необхідністю розширення функціональних можливостей або зменшення завантаженості сайту. Після під'єднання модулів їх можна тільки видаляти, окрім декількох базових без наявності яких сайт не буде працювати.

## **2.2 Протікання робочого процесу у Drupal**

Для докладнішого розуміння, як інформація протікає поміж рівнів системи, розглянемо п'ять основних які представлені на рисунку 2.2.

В основі системи лежить сукупність вузлів, пул даних. Перш ніж щось відобразити на сайті, необхідно його увести у якості даних.

Наступний рівень – місце для модулів, що таке модуль і які вони існують було розглянуто у попередньому розділі.

На наступному рівні знаходяться блоки та меню. Блоки часто забезпечують висновок з модуля або можуть бути створені для відображення того, що потрібно, а потім розміщення у різних місцях (регіонах) в макеті шаблону (теми). Блоки можуть бути налаштовані на висновок різними способами, а також тільки показ на певних сторінках або тільки для певних користувачів. Меню – це навігатори у Drupal, які визначають контент, що

приходить в кожен певний шлях меню (відносний URL). Меню є основним елементом Drupal, який надає посилання на всі сторінки, створені в Drupal.

Далі ідуть дозволи користувача. Тут настройки налаштовуються, щоб визначити, які типи користувачів можуть робити зміни а які переглядати. Дозвіл визначається для різних ролей, та в свою чергу, користувачам призначаються ці ролі для присвоєння їм певних дозволів.

На верхньому рівні знаходиться тема сайту. В основному вона складається з XHTML та CSS, з деякими змішаними змінними Twig, по цьому згенерований Drupal контент може знаходитись у відповідних місцях. В кожному тему також підключено набір функцій, які можливо використовувати для перевизначення стандартних функцій в модулях, щоб забезпечити повний контроль над тим, як модулі генерують свою розмітку під час виводу.

Цей направлений потік знизу до гори керує роботою Drupal. На діаграмі «А» (див. рис. 2.2) змодельована ситуація, що модуль було завантажено до системи, але не активовано – це робить подальші компоненти недієздатними.

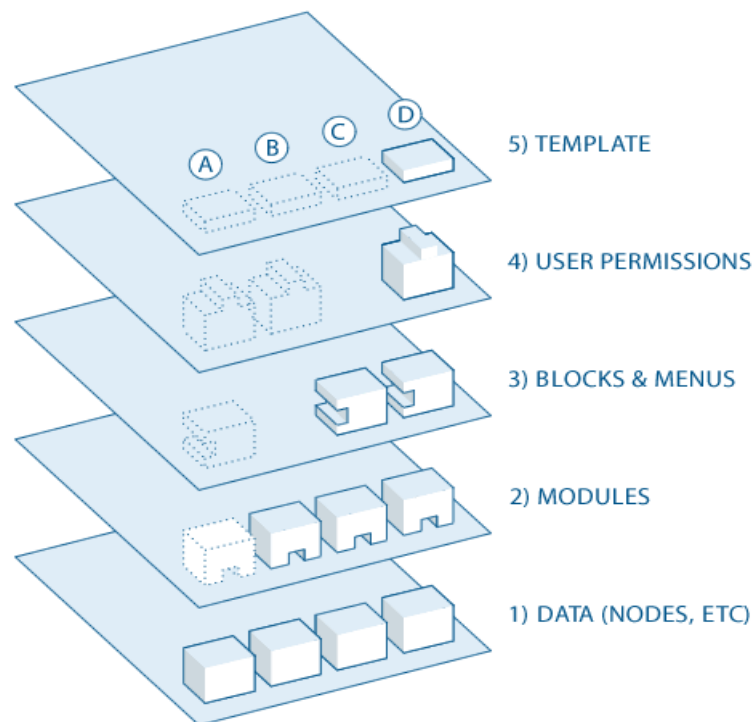


Рисунок 2.2 – Протікання інформації поміж рівнів системи Drupal

Змодельована ситуація «В», коли модуль встановлено та активовано, проте ще не видно, того що необхідно на сайті так як блок не було розміщено (див. рис. 2.2). Інша ситуація коли настройки розширень користувачів конфліктують з тим, що вони бажають, та користувачі сайту не налаштовані на вивід інформації «С» (див. рис. 2.2).

### 2.3 Діаграма компонентів

UML (англ., Unified Modeling Language, UML) – уніфікована мова моделювання, що використовується у парадигмі об'єктно-орієнтованого програмування та є невід'ємною частиною уніфікованого процесу розроблення ПЗ. В основі роботи UML є побудова різних діаграм завдяки яким буде легше уявити систему, та перевести діаграми у програмний код [19].

Фізичне представлення системи описує діаграм компонентів. Завдяки цій діаграмі можливо визначити архітектуру системи яка розробляється, встановлюються залежності між програмними компонентами, в якості яких може виступати вихідний, бінарний і виконуваний код. Основні графічні елементами діаграми компонентів:

- а) компоненти;
- б) інтерфейси;
- в) залежності між компонентами та інтерфейсами.

Розробка діаграми компонентів полягає у:

- а) візуалізації загальної структури вихідного коду програмної системи;
- б) специфікації виконуючого варіанту програмної системи;
- в) забезпечення багаторазового використання окремих фрагментів програмного коду;
- г) уявлення концептуальної і фізичної схем баз даних.

В даній роботі діаграма компонентів буде мати наступних вигляд, як зображено на рисунку 2.3:

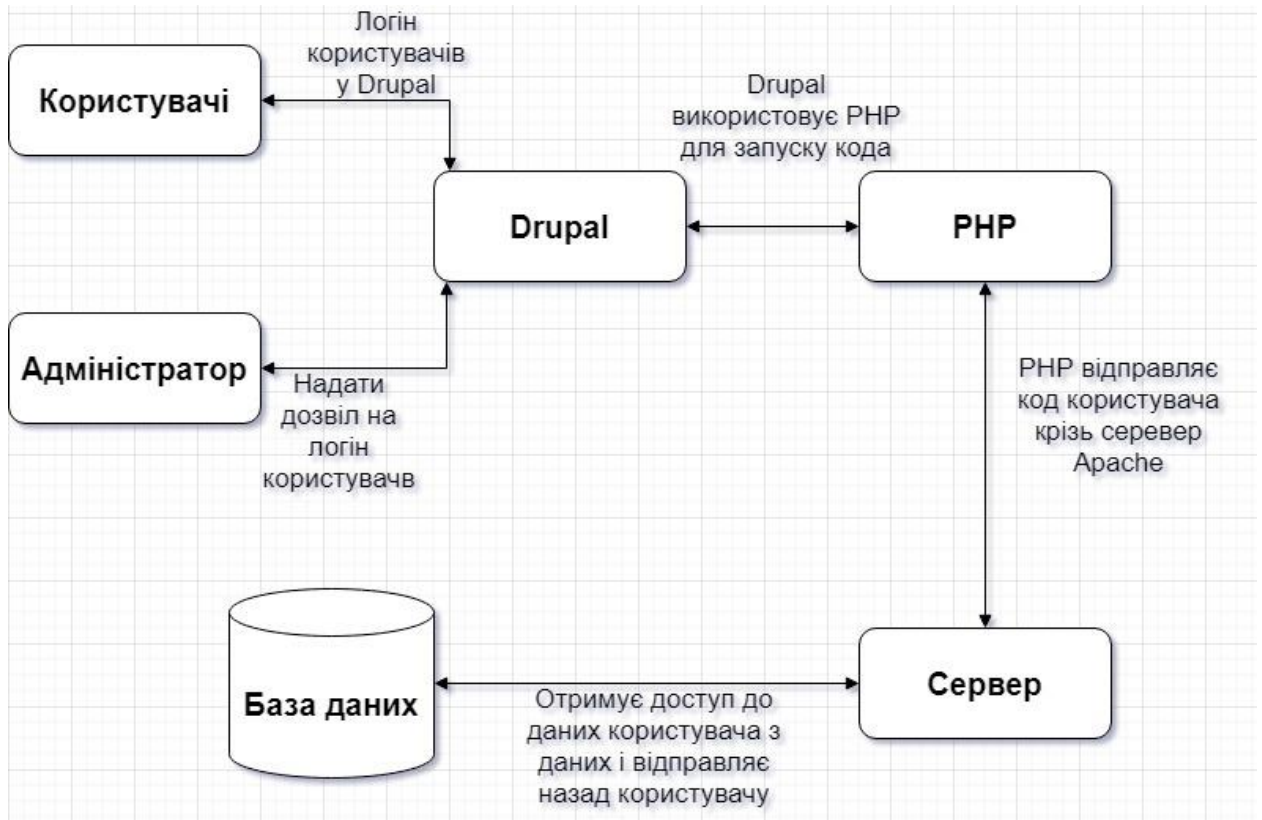


Рисунок 2.3 – Діаграма компонентів

Користувачі – це користувачі спільноти Drupal. Користувач відправляє запит на сервер, використовуючи Drupal CMS, а веб-браузери, пошукові системи діють як клієнти.

Адміністратор – адміністратор може надати дозвіл доступу авторизованим користувачам і зможе заблокувати несанкціонований доступ. Адміністративний обліковий запис має всі привілеї для управління контентом та адмініструванням сайту.

Drupal – Drupal – це безкоштовна система управління контентом з відкритим вихідним кодом (CMS), яка дозволяє організувати, управляти і публікувати ваш контент і розроблена на мові PHP. Drupal CMS може використовуватися для створення великих і складних сайтів. Drupal легко взаємодіє з іншими сайтами і технологіями. Крім того, можна обробляти складні форми і робочі процеси.

PHP – Drupal використовує PHP для роботи з додатком, створеним користувачем. Для отримання даних з бази даних потрібно веб-сервер. Вимоги до пам'яті PHP залежать від модулів, які використовуються на сайті. Drupal 6 вимагає як мінімум 16 МБ, Drupal 7 вимагає 32 МБ, а Drupal 8 вимагає 64 МБ.

Сервер – це веб-сервер, на якому користувач взаємодіє і обробляє запити через HTTP (протокол передачі гіпертексту) і обслуговує файли, які формують веб-сторінки, для веб-користувачів. Зв'язок між користувачем і сервером відбувається по HTTP.

База даних – зберігає інформацію користувача, контент та інші необхідні дані сайту. Вона використовується для зберігання адміністративної інформації для управління сайтом Drupal. Drupal використовує базу даних для отримання даних і дозволяє зберігати, змінювати і оновлювати базу даних.

## **2.4 Діаграма варіантів використання (прецедентів)**

Діаграми прецедентів застосовуються для моделювання вигляду системи з точки зору прецедентів (або варіантів використання). Даний вид діаграм використовується для моделювання динамічних аспектів системи.

Діаграма прецедентів – діаграма, на якій зображено відношення між акторами та прецедентами в системі (див. рис. 2.4).

До діаграм прецедентів звично входить:

- а) прецеденти або варіанти використання;
- б) актори тобто учасники системи;
- в) відношення залежності, узагальнення та асоціації.

UML дає можливість моделювати контекст опираючись на варіанти використання, у яких значна частина уваги приділяється учасникам системи тобто акторам [19].

Важливою частиною є вірне визначення акторів у роботі до них відносяться актори – адміністратори та актори – користувачі це дозволить описати їхні класи взаємодії з системою.

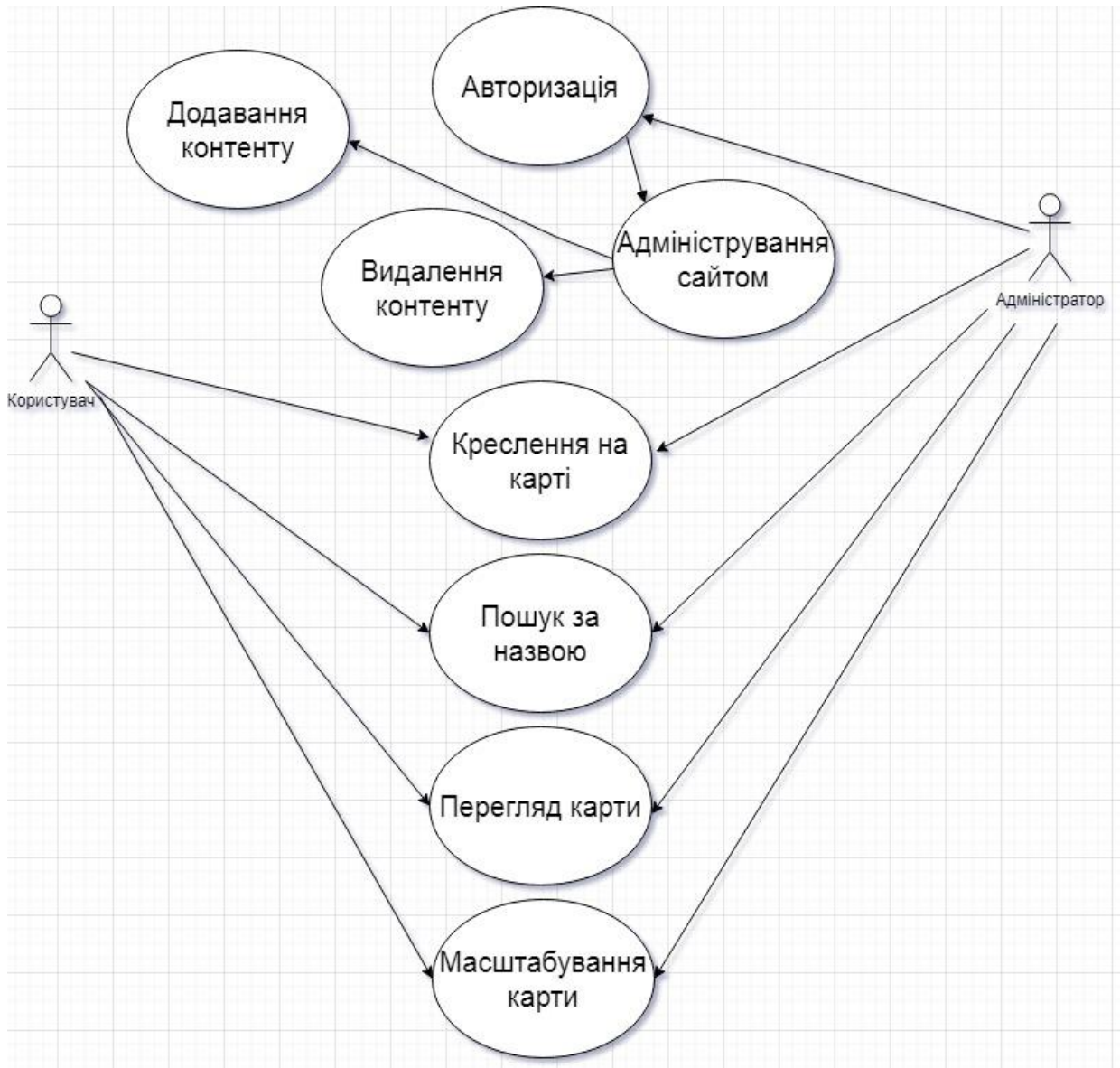


Рисунок 2.4 – Діаграма прецедентів

Саме моделювання розпочнемо з наступних кроків:

- а) ідентифікуємо акторів, які оточують систему: які будуть відноситись до користувачів, та такі які виконують функції підтримки та адміністрування;
- б) організуємо схожих акторів за допомогою відношень;

в) введемо певні стереотипи кожному актору, для полегшення розуміння;

г) розташовуємо акторів системи на діаграмі варіантів використання і визначаємо засоби їх зв'язку стосовно прецедентів.

З діаграми прецедентів видно, що користувачі та адміністратор сайту – це актори, які можуть виконувати дії в залежності від повноважень. У таблицях 2.1 – 2.5 наведені можливі сценарії використання різними користувачами розробленого модуля.

Таблиця 2.1 – Сценарій використання «Перегляд карти»

Варіант використання	Перегляд карти
Актори	Користувачі та адміністратори сайту
Короткий опис	Перегляд інформації, отриманої на карті
Мета	Користувач переглядає інформацію на карті. Система забезпечує доступ до карти та навігацію.

Таблиця 2.2 – Сценарій використання «Креслення на карті»

Варіант використання	Креслення на карті
Актори	Користувачі та адміністратори сайту
Короткий опис	Нанесення на карту точок, маркерів, полігонів
Мета	Користувач може креслити на карті: ставити маркери, додавати кола.

Таблиця 2.3 – Сценарій використання «Пошук за назвою»

Варіант використання	Пошук за назвою
Актори	Користувачі та адміністратори сайту



Короткий опис	Отримання інформації, представленої на карті
Мета	Користувач вводить назву місця, у поле «Адрес» яке він хоче знайти та шукає інформацію на карті. Система забезпечує доступ до карти та зручну навігацію. Стає можливим при встановленні додаткового модуля з геокодуванням.

Таблиця 2.4 – Сценарій використання «Масштабування карти»

Варіант використання	Масштабування карти
Актори	Користувачі та адміністратори сайту
Короткий опис	Масштабування карти, отримання інформації координат
Мета	Користувач бачить масштаб карти, може отримати координати точки.

Таблиця 2.5 – Сценарій використання «Адміністрування сайту»

Варіант використання	Додавання, редагування, видалення контенту
Актори	Адміністратори сайту
Короткий опис	Редагування інформації на карті
Мета	Адміністратор додає нові данні на карту, або видаляє застарілі дані, створює новий контент.

Підсумовуючи матеріал викладений у другому розділі роботи, можна зробити такі висновки, що, Drupal – доволі потужний інструмент веб-

розробника, у якому вже є багато реалізованих модулів для розрішення певних задач, а для необхідності отримання додаткових можливостей існують додаткові модулі також за необхідністю можливо розробити власні модулі і під'єднати їх до Drupal. Проте не слід і забувати про складність в опануванні даної CMS.

Після аналізу діаграм компонентів, структури та протікання інформації стає зрозуміло як працює система Drupal.

Після побудови діаграми компонентів, описавши акторів, прецеденти та визначивши сценарії використання – стає зрозуміло яким повинен бути майбутній модуль з віджетом карт.

## 3 ЕТАПИ РОЗРОБКИ ПЛІГНА ІНТЕГРАЦІЇ OPEN STREET MAP

### 3.1 Базові налаштування модуля Drupal

Перш ніж почати розробку користувацького плагіна у CMS Drupal необхідно створити власний користувацький модуль. Існує декілька способів створення модулів Drupal:

а) використовуючи Drupal Console – це інструмент який полегшує розробку модулів тим що він генерує шаблонний код;

б) власне створюючи усі файли та директорії самому, який ми і будемо використовувати у роботі.

Усі модулі, які завантажені з Drupal.org та розроблені користувачем у Drupal 8 знаходяться у папці /modules/, проте хорошою практикою є створення власної директорії, наприклад – /custom\_modules/ у якій будуть знаходитись власно розроблені модулі, проте наголошую, що це є необов'язковим.

Отже перші дії при розробці модуля:

а) у кореневій папці /custom\_modules/ створюємо папку для нашого модуля: drupal\_osm;

б) створюємо пустий файл drupal\_osm.info.yml, який відповідатиме за опис модуля. Інформація, яка в ньому зазначена буде виводитись на сторінці управління модулями.

YML-файли у Drupal це стандартний формат конфігурацій і налаштувань. Поля і значення у YML-файлах розділяються двокрапкою, новий рядок йде з відступом у два пробілу. Форматування дуже важливо для YML-файлу, якщо у форматуванні є помилка, зайвий пробіл або не вистачає двокрапки, то Drupal може видати помилку або неправильно прочитати YML-файл.

Наступним кроком є написання програмного коду, який складається з обов'язкових та необов'язкових частин:

name: Drupal OSM Module

description: This module implements integration Open Street Map in field  
CMS Drupal

package: Custom

type: module

core: 8.x

Перші три рядки коду використовуються у інтерфейсі адміністрування, тобто дозволяють користувачам увімкнути або вимкнути наш модуль. Розберемо, за що відповідають перелічені у кодї ключі:

а) name та description – відповідають за назву та опис модуля, текст якого буде відображатися у панелі адміністрування модулем, опис необхідно виконувати виключно англійською мовою;

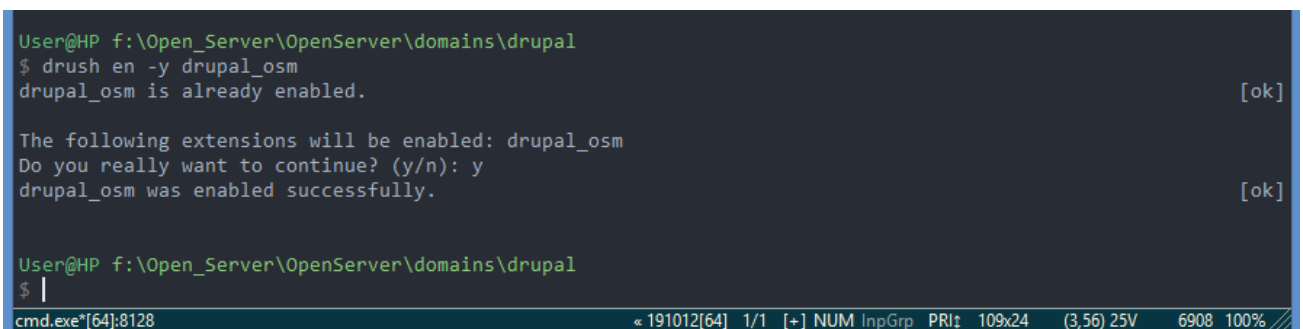
б) package – дозволяє групувати модулі по певним групам;

в) type – вказує на тип розширення модуля;

г) core – показує з якою версією ядра сумісний наш модуль.

Обов'язковими ключами є такі як: name, type та core.

Після збереження файлу, під'єднаємо наш модуль до сайту використовуючи drush та команду: «drush en -y drupal\_osm», як зображено на рисунку 3.1:



```
User@HP f:\Open_Server\OpenServer\domains\drupal
$ drush en -y drupal_osm
drupal_osm is already enabled. [ok]

The following extensions will be enabled: drupal_osm
Do you really want to continue? (y/n): y
drupal_osm was enabled successfully. [ok]

User@HP f:\Open_Server\OpenServer\domains\drupal
$ |
```

Рисунок 3.1 – Під'єднання модуля з використанням drush

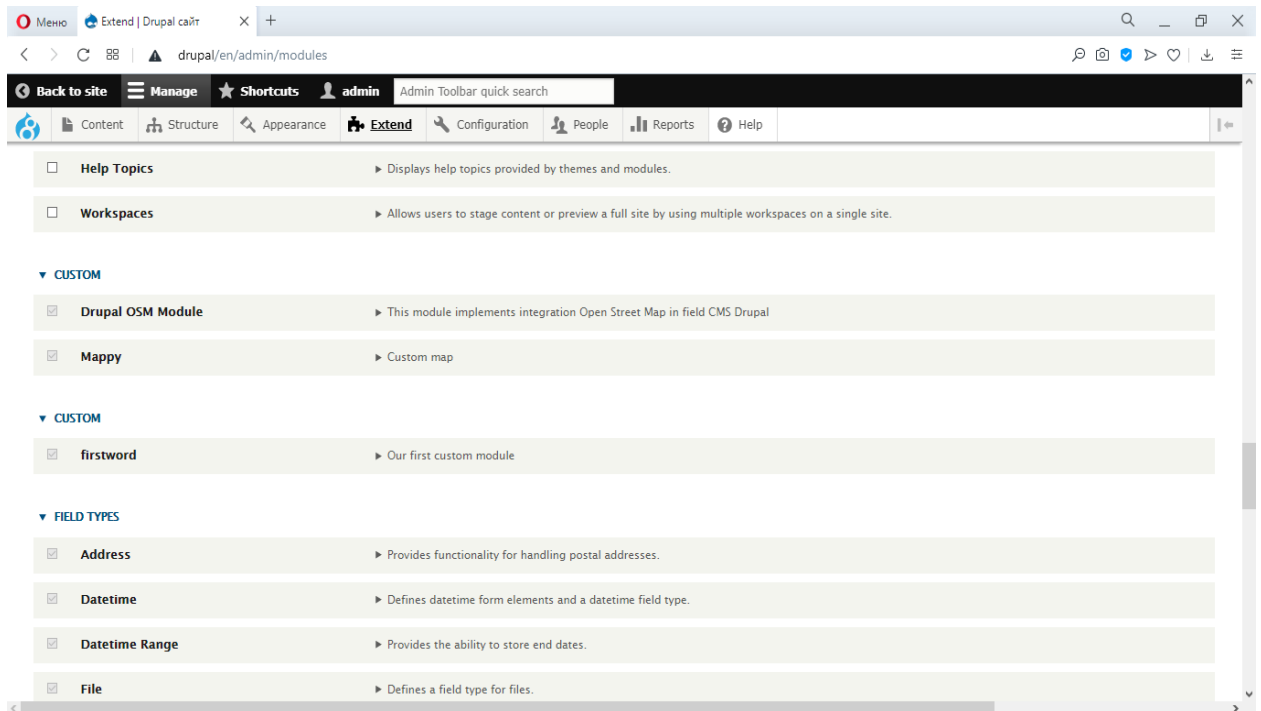


Рисунок 3.2 – Відображення модуля у панелі адміністрування Drupal

Отже коли модуль під'єднано переходимо до адміністративної панелі нашого сайту і перевіряємо успішність виконаної вище дії. На рисунку 3.2 зображено успішне під'єднання нашого модуля у категорії custom.

### 3.2 Реалізація хуків

Хуки дозволяють модулям змінювати та розширювати поведінку ядра Drupal або іншого модуля. Вони є одним із різноманітних способів взаємодії компонентів коду в Drupal між собою. Використовуючи хуки, при розробці модулів можна змінювати принцип роботи ядра або іншого модуля – без зміни існуючого коду [16].

Хуків у Drupal 8 багато, проте якщо порівняти з Drupal 7 менше. Те що у Drupal 7 можливо було реалізувати завдяки хуку 8 версія потребує ще додаткового відповідного плагіну.

У даній роботі для розробки модуля з плагіном карт були використані такі хуки:

а) `hook_theme ()` це один з фундаментальних хуків у Drupal 8. Все що знаходиться на сторінках, так чи інакше проходить через даний хук. Всі шаблони, які створюються чи вже існують у темі – це робота `hook_theme`. `hook_theme` працює як в темах так і в модулях. Він повертає масив, ключі у першому рівні якого стають "хуками теми" (theme hooks)

б) `variables`: використовується спільно з `#theme`. Цей theme hook використовується для власного шаблону під певне завдання. Це масив, який може бути порожнім, або містити всі можливі змінні, які необхідні шаблону з їх значеннями за замовчуванням. При необхідності зробити файл шаблону, який буде приймати дані, тут вказуються всі дані які необхідні з параметрами за замовчуванням, якщо параметри невідомі використовуємо значення `NULL`;

в) `template`: назва файлу шаблону, який буде використовуватися для виведення даних тобто рендерінгу. Всі дані будуть передані в даний шаблон і його результат вже буде використаний для виведення. Розширення тут створюється автоматично та додано двигуном який використовується для рендера автоматично. Назва не повинна містити нижніх підкреслень, тільки тире [18]. Назву theme hook встановлюємо як значення з заміною нижніх підкреслень на тире.

Використовуючи хуки програмний код має вигляд:

Файл `drupal_osm.module`:

```
function drupal_osm_theme() {
  return [
    'drupal_osm' => [
      'variables' => [
        'description' => null,
      ],
    ],
  ],
}
```

```

    'template' => 'drupal_osm',
    'render element' => 'children',
  ],
];
}

```

### 3.3 Налаштування відображення матеріалу

Для інтеграція OpenStreetMap до CMS Drupal будемо використовувати плагіни. Плагіни – це невеликі частини функціональності, яку можна замінити. Block API в Drupal 8 була змінена таким чином, що зараз блоки є плагінами (plugins) і сутностями (entities) одночасно. Блоки в Drupal 8 поєднують в собі дві різні API структури: Block Plugin API, яке є автономним повторно використовуваних API, і Block Entity API, яке представляє собою конкретний варіант використання блоку і управління його видимості.

Ця зміна дозволяє використовувати той же самий блок в різних регіонах, що було неможливо без додаткових модулів до Drupal 8.

Наступним кроком у створенні є створення власного блоку. На початку файлу йде namespace, це визначення де повинен бути файл плагіна блоку, щоб Drupal міг автоматично під'єднати його. Також ми підключаємо класи з інших файлів, використовуючи use:

```

namespace Drupal\drupal_osm\Plugin\Block;

use Drupal\Core\Block\BlockBase;
use Drupal\Core\Form\FormStateInterface;
use Drupal\Core\Plugin\ContainerFactoryPluginInterface;
use Drupal\node\Entity\Node;
use Symfony\Component\DependencyInjection\ContainerInterface;
use Drupal\Core\Entity\EntityTypeManager;

```

Далі в коментарі йде анотація, тут ми вказуємо `@Block`, що це буде плагін блоку. `id`, `admin_label` автоматично підхоплюється Drupal, тому `id` повинен бути унікальним. `@Translation` допомагає перевести потім `label` блоку через адміністрування Drupal:

```
/**
 * Provides a 'CustomOsmBlock' block.
 *
 * @Block(
 *   id = "custom_osm_block",
 *   admin_label = @Translation("Custom osm block"),
 * )
 */
```

Додаємо форму з налаштуваннями:

```
/**
 * {@inheritdoc}
 */
public function defaultConfiguration() {
  return [
    'description' => $this->t('My custom Open Street Map'),
  ] + parent::defaultConfiguration();
}
```

За допомогою даного методу оголошуємо форму з настройками для даного блоку використовуючи Form API:

```
/**
 * {@inheritdoc}
 */
public function blockForm($form, FormStateInterface $form_state) {
  $form['description'] = [
    '#type' => 'textfield',
```



```

        '#title' => $this->t('Description'),
        '#description' => "",
        '#default_value' => $this->configuration['description'],
        '#maxlength' => 64,
        '#size' => 64,
        '#weight' => '0',
    ];

    return $form;
}

```

Цей метод зберігає значення поля `description` в якості нового пункту конфігурації блоку (з ключем того ж імені для відповідності):

```

/**
 * {@inheritdoc}
 */
public function blockSubmit($form, FormStateInterface $form_state) {
    $this->configuration['description'] = $form_state->getValue('description');
}

```

Цей метод повертає перше значення поля «Geofield»:

```

private function getGeofieldValue(){
    $result = null;
    $node = \Drupal::routeMatch()->getParameter('node');
    if ($node){
        $result = $node->get('field_geofield_osm')->getValue()[0];
    }
    return $result;
}

```

Метод `build ()` - найбільш важливий з них, так як він повертає відображається масив, який буде виводити блок. Отримаємо перше значення «Geofield» поточного вузла та його заголовок, які будуть використовуватися

Javascript. Використовуємо масив візуалізації у методі побудови, який визначатиме опис та drupalSettings для Javascript:

```
/**
 * {@inheritdoc}
 */
public function build() {
    $build = [];
    // Get the current node object
    $node = \Drupal::routeMatch()->getParameter('node');
    if ($node instanceof Node) {
        $build = [
            '#theme' => 'my_maps',
            '#description' => $this->configuration['description'],
            '#attached' => array(
                'library' => array(
                    'drupal_osm/custom_map',
                ),
                'drupalSettings' => array(
                    // Return the first Geofield value
                    // 'geofield' => $node->get('field_geofield_test')->getValue()[0],
                    'geofield' => $this->getGeofieldValue(),
                    'title' => $node->getTitle(),
                ),
            ),
        ];
    }
    return $build;
}
```

Наступним кроком є створити div з ідентифікатором «map» у нашому файлі шаблону, також потрібно визначити висоту та ширину div map.

У модулі, в папці шаблонів, створюємо шаблон гілки. Ім'я файлу має збігатися з тим, що ввели в `hook_theme`.

Файл `templates/drupal_osm.html.twig`:

```
<p>{{ description }}</p>
```

```
<div id="map" style="width: 500px;height: 500px;"></div>
```

Після створення `div` розмістимо у ньому карту створену завдяки власного файлу `js` та бібліотеки `Leaflet`.

Файл `js/custom_map.js`:

Ми обертаємо код `jQuery` в таку конструкцію, тому що `jQuery` в `Drupal` запускається у режимі `.noConflict()`. Це потрібно для того щоб використовувати знак долара `$`, і це не конфліктувало з іншими `javascript` фреймворками `Prototype`, `MoوTools`. Проте в цю конструкцію необхідно обертати весь `jQuery`-код:

```
(Function ($) {
```

```
  }) (jQuery);
```

Визначаємо функцію `init`, яка буде отримувати об'єкт «`geofield`» та заголовок «`title`» вузла з `drupalSettings`, які будуть передані блоком `CustomOsmBlock` та змінну «`point`» яка зберігатиме у собі значення широти та довготи:

```
function init (geofield, title) {
```

```
  var point = {lat: geofield.lat, lng: geofield.lon};
```

Створюємо карту використовуючи завантажену бібліотеку `Leaflet`, з центром у точці з координатами `point` та певним масштабом:

```
var map = L.map('map', {
  zoomControl: true,
  scrollWheelZoom: true,
  center: [point],
```

```
zoom: 13
```

```
});
```

Далі необхідно завантажити плитки з офіційного сайту OpenStreetMap [12], проте згідно їхньої політики використання плиток і їх рекомендацій обираємо альтернативного постачальника плиток Thunderforest [13] та додаємо їх до карти:

```
var baseLayer = L.tileLayer('https://tile.thunderforest.com/cycle/{z}/{x}/{y}.png?
apikey=666fc9550f39435cbf7f969ee439ff1e', {
attribution: '&copy; <a href="https://www.openstreetmap.org/copyright"
>OpenStreetMap</a> contributors, &copy; <a href=https://cartodb.com/attributions
>CartoDB</a>'
});
```

```
map.addLayer(baseLayer);
```

Додаємо маркер у зазначену точку, виводимо значення поля «title» у цій точці та додаємо на карту:

```
var markerOptions = {
  title: title,
  clickable: true,
  draggable: true
}
```

```
var marker = L.marker([point], markerOptions)
marker.addTo(map);
```

Далі для більш інформативності на карті створимо позначку масштабування та додаймо її до карти:

```
var scale = L.control.scale();
scale.addTo(map);
```

Перед підключенням бібліотек необхідно завантажити їх з офіційного сайту Leaflet [14], та розмістити їх у директоріях:

a) js-файли /drupal\_osm/js/leaflet.js

б) css-файли /drupal\_osm/css/leaflet.css

Створюємо новий файл у корені модуля drupal\_osm.libraries.yml. Далі вказуємо нові бібліотеки та шляхи до наших файлів js та css. У Drupal 8 jQuery більше не завантажується на кожну сторінку, тому його потрібно явно включити як залежність. Важливо дотримуватися відступу, щоб він був в два пробіли:

Файл drupal\_osm/drupal\_osm.libraries.yml:

leaflet:

css:

theme:

css/leaflet.css: { }

js:

js/leaflet.js: { }

custom\_map:

version: VERSION

js:

js/custom\_map.js: { }

dependencies:

- drupal\_osm/leaflet
- core/jquery
- core/drupal

Особливості використання:

а) встановлюємо додатковий модуль Geofield за допомогою Composer та команди: «\$ composer require 'drupal/geofield'» та під'єднаймо його до нашого сайту: «\$ drush en geofield». Цей модуль дозволить зберігати географічні данні тобто наші координати;

б) у панелі адмініструванням сайту переходимо до вкладки «Структури» обираємо «Тип контенту», створюємо власний контент Test map та це необов'язково можна редагувати будь який існуючий полями як зображено на рисунку 3.3:

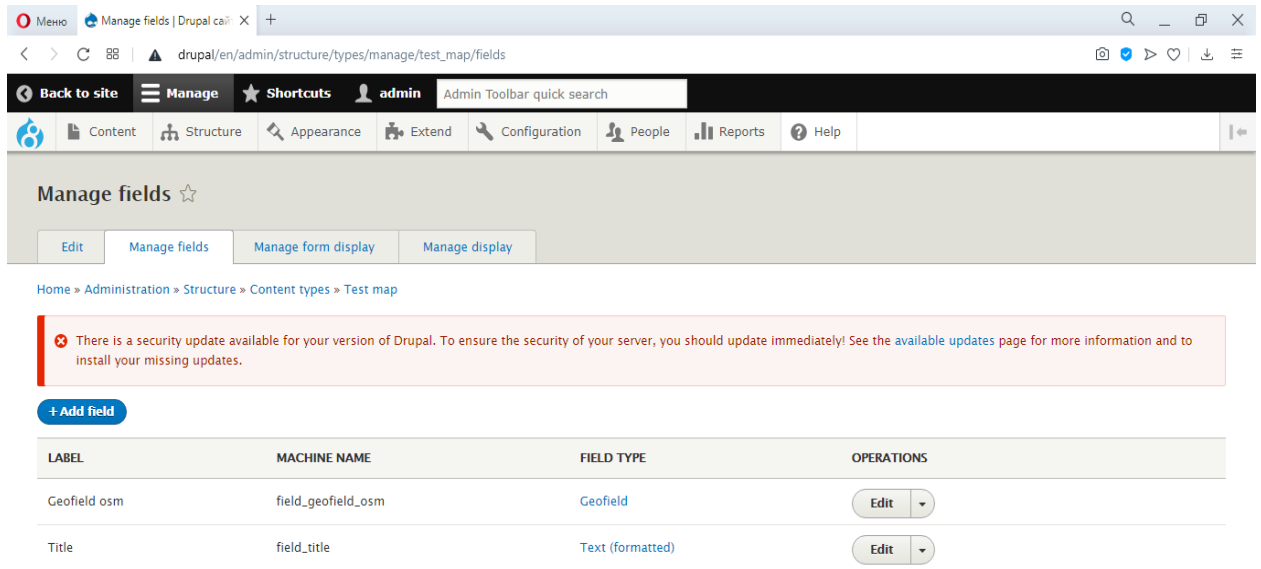


Рисунок 3.3 – Управління полями матеріалу

Наступним кроком буде налаштування плагіну, а саме:

- обрати з яким типом контенту він буде використовуватись;
- обрати регіон тобто місце на сторінці сайту де необхідно розмістити плагін з інтегрованою картою.

Докладніше зображено на рисунку 3.4:

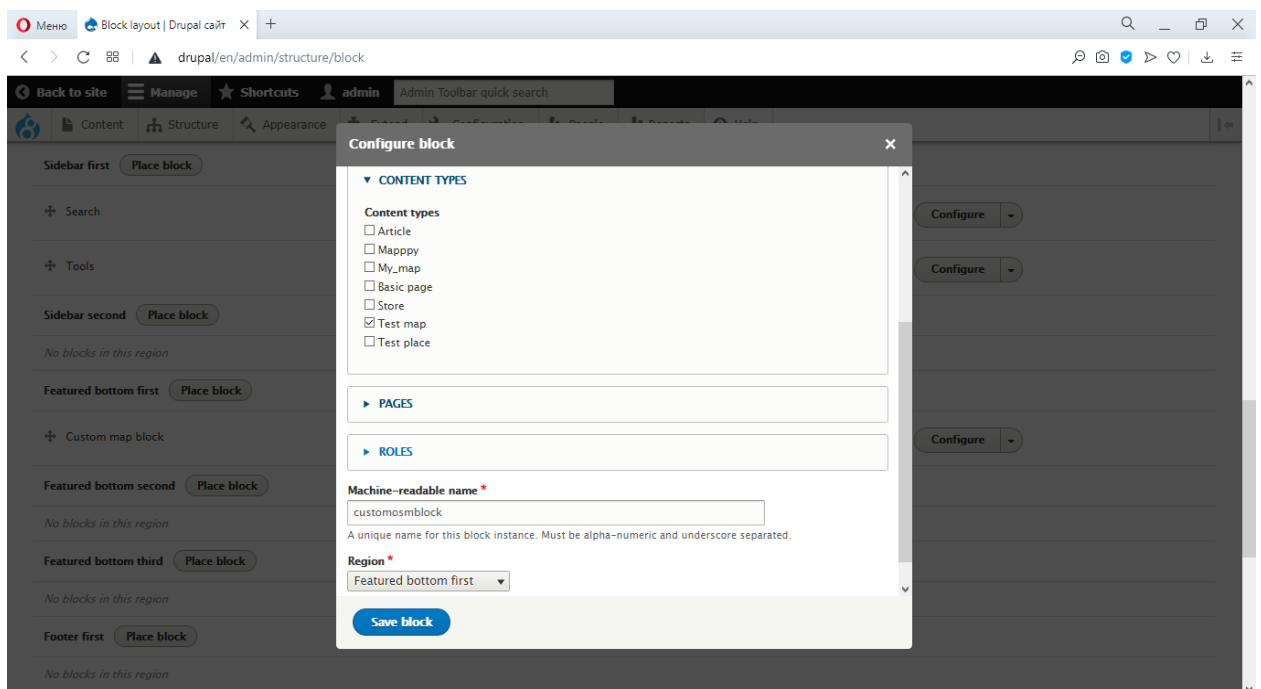


Рисунок 3.4 – Налаштування відображення плагіну

Далі, для створення нового матеріалу, який буде відображено на сайті, або при редагуванні його, необхідно ввести дані у всі запропоновані поля [17].

А саме:

а) в полі «Title» написати заголовок інформаційного матеріалу, який буде використовуватись для підпису маркера на карті;

б) в полі «GEOFIELD OSM» вписати широту та довготу яка буде використовуватись для позиціювання карти та маркера.

Все вище наведене зображено на рисунку 3.5:

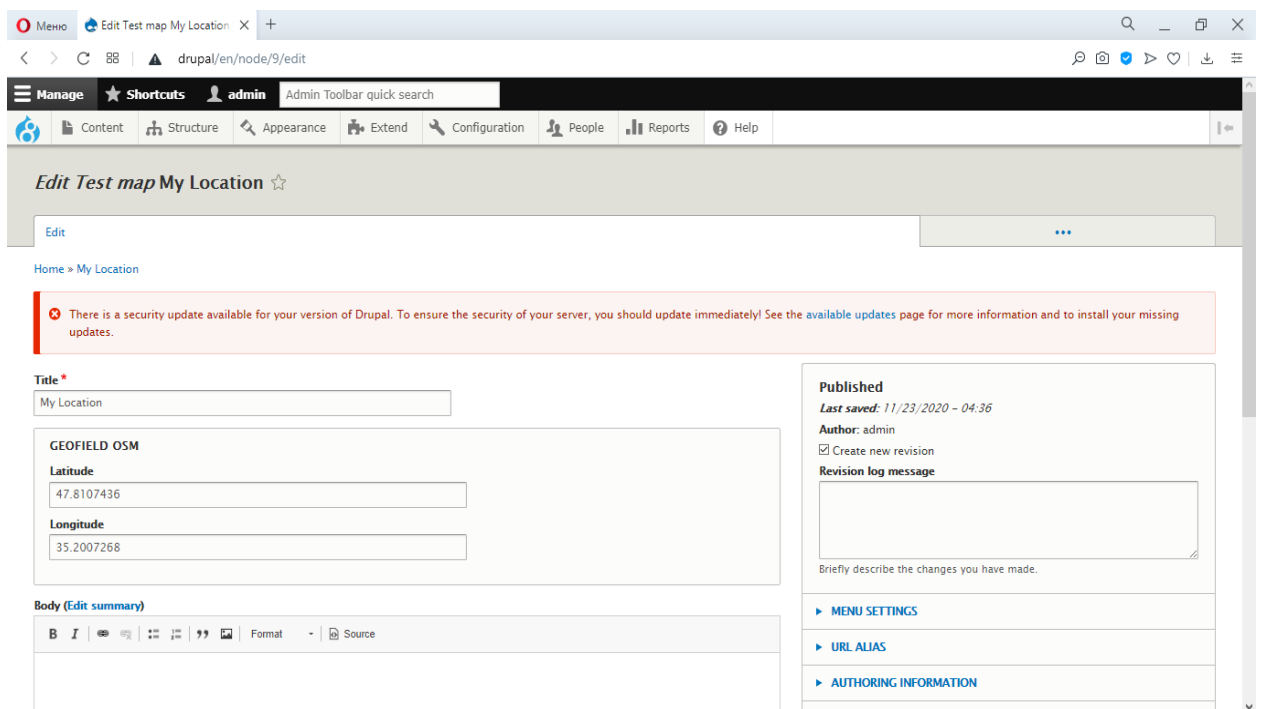


Рисунок 3.5 – Редагування полів матеріалу

По завершенню всіх налаштувань та внесення даних у всі поля отримаємо: назву, поле з координатами та блок з картою у обраному регіоні який за необхідністю можна змінити це зображено на рисунках 3.6 та 3.7. На правах адміністратора ми можемо цю карту переглядати, редагувати, а також видаляти об'єкти на карті або додавати їх.

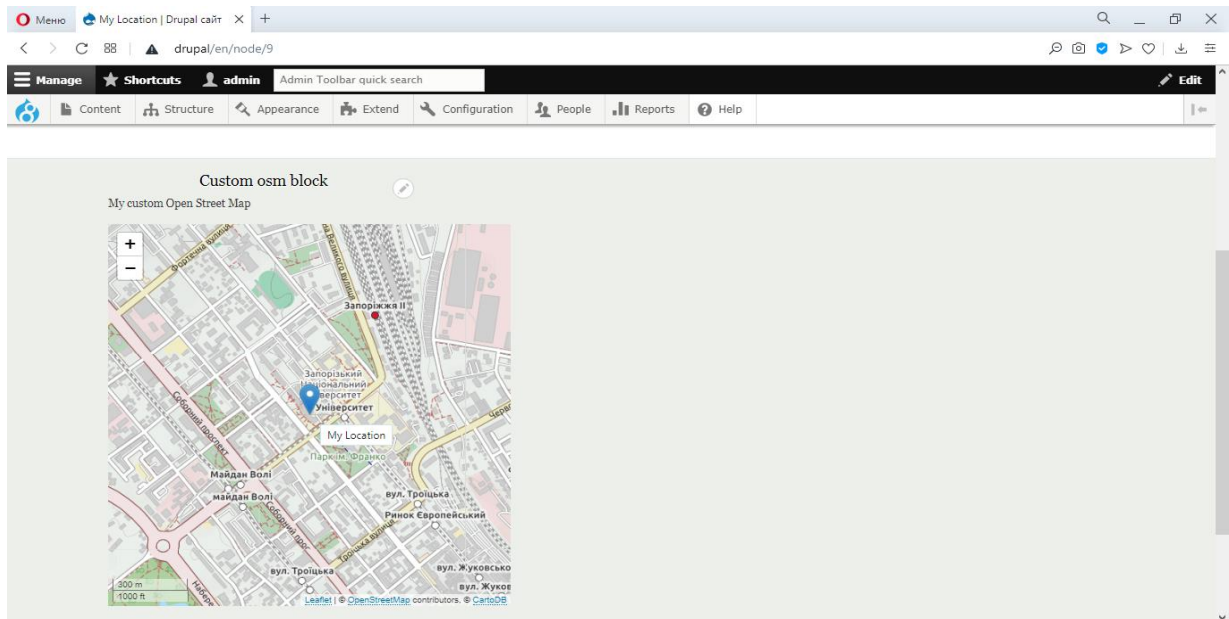


Рисунок 3.6 – Створення карти у регіоні Featured bottom first

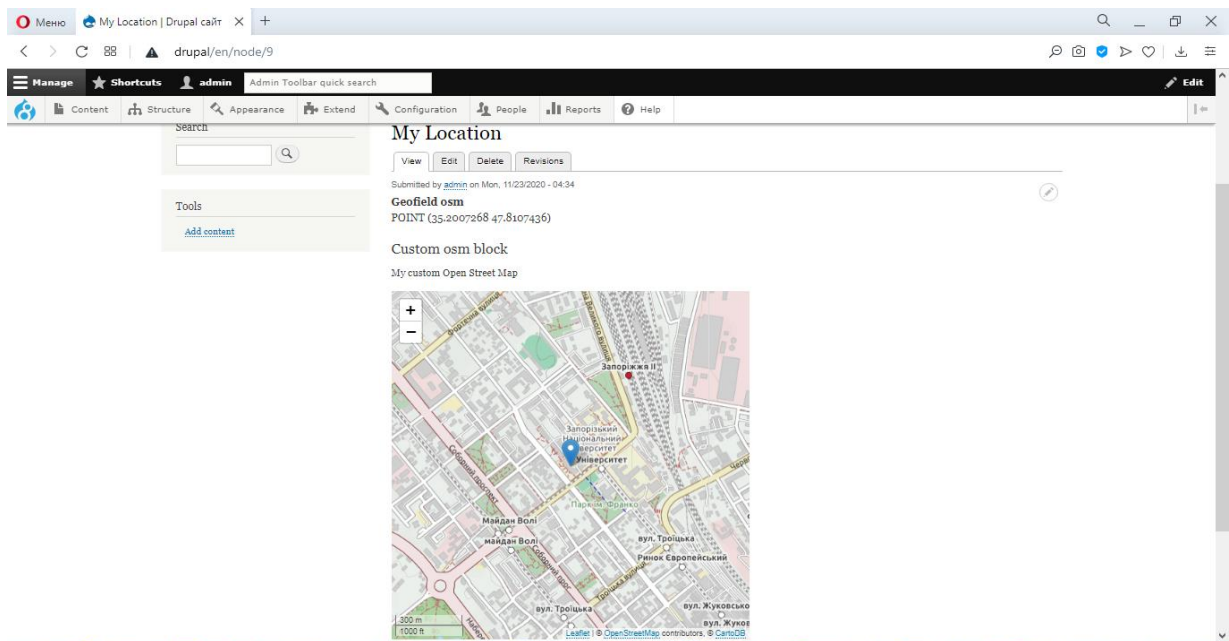


Рисунок 3.7 – Створення карти у регіоні Content

Додаткові налаштування при необхідності використання пошуку за назвою:

а) з порталу Drupal.org за допомоги Composer та drush встановлюємо та під'єднаємо модуль «Address»;

б) обираємо у налаштуваннях геокодування з поля адрес за допомоги плагіна від nominatium.



## ВИСНОВКИ

В даній кваліфікаційній роботі було здійснено аналіз CMS та визначені три найпоширеніші, наведено їх позитивні та негативні якості. Здійснено описання, порівняння та проаналізовані сфери використання найпоширеніших CMS, таких як: WordPress, Joomla! та Drupal. В процесі аналізу функціональності наведених вище систем досліджено, що Drupal, як найбільше підходить для створення модуля з плагіном відображення карт OpenStreetMap.

Виявлено, що функціональність Drupal – велика кількість модулів, тем, тощо. До переваг треба віднести:

- а) безкоштовна CMS з відкритим вихідним кодом;
- б) високий рівень безпеки;
- в) доволі високий рівень функціональності, що дозволяє поєднати майбутній модуль з існуючими плагінами.

Проаналізовано роботу CMS Drupal, загальний принцип роботи якої полягає у модульності – тобто додавання певних нових функціональних можливостей за допомогою встановлення нових модулів.

Проаналізувавши діаграми компонентів, структури та протікання інформації стає більш зрозуміліше як працює система Drupal

Також можна зробити висновки, що діаграми прецедентів застосовуються для моделювання вигляду системи з точки зору варіантів використання. Даний вид діаграм використовується для моделювання динамічних аспектів системи.

У результаті виконання дипломної роботи були вирішені такі задачі:

- розглянуто загальні характеристики CMS;
- проведено порівняльний аналіз CMS, обрані найпоширеніші;
- розглянуто основні переваги і недоліки найпоширеніших CMS їх характеристики;

- розглянуто етапи розробки модулів у CMS Drupal;
- проаналізовано основні програмні засоби для створення модуля карт у Drupal;
- спроектовано модуль для інтеграції карт OpenStreetMap до CMS Drupal;
- розроблено модуль для інтеграції карт OpenStreetMap до CMS Drupal.

Модуль з інтегрованою картою OpenStreetMap можливо використовувати як аналог сервісам від Google та Yandex на власному сайті для сторінки контакти.

Можливими напрямками розвитку може бути:

- а) вдосконалювання інтерфейсу;
- б) розширення функціональних можливостей.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Горнаков С. Г., Осваиваем популярные системы управления сайтом (CMS). Москва: Пресс, 2009. 336 с.
2. Басюк Т.М. Принципи побудови системи аналізу та просування інтернет-ресурсів Вісник Нац. ун-ту “Львівська політехніка” “Комп’ютерні науки та інформаційні технології”, 2012. № 784, С.43–48.
3. Usage statistics of content management systems. *W3Techs*. URL: [https://w3techs.com/technologies/overview/content\\_management](https://w3techs.com/technologies/overview/content_management) (дата звернення 01.11.2020)
4. Tech Reports. *WhatCms*. URL: [https://whatcms.org/Tech\\_Reports](https://whatcms.org/Tech_Reports) (дата звернення)
5. Офіційний сайт WordPress. *Веб розробка*. URL: <https://wordpress.org/> (дата звернення 01.11.2020)
6. Top Content Management System Technologies. *SimilarTech*. URL: <https://www.similartech.com/categories/content-management-system/> (дата звернення 01.11.2020)
7. Usage Statistics. *Built With*. URL: <https://trends.builtwith.com/cms/> (дата звернення 01.11.2020)
8. Офіційний сайт Joomla!. *Веб розробка*. URL: <https://www.joomla.org/> (дата звернення 02.11.2020)
9. Офіційний сайт Drupal. *Веб розробка*. URL: <https://www.drupal.org/> (дата звернення 02.11.2020)
10. Офіційний сайт Open Server. *Веб розробка*. URL: <https://ospanel.io/> (дата звернення 02.11.2020)
11. Джон К. Вандюк, Мэтт Вестгейт. CMS Drupal. Руководство по разработке системы управления сайтом. Вильямс, 2009. 576 с.
12. Офіційний сайт OpenStreetMap. *Веб карта*. URL: <https://www.openstreetmap.org/> (дата звернення 19.11.2020)

13. Tile Layer URLs. Thunderforest. URL: <https://www.thunderforest.com/>  
(дата звернення 20.11.2020)
14. Download Leaflet. Leaflet. URL: <https://leafletjs.com/download.html>  
(дата звернення 22.11.2020)
15. Drupal – википедія. Електронний ресурс. URL: <http://ru.wikipedia.org/wiki/Drupal> (дата звернення 05.11.2020)
16. Справочник API. Drupal API. Електронний ресурс. URL: <http://api.drupal.ru/> (дата звернення 20.11.2020)
17. Разработка модулей для CMS Drupal 7. Практическое руководство. Русаков Е.М. 2012 <https://drupal.ru/taxonomy/term/12520> (дата звернення 22.11.2020)
18. Плєскач В. Л. Інформаційні системи і технології на підприємствах. Засоби створення Web-сайтів. URL: <http://westu-dents.com.ua/glavy/27290-zasobi-stvorennya-Web-saytv.html> (дата звернення 22.11.2020)
19. Олійник А.О., Колпакова Т.О., Льовкін В.М., Методичні вказівки до виконання лабораторних робіт з дисципліни “Архітектура та проектування програмного забезпечення” для студентів напряму “ Програмна інженерія”, Запоріжжя: ЗНТУ, 2012. – 72 с.