

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

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА
на тему: **«РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ
ДЛЯ ВИКЛИКУ ТАКСІ З ВИКОРИСТАННЯМ
CMS DRUPAL»**

Виконав: студент 2 курсу, групи 8.1218-з
Спеціальності

121 інженерія програмного забезпечення

(шифр і назва спеціальності)

Дараєв Антон Юрійович

(ініціали та прізвище)

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

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент декан математичного факультету,
професор, д.т.н., Гоменюк С. І.

(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
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 СТУДЕНТУ

Дараєву Антону Юрійовичу
(прізвище, ім'я та по-батькові)

1. Тема роботи «Розробка інформаційної системи для виклику таксі з використанням CMS Drupal»

керівник роботи Лісняк Андрій Олександрович, к.ф.-м.н., доцент
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 29 » травня 2019 року № 812-с

2. Строк подання студентом роботи 21.12.2019 р.

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Постановка задачі.
2. Огляд технологічних підходів для розробки інформаційної системи.
3. Етапи розробки інформаційної системи для .виклику таксі.

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

7. Дата видачі завдання _____ 29.05.2019 _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	15.09.2019	
2.	Збір вихідних даних.	19.09.2019	
3.	Обробка методичних та теоретичних джерел.	28.09.2019	
4.	Розробка першого та другого розділу.	22.10.2019	
5.	Розробка третього розділу.	19.11.2019	
6.	Оформлення та нормоконтроль кваліфікаційної роботи.	21.12.2019	
7.	Підготовка доповіді та презентації.	28.12.2019	
8.	Захист кваліфікаційної роботи.	10.01.2020	

Студент _____
(підпис)А. Ю. Дарасв _____
(ініціали та прізвище)Керівник роботи _____
(підпис)А. О. Лісняк _____
(ініціали та прізвище)**Нормоконтроль пройдено**Нормоконтролер _____
(підпис)О. В. Кудін _____
(ініціали та прізвище)

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат	4
Summary	5
Вступ.....	7
1 CMS. Загальний огляд та характеристики.....	9
1.1 Загальна характеристика CMS.....	9
1.2 Індивідуальні CMS.....	11
1.3 Основні функції CMS	12
1.4 Різновиди існуючих CMS.....	14
1.5 Загальний огляд CMS WordPress	16
1.6 Загальні характеристики CMS Joomla	19
1.7 Огляд та характеристики CMS Modx.....	20
1.8 Загальний огляд CMS Drupal.....	22
Висновки до розділу 1	24
2 Проектування модуля відображення матеріалів на картах.....	25
2.1 Діаграма розгортання. Діаграма компонентів	25
2.2 Структура CMS Drupal	27
2.3 Діаграма варіантів використання (прецедентів).....	30
Висновки до розділу 2	31
3 Етапи розробки модуля Google карт	32
3.1 Базові налаштування модуля Drupal.....	32
3.2 Схема конфігурації, метадані	33
3.3 Налаштування відображення матеріалу	37
Висновки	47
Перелік посилань.....	48

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка інформаційної системи для виклику таксі з використанням CMS Drupal»: 49 с., 13 рис., 18 джерел.

ВЕБ-СЕРВЕР, ВЕБ-СТОРІНКА, КАТЕГОРІЇ, МОДУЛЬ, ПОСИЛАННЯ, РУБРИКА, САЙТ, СТРУКТУРА, ШАБЛОН.

Об'єкт дослідження – етапи створення інформаційної системи для CMS Drupal.

Предмет дослідження – інформаційна система для виклику таксі.

Мета роботи – розробка інформаційної системи для виклику таксі з використанням CMS Drupal.

Методи дослідження – аналітичний, пошуковий, порівняльний.

У кваліфікаційній роботі розглянуто особливості розробки інформаційно системи з використанням CMS Drupal. Розглянуто основні поняття – модуль, сайт, веб-вузол, веб-сервер. Розглянуто існуючі інформаційні системи для виклику таксі як у системі Drupal, так і в інших системах керування контентом. Розроблений модуль можна використовувати як ресурс для пошуку необхідної інформації на Google картах, прокладання маршруту та визначення його тривалості.

SUMMARY

Bachelor's qualifying paper «Development of an Information System for Taxi Calling using CMS Drupal»: 49 pages, 13 figures, 18 references.

WEB PAGE, WEB-SERVER, CATEGORY, LINKS, RUBRIC, SITE, STRUCTURE, HOSTING, TEMPLATE.

Object of research – stages of creating an information system for CMS Drupal.

Subject of research – information system for calling taxi.

Aim of the study is to develop an information system for calling taxi for CMS Drupal.

Methods of research are analytical, comparative, analysis and searching of information.

This paper describes the features of developing an information system for calling taxi using CMS Drupal. The basic concepts – an information system, a site, a web site, a web server is considered. The modules of maps exist both in the Drupal system and in other CMS. The developed module we can use as a resource for finding the necessary information on Google Maps laying the route and determining its duration.

ВСТУП

Сучасний світ неможливо уявити без інформаційних технологій та Інтернету, який охоплює всі сфери нашого життя. В Інтернет переноситься пошта, телефонія, бізнес.

Веб-сфера стрімко розвивається в наші дні. На заміну сірим, «академічним» сайтам прийшла ціла плеяда яскравих сайтів з найрізноманітнішими призначеннями: візитки, торгіві, інформаційні, розважальні, соціальні мережі, особисті та корпоративні блоги.

Однією з проблем перших сайтів було наповнення матеріалом – для виконання такого, звичного сьогодні, завдання були необхідні спеціалісти-розробники, котрі знали архітектуру сайту та могли все виконати правильно. Звісно, такий підхід до заповнення сайтів вимагав додаткових витрат, тож не всім це подобалось, чим і скористалися розробники створивши CMS (Content Management System) – системи, що дозволяли редагувати старий та додавати новий контент на сайт. Однак, далеко не завжди базові можливості CMS задовольняють вимоги користувачів і замість того, щоб переписувати все з нуля, або вносити поправки безпосередньо в код ядра була створена система створення модулів, що дозволило розробникам та досвідченим користувачам змінювати функціонал сайту/ CMS без ризику зруйнувати все.

Drupal – CMS (система управління контентом) з відкритим вихідним кодом, на базі якої можна створювати сучасні сайти, будь-то особистий блог або інтернет-магазин, Drupal – інструмент, який дозволить це втілити в життя.

У порівнянні з WordPress, Drupal більш потужний і гнучкий, тому він буде мати більше налаштувань і можливостей. Разом з тим, настроювання сайту на Drupal буде більш складним, але у нього є багато можливостей і пропозицій:

а) велика кількість функцій: порівнюючи з іншими сучасними системами управління контенту, Drupal більш функціональний;

б) більше можливостей для налаштування: можливості налаштування Drupal з кількістю плагінів і варіантів конфігурації дозволяє розробникам реалізувати специфічні вимоги і уявлення про сайт. В файли ядра Drupal також можна легко вносити поправки, що теж є чудовою нагодою в процесі розробки;

в) безпека: Drupal надає високий рівень безпеки і здатність генерації докладних звітів про безпеку.

Модулі в Drupal є невід'ємною частиною. Вони слугують інструментом для розширення функціональності сайту та додають йому нових властивостей. Основною перевагою використання модулів в Drupal є те, що ви маєте можливість побудувати сайт на ваш розсуд. Модулі надають можливості для пошуку, персоналізації вмісту, а також є корисними для сайтів електронної комерції. Не зважаючи на те, чи потрібно групувати будь-які запити, чи створювати інтерактивні меню — модулі Drupal дозволяють це зробити.

Є три типи модулів: ті, що входять до складу ядра (модулі ядра), ті, які можна завантажити з Drupal.org, але не є частиною ядра (контрибні модулі), модулі, написані спеціально для певного проекту (кастомні модулі).

У даній роботі поставлена задача розробити інформаційну систему виклику таксі з використанням CMS Drupal. Отримана у результаті інформаційна система разом зі створеним модулем відображення матеріалів на картах допоможе службам таксі створювати мітки у місті за допомогою Google карти з можливістю вказувати свої параметри для планування маршруту.

1 CMS. ЗАГАЛЬНИЙ ОГЛЯД ТА ХАРАКТЕРИСТИКИ

1.1 Загальна характеристика CMS

CMS – це система управління контентом, набір скриптів для створення, редагування і управління контентом сайту. На професійному жаргоні CMS також називають «движок». Прикладами CMS є WordPress, Joomla, PrestaShop, Drupal.

Якщо раніше більшість сайтів були статичними; і вимагали внесення правок в їх вміст вручну, то зараз динаміка розвитку проектів вимагає готовності швидко реагувати на зміни і впроваджувати їх з максимальною оперативністю. При цьому не всі користувачі хочуть або можуть собі дозволити звертатися до розробників, особливо якщо сайт вимагає постійної роботи над ним.

У свою чергу, системи управління контентом дозволяють користувачам, які не володіють навичками розробки сайтів і знаннями мов програмування, самостійно працювати над створенням і зміною сайту.

Існує велика кількість доступних CMS. Завдяки їх функціональності ці системи можна використовувати в різних компаніях. Незважаючи на широкий вибір інструментальних та технічних засобів, наявних в CMS, існують загальні для більшості типів систем характеристики [1].

Системи управління веб-сайтом часто розраховані на роботу у певному програмному середовищі.

Основним завданням такої системи є збір і об'єднання в єдине ціле, на основі ролей і завдань, різних джерел інформації. Ці джерела можуть бути доступні як всередині самої організації, так і поза її межами. До того ж дана система забезпечує можливість взаємодії різних співробітників, проектів і робочих груп, з тими базами знань і даних, які були раніше створені, в

такому вигляді і таким способом, щоб зробити процес пошуку і повторного використання максимально комфортним і звичним.

У такій системі управління контентом визначається все різноманіття існуючих даних: стандартні документи, музика і звуки, відео, каталоги всілякої інформації і багато-багато іншого. І саме для управління, зберігання, обробки, перегляду і публікації таких даних різними групами користувачів і служать CMS [1].

Суть роботи CMS укладена в схемі наступним поділом його сайту і його дизайну. Користувачеві надається можливість вибрати шаблон - заготовку, якій заздалегідь визначено оформлення сторінки, і залишається тільки заповнити його потрібною інформацією. Більшість систем управління вмістом ґрунтується на використанні візуального редактора (WYSIWYG - від англ. What You See Is What You Get - «що бачиш, то і отримаєш») - програми, що дозволяє за допомогою інтуїтивно зрозумілого інтерфейсу додавати або змінювати інформацію на сайті. Варто відзначити, що сайт не перебуває з сукупності сторінок як такої, а формується динамічно. Доданий контент зберігається в базі даних, наприклад, MySQL, і використовується при генерації сторінки після отримання відповідного запиту з боку клієнта.

Як правило, CMS використовуються для таких сайтів:

- а) блог, форум (WordPress, phpBB, vBulletin);
- б) інтернет-магазин (Magento, OpenCart, osCommerce);
- в) соціальні мережі (InstantCMS, Social Engine);
- г) персональні сайти (WordPress, Monstra);
- д) корпоративні сайти (Joomla, Drupal);
- е) портали (DLE, Drupal).

Тим не менше, більшість CMS гнучко настроюється і можуть бути використані для розробки сайтів різної спрямованості. Наприклад, найбільш популярним і універсальним варіантом є WordPress, на якому можливо створити практично будь-який проект: від особистого сайту до великого порталу або інтернет-магазину.

Переваги CMS:

- а) просто і зручно використовувати;
- б) доступний широкий функціонал за рахунок доповнень, тем, розширень;
- в) сайт можна створити за короткий проміжок часу;
- г) наявність документації.

Недоліки:

- а) не підходять для нетипових завдань;
- б) популярні CMS уразливі;
- в) необхідно стежити за оновленням CMS і сумісністю версій доповнень;
- г) підвищене споживання ресурсів, особливо при використанні плагінів.

1.2 Індивідуальні CMS

Існують CMS, розроблені на замовлення під конкретний проект, як наслідок, їх функціонал не такий широкий, ніж у масових систем, але максимально відповідає поставленим завданням і не містить зайвих інструментів. На індивідуальних CMS створюються складні і масштабні сайти зі специфічними особливостями.

Переваги індивідуальних CMS:

- а) CMS більш стабільна і менш вразлива;
- б) адаптована під потреби проекту;
- в) менше навантажує сервер, тому що не переобтяжена зайвими функціями.

Недоліки індивідуальних CMS:

- а) розширення функціоналу або вирішення технічних труднощів зажадає звернення до розробника;

б) надається на платній основі, часто дорожче, ніж придбання ліцензії на одну з масових CMS.

1.3 Основні функції CMS

Використання CMS має цілий ряд переваг. Користувач має можливість ефективно управляти своїм сайтом. Завдяки різноманітності CMS можна підібрати підходяще програмне забезпечення. Така програма дозволить швидко і продуктивно вирішувати поставлені завдання. За допомогою CMS можливо створювати, змінювати і видаляти розділи. Крім того, можна редагувати дані без стороннього втручання. Це є основною перевагою такої системи над статичними сайтами. В процесі роботи веб-ресурсу користувачі знаходять безліч помилок. Ця програма дозволяє швидко і ефективно усувати несправності. Сайт працює на сучасних і перевірених технічних рішеннях. Витрати на створення сайту істотно знижуються. Розробнику не потрібен час на вирішення технічних завдань [1].

Завдяки цим перевагам можна виділити основні функції CMS:

- а) створення – користувач отримує універсальний інструмент зі створення контенту;
- б) управління – можливо обмежувати доступ до інформації, додавати, редагувати, видаляти і багато іншого;
- в) публікація – стаття повністю адаптується до дизайну сайту, необхідно тільки внести її в потрібне поле;
- г) подання – можна змінювати кольорову палітру, розташування і багато іншого будь-якого елемента на вашому сайті.

Ці, на перший погляд, прості програми мають широкі можливості. Кожна компанія прагне створити зручний і зрозумілий інтерфейс. Ці основні функції дозволили виділити ряд переваг:

- а) зручність (дружній інтерфейс і зрозуміле управління є основними плюсами цього програмного забезпечення);
- б) економія (не знадобиться вдаватися до послуг фахівців і платити за це додаткові гроші);
- в) безпека (завжди можна зробити відкат і будь-які допущені помилки не будуть критичними. Також система протистоїть атакам хакерів. Вся інформація буде в безпеці);
- г) документація (модулі мають help-файли, які допоможуть розібратися в функціях CMS);
- д) мультишаблонність (більшість «движків» підтримує численні шаблони. Крім того, розробники постійно оновлюють систему і надають на нові види);
- е) функціональність (кожен розділ або підрозділ має свої настройки і конфігурації. Залежно від вимог можна змінити сайт аж до найдрібніших деталей);
- ж) комплексність (можна створювати окремі вкладки і надавати контроль над ними одному або декільком співробітникам (стрічка новин, блоги та ін.);
- з) незалежність (користувач програмного забезпечення не залежить від розробників. Він може змінювати конфігурації від свого профілю);
- и) можливість розширення (можна змінити систему і при цьому не втратити інформацію. В процесі перенесення сайт буде функціонувати як і раніше);
- к) привабливий зовнішній вигляд (різна колірна гамма і зручна панель дозволить отримувати естетичне задоволення від роботи);
- л) керованість (система проста у використанні. За допомогою пари кліків можна змінити структуру, поміняти місцями пункти, розділи, кореневу папку, підняти рядок вище або нижче і багато іншого) [1].

Кожна CMS незалежно від виробника створювалася з урахуванням всіх вимог користувача.

1.4 Різновиди існуючих CMS

Деякі системи орієнтовані тільки на вирішення конкретних завдань (ведення блогів, інтернет магазини, форуми), інші є універсальними і надають розробником зручне середовище проектування і програмування для розробки чого завгодно. Частина CMS складається з безлічі функціональних блоків і модулів, інші монолітні, неподільні, та ще й зашифровані. Одні системи поставляються безкоштовно і з можливістю внесення своїх доробок, а деякі надаються за гроші і не допускають можливість редагування ядра [2].

До цих пір немає єдиної і чіткої класифікації, прийнятої ринком, існуючих CMS, однак це не заважає виділити лідерів в даній області.

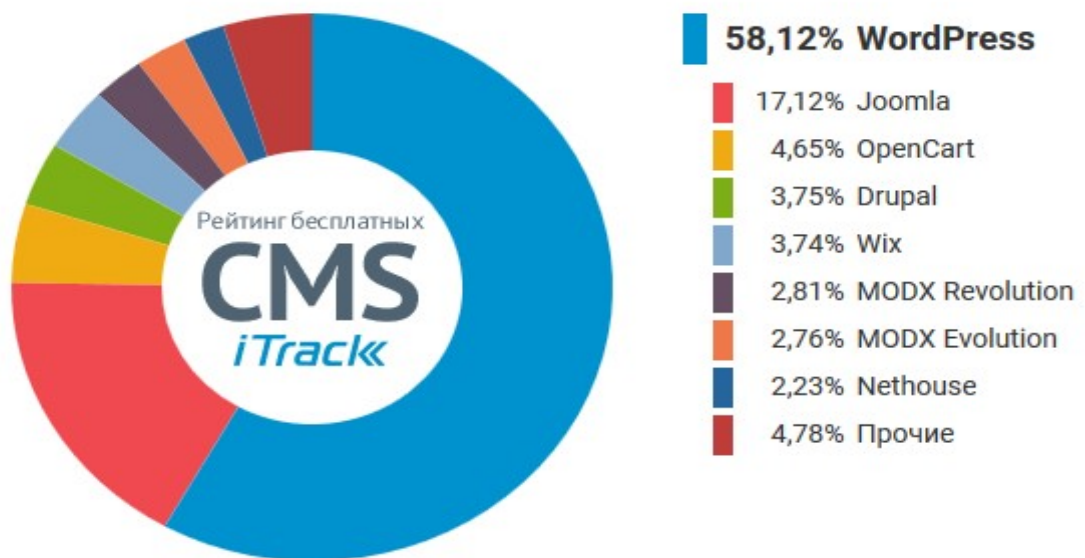


Рисунок 1.1 – Загальний рейтинг безкоштовних CMS за 2019 рік

Можна навести короткий їх перелік: Joomla, Drupal, MODx, 1С Бітрікс, WordPress, DLE, eZ publish, Koobi, Simple Machines Forum (SMF), MediaWiki, SiteDNK, CM5, NetCat, TYPO3, Plone, CMS Made, Simple, Movable Type, InstantCMS, MaxSite CMS, UMI.CMS, Magento, Twilight CMS, PHP-Fusion, Melbis, Miva Merchant, S.Builder, ABO.CMS, PHP -Nuke, RUNCMS,

Adlabs.CMS, Introweb-CMS. На рисунку 1.1 зображено загальний рейтинг CMS за 2019 рік.

Безкоштовність CMS не означає низьку якість коду або наявність обмежень у використанні. Навпаки, найчастіше, відриті і безкоштовні CMS багато в чому виграють у своїх комерційних побратимів саме через свою загальнодоступність. Над цими системами працюють не тільки колективи професійних розробників, але і сотні добровольців, які прагнуть зробити CMS ще краще і безпечніше.

Системи управління сайтом можна розділити на наступні види [2]:

а) студійні CMS – системи управління вмістом, написані всередині веб-студії і використовувані в ній же для розробки власних або клієнтських сайтів. Можуть не мати власної назви. Найчастіше містять тільки той функціонал, який потрібен в конкретному проекті, не навантажуючи сервера хостингу зайвими надбудовами, які не використовуються в даному проекті;

б) тиражні CMS – CMS, розроблені спеціалізованими компаніями або веб-студіями для подальшого масового продажу кінцевим користувачам або іншим веб-студіям, які з яких-небудь причин не мають власної студійної CMS. Оплачуються найчастіше кожна окрема ліцензія для кожного створюваного сайту. Бувають винятки, наприклад одна ліцензія на CMS від «1С-Бітрікс» дозволяє створити відразу 2 сайта, але тільки з однією базою даних. Тиражні CMS містять зазвичай великий функціонал, який, як показує практика навіть на 50% зазвичай не використовується власниками подібних сайтів. Зазвичай містять вичерпну документацію для розробників. Приклад тиражних CMS: «1С-Бітрікс», «UMI.CMS», «NetCat», «AMIRO.CMS»;

в) Open Source CMS (безкоштовні) – системи управління, які розробляються спільнотами програмістів, підтримуються ними ж і мають відкритий вихідний код. Будь-який програміст може взяти безкоштовну CMS і доопрацювати її на власний розсуд. Оскільки розробку веде безліч програмістів з документацією дуже часто виникають проблеми, що підвищує поріг входження програміста в роботу з даними CMS. Мінусом подібних CMS є їхня масовість і, як наслідок, загальновідомість всіх «дірок» в безпеці,

що часто призводить до зараження сайтів на Open Source CMS вірусами, що може загрожувати тривалим падінням позицій сайту в результатах видачі пошукових систем. Приклад безкоштовних CMS: «WordPress», «Drupal», «Joomla», «MODX», «Opencart», «Instant CMS», «TYPO 3» [1].

Також поділяють всі CMS за метою призначення на такі типи:

- а) web CMS для управління веб-сайтами (наприклад, енциклопедіями, подібними до Вікіпедії, онлайн-виданнями, блогами, форумами, корпоративними чи персональними веб-сторінками та ін.);
- б) транзакційні CMS для забезпечення транзакцій у електронній комерції;
- в) інтегровані CMS для роботи з документацією на підприємствах;
- г) електронні бібліотеки (Digital Asset Management) для забезпечення циклу життя файлів електронних медіа (відео, графічних, презентації тощо);
- д) системи для забезпечення циклу життя документації (інструкції, довідники, описи);
- е) освітні CMS – системи для організації Інтернет курсів та відповідного циклу життя документації;
- ж) платформенні CMS (Platform Content Management Systems) підтримують автоматизацію роботи з комп'ютерними файлами, папками, програмами у визначеному програмному середовищі;
- з) корпоративні СКВ (Enterprise content management systems) з різноплановим пристосуванням для потреб підприємницької діяльності. Підтримують цикл життя внутрішньої і зовнішньої документації [2].

1.5 Загальний огляд CMS WordPress

Система WordPress спочатку зарекомендувала себе як інноваційна блог-платформа з високою популярністю. WordPress – це проста у встановленні та використанні система керування вмістом з відкритим кодом,

яка широко використовується для створення веб-сайтів. WordPress використовується всюди, починаючи від особистих блогів і закінчуючи комерційними сайтами. Вбудований візуальний і текстовий (HTML) редактор має зрозумілий інтерфейс, тому користуватися ним дуже просто вже з першого знайомства. Якщо людина знає Microsoft Word, то освоїти редактор WordPress – дуже легко [6].

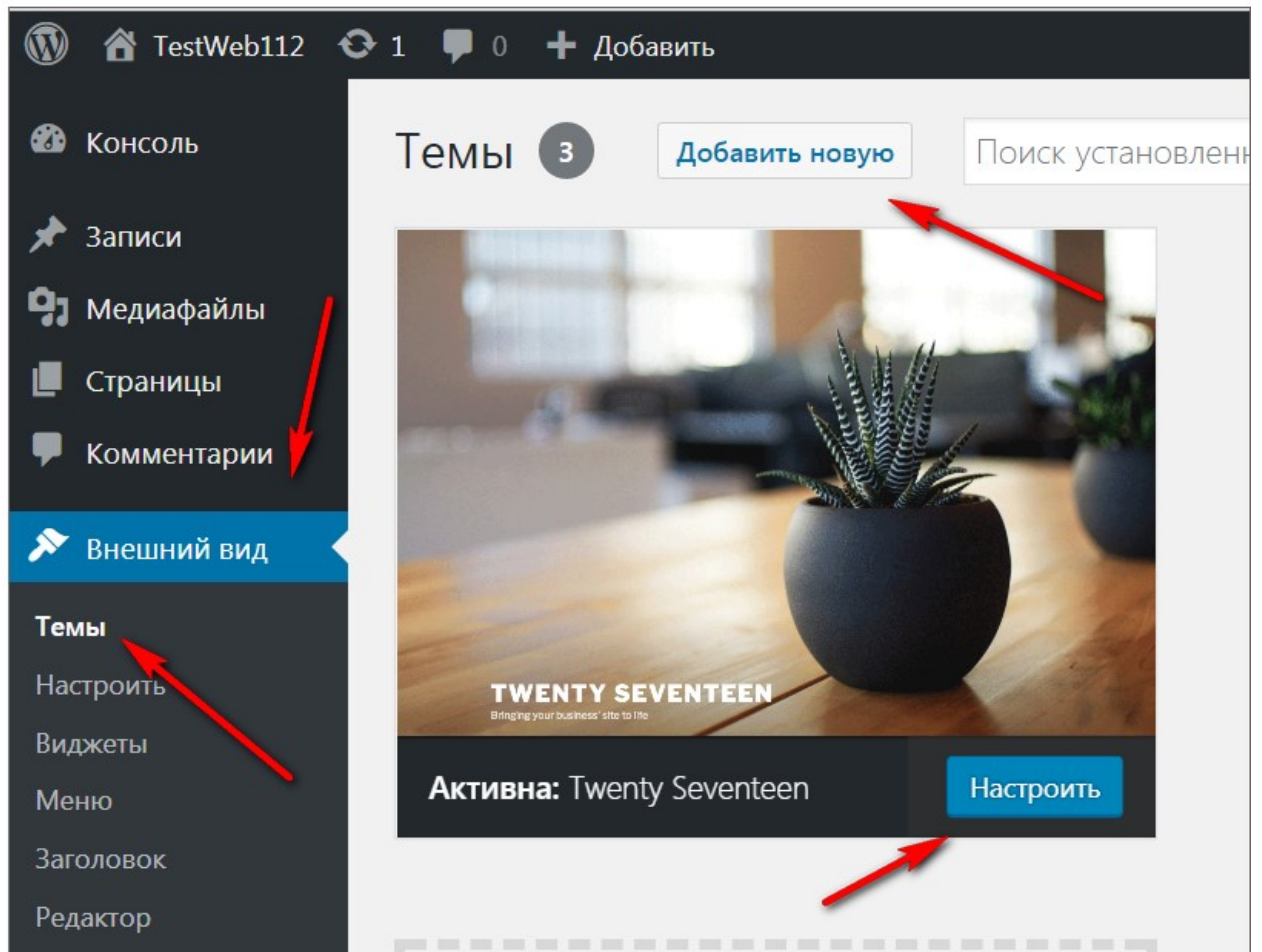


Рисунок 1.2 – Приклад роботи редактора WordPress

Тут можна робити з текстом і картинками все, що завгодно – форматувати, вставляти посилання, відео, зображення і звукові файли. Всі ці операції нескладні і робляться одним-двома кліками. Приклад роботи редактора зображено на рис. 1.2.

Переваги редактора:

а) це найбільш популярна CMS: більше половини користувачів віддають перевагу саме WordPress;

б) найбільш широкий набір плагінів, тем, віджетів для галерей, форумів, багатомовність, різні каталоги, магазини і так далі;

в) технічний досвід не обов'язковий. Адміністративна панель набагато простіше, ніж в інших CMS. PHP і CSS файли можна редагувати безпосередньо в адміністративній частині. Наприклад, можна легко вставити текст з будь-якого текстового редактора;

г) WordPress – досить сильний інструмент для розробників і дизайнерів, які створюють сайти для клієнтів.

Ця CMS може бути кращим варіантом, якщо потрібен легко адмініструємий веб-сайт. Використання даної системи забезпечує широкий, але в той же час простий функціонал. WordPress – кращий варіант для новинних сайтів або блогів [6].

З недоліків варто виділити наступні:

а) досить вагоме навантаження на сервер;

б) часом низька швидкість завантаження сайту;

в) обмежений набір базових функцій;

г) дублювання сторінок і зображень;

д) велика кількість неякісних тем і доповнень з помилками;

е) відносна недовіра пошукових систем до сайтів на безкоштовних двигунах.

Редактор WordPress з великою кількістю доповнень до його двигуна може істотно навантажувати сервер, на якому розташовується сайт. Щоб уникнути надмірного навантаження потрібно скорочувати кількість встановлених плагінів до необхідного мінімуму. Крім того, важливо розміщувати блог на якісному хостингу. Ці опції допоможуть також збільшити швидкість завантаження сайту. Крім того, для прискорення роботи блогу потрібно оптимізувати базу даних і використовувати плагін для кешування сторінок [9].

Одним з недоліків WordPress є те, що він автоматично генерує дублі сторінок, а це негативно позначається на просуванні блогу. Потрібно видаляти дублі сторінок і запобігати їх появі. Також двигун WordPress при завантаженні картинок створює їх копії різних розмірів.

1.6 Загальні характеристики CMS Joomla

Серед популярних CMS ви також можете зустріти Joomla, яка набула широкого поширення завдяки простому інтерфейсу і корисного функціоналу [18].

Цю систему управління використовують як для великих, так для невеликих інтернет-проектів. ПО працює на відкритому коді. Кожен користувач має можливість налаштовувати продукт в залежності від його побажань. Ще одна перевага – безкоштовне розповсюдження. Вам не доведеться платити за програму і оновлення.

Joomla використовується для різних інтернет-проектів. Її функціонал дозволяє зробити роботу сайту ефективною незалежно від його масштабу. Для того, щоб додавати статті, завантажувати файли і керувати сайтом в цілому необхідно встановити «движок» на хостинг. Процес не займе багато часу. ПО підтримує хостинг на PHP, MySQL і Apache. Більшість провайдерів пропонує автоматичну установку [18] (рис.1.3).

Існує чотири версії Joomla – з довгостроковою підтримкою і стандартно. Крім того, розробники стежать за сучасним ринком і підтримують систему управління. Ви зможете завантажити безліч доповнень на свій «движок». Кожна програма має пакет корисних і ефективних функцій [18].

Ось деякі з переваг цієї CMS:

- а) збереження незаповнених публікацій;
- б) драйвер PostgreSQL;
- в) швидкодію за рахунок PHP Memcached;

- г) новинні стрічки можна використовувати як на SimplePie, так і JFeed;
- д) оновлення системних тестів CMS і інші.



Рисунок 1.3 – CMS Joomla

Також існують і недоліки системи. При роботі з CMS Joomla створюється багато зайвого коду. Необхідність його використання не завжди знаходить виправдання, крім того, початківець веб-майстер не зможе впоратися з ним. Довге завантаження сторінок сайту. Це пов'язано з першим недоліком – зайвим кодом. Адміністративна панель не має дружній інтерфейс. Тому працювати новачкам з CMS Joomla не рекомендується, особливо якщо існує можливість використовувати інші системи.

1.7 Огляд та характеристики CMS Modx

Modx знаходиться в першій п'ятірці зведеного топа всіх CMS: комерційних коробкових і Open Source [18]. Сам движок відноситься до Open Source, тобто має відкритий вихідний код, що дозволяє створювати не тільки сайти, але і додатки. Переваги Modx:

- а) проста, інтуїтивно зрозуміла панель адміністратора;

б) зручний і функціональний модуль SEO і можна настроювати налаштування, необхідні для внутрішньої оптимізації і просування сторінки в пошукових системах;

в) адекватні вимоги до хостингу. Для роботи Modx достатньо наявності бази даних MySQL, веб-сервера Apache або IIS і підтримка PHP;

г) власний синтаксис тегів. Спочатку це здається незвичним, але на практиці стає ясно, що такий підхід дозволяє залишати код шаблону чистим і зрозумілим, на відміну від інших Open Source CMS;

д) Modx пропонує додати на сайт цілий ряд корисних модулів: підписку, платіжні системи, інтернет-магазин, голосування, форум, блоги, банери, галереї і т.д.

Чим більше досвіду у веб-майстра, тим більше можливостей надає Modx. Новачкові може здатися, що система бідна настройками, але це помилкова думка, причиною появи якого є грамотна оптимізація адміністративної панелі [18].

Недоліки Modx.

У Modx є перекладена на українську мову документація і російськомовне співтовариство, але через невелику чисельність, корисних матеріалів менше, ніж у інших популярних безкоштовних CMS.

Невелика кількість готових шаблонів. Якщо для WordPress або Joomla можна легко знайти багато дійсно якісних безкоштовних шаблонів, то на Modx не набереться і декількох десятків. Modx використовує іншу термінологію, ніж інші CMS. Modx поєднує в собі функціональність, кроссбраузерність і простоту управління. Якщо робота з Revolution здається складною, можна встановити Modx Evolution. Modx не обмежує веб-майстра свободу вираження, пропонуючи зробити сайт будь-якого типу – від простого односторінного з пропозицією послуги до корпоративного сайту або інтернет-магазину. Але для цього потрібно знати хоча б основи HTML і CSS, так що зовсім новачкам в веб-програмуванні на цьому движку буде важко. Modx Revolution добре захищена, а у Modx Evolution захист трохи

слабше, тому для сайтів, на яких передбачені платежі, краще використовувати версію Revolution [18].

1.8 Загальний огляд CMS Drupal

Drupal – популярна вільна модульна система керування вмістом (СКВ) з відкритим початковим кодом, написана на мові програмування PHP (рис. 1.4).

Drupal використовують як back end фреймворк для різних веб-сайтів, від особистих блогів до корпоративних та державних сайтів. Drupal також використовується у системах управління знаннями та для ділової співпраці.

Drupal може працювати у таких популярних системах як Windows, Mac OS X, Linux, власне, на будь-якій платформі, яка підтримує роботу веб-сервера Apache, Nginx, Lighttpd або Microsoft IIS; також потрібна наявність системи керування базами даних MySQL/MariaDB, PostgreSQL 8.3, SQLite чи інші комерційні [12].

У дистрибутив системи входить набір модулів, що дають наступні можливості:

- а) збір інформаційних стрічок (RSS, RDF, Atom);
- б) ведення блогів, підшивань і форумів;
- в) створення форм для відправки повідомлень;
- г) локалізація системи;
- д) перейменування посилань (призначення посиланням зрозумілих і зручних псевдонімів);
- е) проведення опитувань;
- ж) призначені для користувача профілі, що налаштовуються;
- з) пошук за змістом (за зміст вважається і повідомлення на форумах, і сторінки, і будь-які інші призначені елементи);
- и) ведення журналу статистики (відвідуваності);

к) таксономія (впорядкування матеріалу за категоріями) – дуже «цінна» можливість;

л) формування сторінок з матеріалами в різних формах і форматах подання та інші.

Структура та потужна база модулів Drupal дозволяє порівняно швидко створювати потужні інтерактивні сайти.

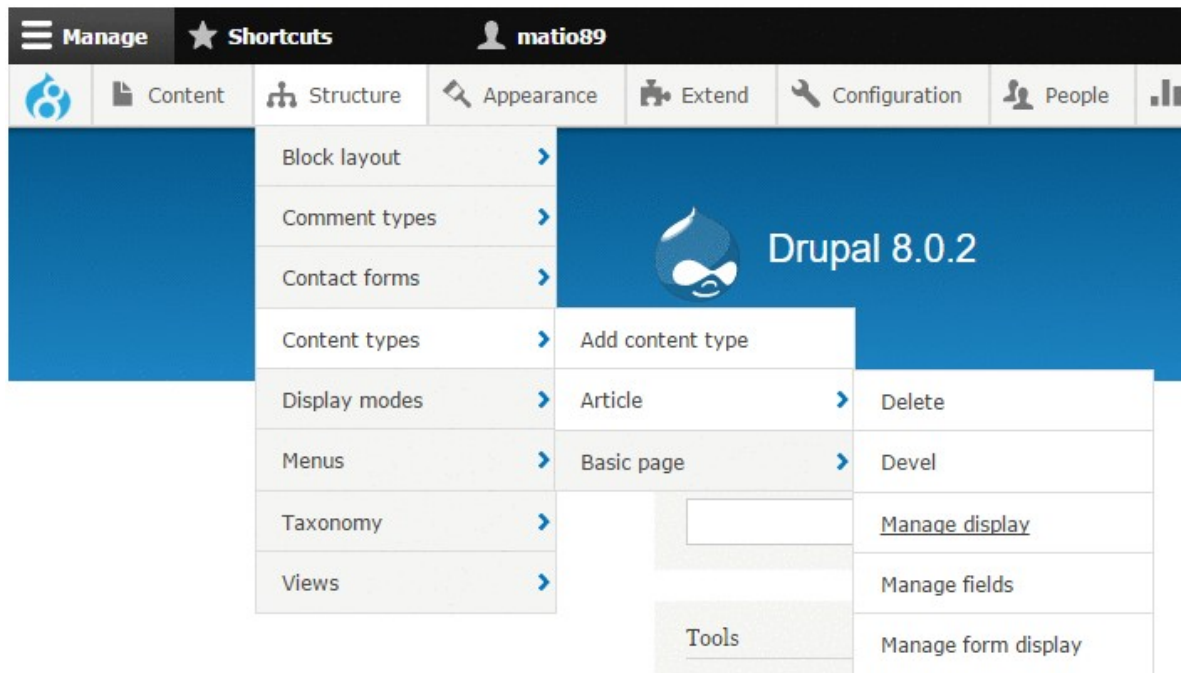


Рисунок 1.4 – Приклад роботи CMS Drupal

До базового пакету системи, окрім модулів створення статичних сторінок (сторінок з постійною адресою) та нових статей входять модулі для організації блогів (електронних журналів користувачів), форумів (місць для інтернет-дискусій), «книг» (інформаційних добірок, праця над якими ведеться колективно), синдикації (імпорту новин з інших сайтів), модуль керування інформаційними блоками на сторінках, що полегшують керування їх виглядом, модуль керування меню.

Drupal підтримує різні теми оформлення та дозволяє створювати свої теми оформлення [13].

Спільнотою розробників Drupal створено багато додаткових модулів, серед яких варто згадати модулі інтернаціоналізації (створення багатомовних

сайтів), модулі керування файлами, що дозволяють викладати на сайтах звукові та відео-файли, модулі категоризації вмісту, модулі організації користувачів у групи та спільноти.

Недоліки даної системи. Непідготовленому користувачеві нелегко створити і підтримувати проект на цій CMS. Drupal повільніший і важчий для хостингу, ніж будь-які інші CMS.

Висновки до розділу 1

Отже, в процесі аналізу функціональності найбільш популярних CMS можна дійти до висновку щодо переваг Drupal.

Дана CMS універсальна – вона підходить як для компактних ресурсів, так і для досить великих інформаційних проектів. Функціональність Drupal – велика добірка модулів, версій дизайнів, тощо. На сьогодні над розробкою і програмною підтримкою CMS Drupal працюють сотні вебмайстрів по всьому світу.

Серед переваг можна виділити:

- а) відкритий вихідний код, завдяки якому система постійно перевіряється і поліпшується різними фахівцями;
- б) високий рівень безпеки;
- в) функціональність на межі універсальності.

Модулі Drupal забезпечують додавання в «тіло» CMS нових можливостей: форми замовлення, відеочату, системи оцінок, медіагалерею і т.д. Доповнення може створити будь-який користувач, але перед публікацією у відкритому доступі модуль обов'язково перевіряється.

З Drupal зараз працюють досить серйозні сайти і це диктує дотримання стабільності та безпеки. Ця система управління є можливістю створювати досить якісні функціональні сайти, які були б прості в управлінні.

2 ПРОЕКТУВАННЯ МОДУЛЯ ВІДОБРАЖЕННЯ МАТЕРІАЛІВ НА КАРТАХ

2.1 Діаграма розгортання. Діаграма компонентів

Діаграма розгортання – діаграма в UML, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Діаграма розгортання відображає робочі екземпляри компонент.

Таким чином, діаграма розгортання призначена для візуалізації елементів і компонентів системи, існуючих лише на етапі її виконання, до яких відносяться файли, динамічні бібліотеки, таблиці БД і т. д. Основні цілі, що переслідуються при розробці діаграми розгортання: розподіл компонентів системи по її фізичних вузлах; відображення фізичних зв'язків між вузлами системи на етапі виконання; виявлення вузьких місць системи і реконфігурація її топології для досягнення необхідної продуктивності. Елементами діаграми розгортання є вузли, компоненти і зв'язки між ними.

Діаграма компонентів описує особливості фізичного представлення системи. Вона дозволяє визначити архітектуру розроблюваної системи, встановивши залежності між програмними компонентами, в ролі яких може виступати вихідний, бінарний і виконуваний код. Основними графічними елементами діаграми компонентів є компоненти, інтерфейси і залежності між ними.

Діаграма компонентів розробляється для наступних цілей:

- 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 2.1.

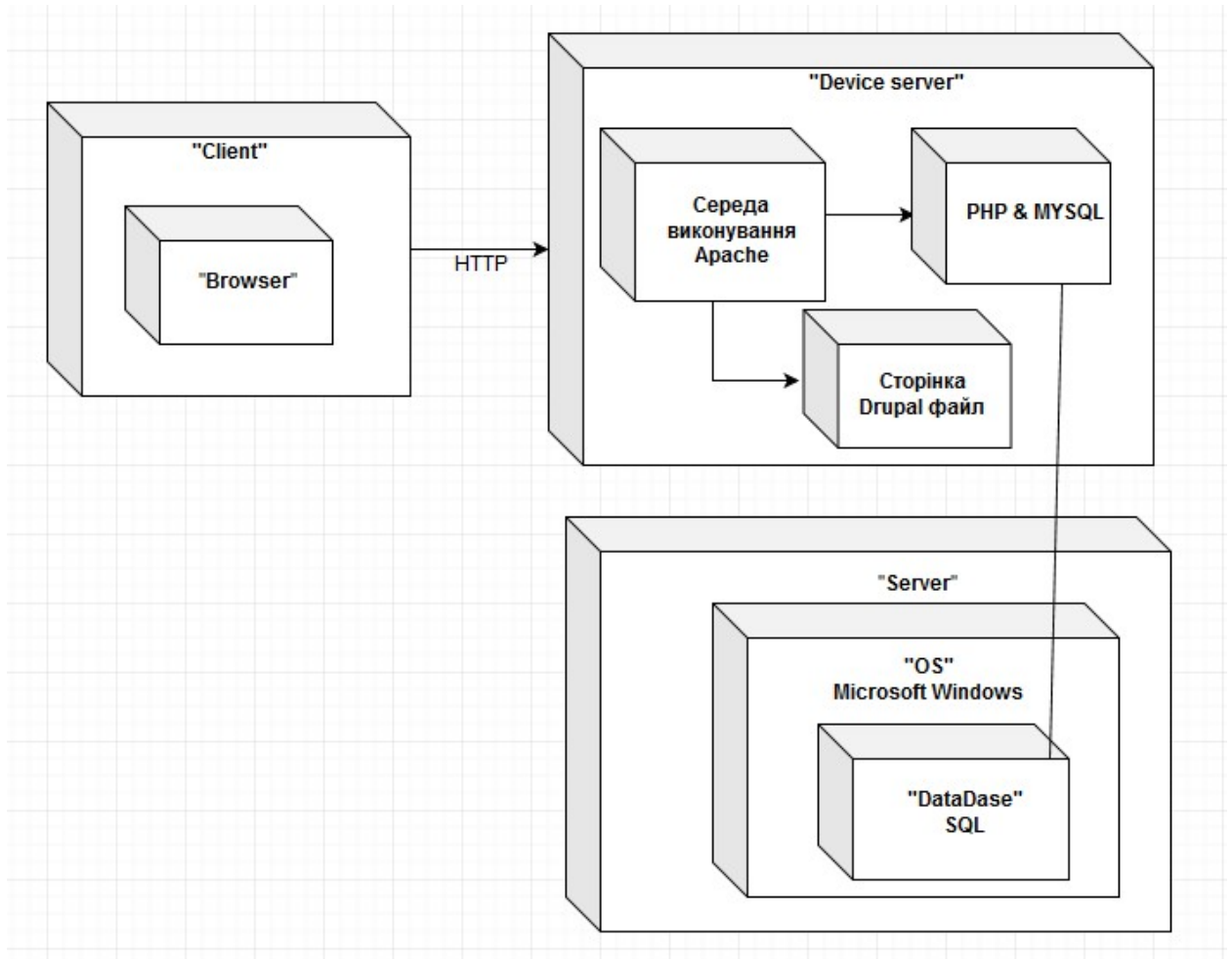


Рисунок 2.1 – Діаграма компонентів

На діаграмі розгортання вузлами системи є комп'ютер користувача з підключенням до мережі інтернет і сервер з певною операційною системою, системою керування вмістом Drupal та базою даних SQL.

Під час роботи з'являється наступна послідовність дій:

- а) клієнт вводить певну адресу в браузер;
- б) запит за цією адресою направляється на nginx;
- в) Nginx визначає, що даний запит динамічний, та передає його далі на сторону Apache;
- г) Apache запускає index.php сайту;

- д) якщо кеш пам'ять є – HTML сторінка повертається користувачу і закінчується виконання запиту;
- е) якщо кеша немає – Drupal починає повне завантаження і рендер сторінки (з'єднання з базою даних, завантаження блоків сторінки і так далі);
- ж) по завершенню виконання завантаження сторінки, вона у відповідному вигляді складається у кеш пам'ять і показується користувачу.

2.2 Структура CMS Drupal

Drupal може працювати на будь-якій операційній системі Windows, Linux, MAC OS і т.д. Drupal, як і будь-яка інша система управління вмістом є інтерфейсом між користувачем (відвідувачем сайту) і тією операційною системою, на яку встановлено Drupal. Drupal – це інтерфейс, який дозволяє отримувати інформацію користувачеві в зручному графічному вигляді, через протоколи передачі даних [13].

Структуру Drupal слід представляти не з самої CMS, а з операційної системи, на яку встановлюється Drupal. Щоб Drupal функціонував необхідно наступне: операційна система, сервер баз даних (MySQL), веб-сервер (Apache) і PHP сервер. Зверніть увагу: Drupal буде функціонувати і без FTP сервера.

Операційна система, на яку встановлюються сервера, необхідні для роботи Drupal, потрібна для того, щоб організувати мережеві підключення до сайту Drupal, а також призначати права доступу до різних папок і файлів системи керування вмістом. Веб-сервер або HTTP сервер надає доступ до сайту по протоколу передачі даних, наприклад HTTP, найчастіше в якості такого сервера використовується Apache. На сервері баз даних зберігається весь вміст сайту: текстовий вміст, облікові записи користувачів сайту, параметри і конфігурації сайту на Drupal, початкові установки сайту. PHP сервер забезпечує взаємодію між сервером баз даних і веб-сервером, у

сервера PHP одна задача – динамічно генерувати сторінки сайту, тобто переміщувати рядки і символи з бази даних до веб-сервера і назад [16].

Необхідно врахувати, що CMS Drupal складається з декількох шарів або рівнів. Найнижчий рівень системи Drupal призначений для обробки сеансів користувачів, механізм аутентифікації, фільтр підсистеми безпеки Drupal і система відображення шаблонів Drupal. Над базовим рівнем Drupal, знаходиться ще один рівень, який призначений для розширення функціоналу сайту на Drupal або іншими словами рівень модулів Drupal. І у Drupal є рівень, який надає зручний графічний інтерфейс для користувачів системи Drupal або система тем оформлень.

Система тем Drupal дозволяє створювати теми різної складності, оскільки вона має дуже гнучкі настройки і розширення. У загальному випадку система тем виводить вміст сайту у вигляді XHTML документа, але підтримуються і інші формати документів. Для оформлення сторінок сайту використовуються каскадні таблиці стилів CSS, і звичайно, при створенні тем Drupal для додання динамічності HTML елементам можна використовувати мову JavaScript і різні бібліотеки, наприклад JQuery.

Модуль Drupal – це файл, написаний на мові PHP, який реалізує функції, з якими вміє працювати система керування вмістом Drupal. Всі можливості Drupal, які доступні звичайному відвідувачу сайту або адміністратора сайту на Drupal реалізовані за допомогою модулів Drupal. За допомогою модулів Drupal реалізуються такі можливості, як вхід в систему, щоб додати новий вміст на сайт, створення галерей, рейтингова система, система голосувань – все це реалізується за допомогою модулів Drupal [15].

В Drupal модулі діляться на два типи: базові модулі Drupal, які встановлюються разом з CMS Drupal, вони створюють функціональний мінімум сайту Drupal і зовнішні модулі, які можна встановити в процесі створення сайту на Drupal, ці модулі створюються сторонніми розробниками і призначені для розширення можливостей Drupal. Існує кілька базових модулів, без яких Drupal не працюватиме, ці модулі відключити не можна, решту модулів можна ввімкнути або вимкнути в будь-який момент.

Загальний вигляд структури Drupal зображено на рисунку 2.2:

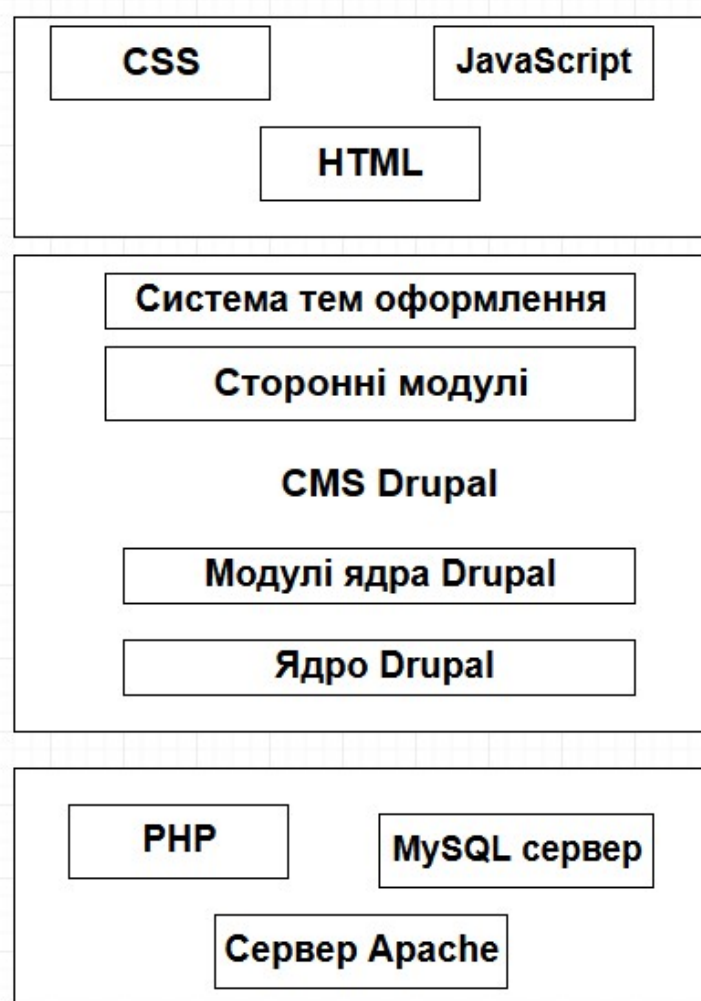


Рисунок 2.2 – Структура Drupal

Розробники модулів Drupal віддають свої переваги створенню модулів, які реалізують якусь одну конкретну функцію. При цьому такі модулі можуть легко взаємодіяти разом з іншими модулями Drupal. Таким чином, комбінуючи модулі один з одним, можна створювати на сайті саме ті можливості, які потрібні. Наприклад, створюючи галерею на сайті Drupal, можна розширити можливості галереї на Drupal іншими модулями, у яких свої власні можливості.

2.3 Діаграма варіантів використання (прецедентів)

Діаграма прецедентів – в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання (рисунок 2.3).

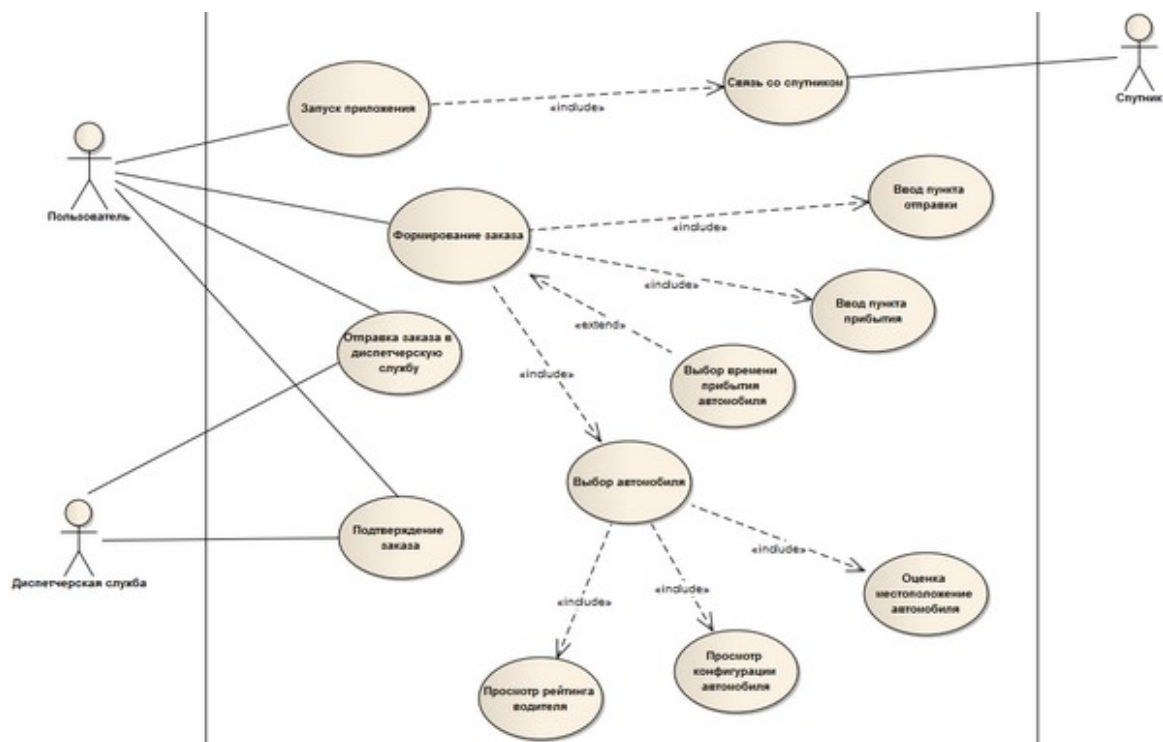


Рисунок 2.3 – Діаграма прецедентів

Дійова особа – зовнішнє джерело взаємодії з додатком через варіанти використання. У нашому випадку дійовими особами є: користувач, супутник і диспетчерська служба.

Варіанти використання:

а) запуск програми користувач запускає програми на своєму пристрої;

б) зв'язок із супутником в момент запуску, додаток з'єднується із супутником для здійснення обміну даними про місце розташування користувача;

в) формування замовлення отримавши дані про своє місцезнаходження, користувач вводить пункт відправлення та пункт прибуття, а також вибирає час прибуття автомобіля і сам автомобіль. Вибір автомобіля складається з перегляду рейтингу водія, конфігурації автомобіля і оцінки місцезнаходження автомобіля;

г) відправка замовлення в диспетчерську службу як тільки користувач завершує формування заявки, додаток відправляє дані про замовлення в диспетчерську службу;

д) підтвердження замовлення диспетчерська служба повідомляє користувачеві повну інформацію про його замовленні, щоб споживач міг її перевірити і після цього підтвердити своє замовлення.

Висновки до розділу 2

Отже, Drupal – це потужна система управління вмістом. В Drupal при реалізації будь-якої функції знайдеться кілька десятків модулів, які її реалізують, кожен з цих модулів робить це по-своєму.

Проаналізувавши діаграми прецедентів, варіантів та компонентів можна зрозуміти як працює система Drupal. Також маємо чітку картину, які дії можуть виконувати користувачі та адміністратори системи.

Також можна зробити висновки, що діаграма розгортання призначена для візуалізації елементів і компонентів системи, існуючих лише на етапі її виконання, до яких відносяться файли, динамічні бібліотеки, таблиці БД і т. д. Основні цілі, що переслідуються при розробці діаграми розгортання: розподіл компонентів системи по її фізичних вузлах; відображення фізичних зв'язків між вузлами системи на етапі виконання; виявлення вузьких місць системи і реконфігурація її топології для досягнення необхідної продуктивності.

3 ЕТАПИ РОЗРОБКИ МОДУЛЯ GOOGLE КАРТ

3.1 Базові налаштування модуля Drupal

Для того, щоб зробити виклик таксі необхідно створити модуль відображення матеріалів на картах. Тобто такий модуль де користувач зможе відмітити на карті початкову точку перебування з координатами де він знаходиться, а також кінцеву точку. Це надасть можливість чітко прокласти маршрут та визначити його довжину.

Для створення модуля відображення матеріалів на картах Google CMS Drupal необхідно почати з наступних дій:

а) в кореневій папці «modules» створюємо папку для нашого модуля: drupal_google_map;

б) створюємо пустий файл drupal_google_map.info.yml.

Ці кроки є необхідними для початку створення модуля.

Файл «helloworld.info.yml» відповідає за базову інформацію про модуль. Наступним кроком є написання програмного коду, який обов'язково містить:

а) назва модуля (відображається у списку модулів): **name:**

Drupal 8 Google Maps Implementation;

б) опис модуля, пишеться виключно англійською мовою:

description: 'This module implements a field for Google Maps API using';

в) оголошуємо, що це модуль: **type: module;**

г) версія ядра для якої модуль належить: **core: 8.x;**

д) версія модуля: **version: 1.0;**

е) пакет для модуля, розділ в модулях де буде знаходитись створений модуль: **package: Geolocation;**

ж) залежності: **dependencies: field, text.**

Після усіх кроків модуль з'являється у списку і його можна вже включати та видаляти, налаштовувати розширення, налаштовувати право доступу до редагування, видалення, оновлення даних, як зображено на рисунку 3.1.

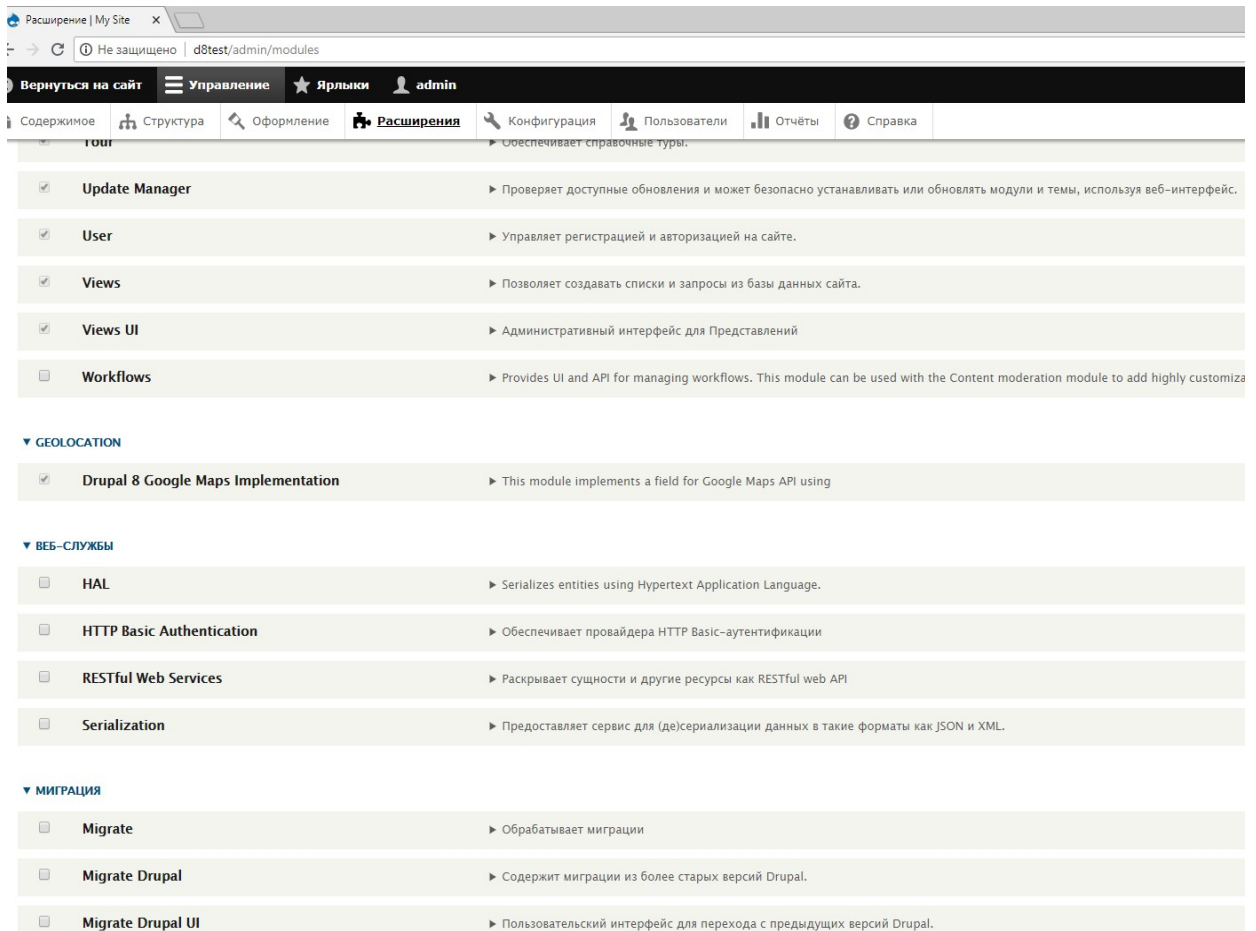


Рисунок 3.1 – Початок створення модуля

3.2 Схема конфігурації, метадані

Drupal 8 містить підтримку мови схеми метаданих Kwalify для файлів налаштування YAML.

Kwalify – це парсер, валідатор схем і інструмент прив'язки даних для YAML і JSON.

YAML і JSON є простим і приємним форматом для структурованих даних, які легше читати і писати, ніж XML [13].

Kwalify підтримує прив'язку даних. Якщо вказати ім'я класу в файлі схеми, парсер Kwalify YAML створює об'єкти екземплярів цього класу замість об'єктів Hash. Це означає, що не потрібно перетворювати Hash на належний об'єкт. Зв'язування даних робить YAML набагато простіше в обігу і маніпуляції

Сам Kwalify написаний на Ruby, і потрібні були невеликі корективи у форматі для створення модуля карт, тому не всі деталі Kwalify безпосередньо застосовуються.

Наступним кроком є створення файлу типу: `confug/schema/drupal_google_map.schema.yml`. Для цього необхідно використати програмний код такого виду, як представлено нижче, де задається висота, ширина вбудованої карти, а також мова підписів на карті:

```
field.formatter.settings.drupal_google_map:
  type: mapping
  label: 'Настройки модуля Drupal Google Map'
  mapping:
    iframe_height:
      type: string
      label: 'Висота вбудованної карти'
    iframe_width:
      type: string
      label: 'Ширина вбудованної карти'
    zoom_level:
      type: integer
      label: 'Зум'
    langcode:
      type: label
      label: 'Мова підписів на карті'
```

translation context: 'Google Maps language code'

apikey:

type: string

label: 'Код облікового запису Google Maps API'

Для взаємодії модулів і ядра Drupal, а також модулів між собою, в Drupal є система хуків. Це виклик функцій, тобто коли виконується виправлення коду, що відбувається після того, як буде додано код функцій, які ми дописуємо в свій модуль. Таким чином можна обробляти дані користувача, меню, таксономії, нод різних континентів в кожен момент зі змін, додавання, видалення або просто завантаження і відображення. Хуків для Drupal 8 досить багато, але менше у порівнянні з Drupal 7. Але це не означає, що Drupal 8 став менш потужним, просто все це реалізується в Drupal 7 хуком, а в Drupal 8 потрібно ще додати для цього відповідний плагін [14].

В даному випадку для реалізації модуля карт були використані наступні хуки:

а) `hook_theme()` один з найбільш фундаментальних хуків в Drupal 8. Все що знаходиться на сторінках, так чи інакше проходить через даний хук. Всі темплейти, які створюються або існують в темі – це все його робота. Даний хук працює як в модулях так і в темах. Він повинен повертати масив, ключі в першому рівні якого будуть ставати "хуками теми" (theme hooks);

б) `variables:` використовується спільно з `#theme`. Даний theme hook буде використовуватися для власного темплейта під якимсь завданням. Відповідно це масив, він може бути як порожнім, так і містити всі можливі змінні, які необхідні темплейту з їх значеннями за замовчуванням. Якщо необхідно зробити темплейт файл, який приймає якісь дані, тут можна вказати всі дані, які потрібні з їх значеннями за замовчуванням, якщо ж невідомо, які значення потрібні за замовчуванням або не потрібні взагалі, то прийнято використовувати NULL;

в) `template`: назва темплейт файлу, який буде використовуватися для рендера. Всі дані будуть передані в даний темплейт і його результат вже буде використаний для виведення. Вказувати розширення тут не потрібно, воно буде автоматично додано відповідно до того движком, який використовується для рендера. Наприклад, з коробки Twig, і відповідно для темплейта `node.html.twig` тут потрібно вказати просто `node`. Назва не повинна містити нижніх підкреслень, тільки тире. Якщо дане значення не встановлено, то воно буде встановлено автоматично по тій же самій логіці. Назва `theme hook` буде встановлено як значення з заміною нижніх підкреслень на тире. Важливо: темплейт файли за замовчуванням шукаються в `/templates`, або в тій папці, яка вказана в `path`, але в обох випадках вони повинні знаходитися в корені даної папки, ніяких вкладень, але якщо зробити перевизначення темплейта `theme hook` в активній темі, там можна класти в `/templates` зі вкладеннями [16].

Використовуючи хуки отримаємо наступний програмний код, де ставимо значення «NULL» для різних параметрів модуля:

Файл `drupal_google_map.module`:

```
function drupal_google_map_theme() {
  return [
    'drupal_google_map_output' => [
      'variables' => [
        'width' => NULL,
        'height' => NULL,
        'zoom' => NULL,
        'langcode' => NULL,
        'apikey' => NULL,
        'address_text' => NULL,
      ],
      'template' => 'drupal-google-map-output',
    ],
  ],
}
```

3.3 Налаштування відображення матеріалу

Типи матеріалів – найважливіше поняття для користувачів, які працюють з Drupal. За допомогою типів матеріалів будують необхідний шаблон для виведення новин або будь-якого іншого контенту.

Типи матеріалів – особливий функціонал сайту в середовищі Drupal, за допомогою якого здійснюється сортування контенту сайту відповідно до вимог до певного шаблону. Причому для окремого типу матеріалу ми можемо створити специфічне оформлення та індивідуальну структуру [16].

Створений матеріал можна розділити на типові області, які будуть присутні в наступних статтях даного типу матеріалу. Нижче наведено типову будову будь-якого типу матеріалу.

У кожному матеріалі буде присутній:

- а) заголовок;
- б) також можна вказувати дату публікації та автора статті;
- в) основний зміст сторінки, в якому буде представлена сутність статті;
- г) також можна додати якусь картинку або фотографію;
- д) для оптимізації навігації по сайту додають ще посилання на інші статті цього типу матеріалу або інших розділів сайту.

Головною особливістю розділу типів матеріалів є створення областей, які будуть виводиться за допомогою полів [15].

Поле – особлива область, за допомогою якої виводиться допоміжна інформація з використанням різних форматів представлення цієї інформації.

Можна створити поля, які будуть відображати заголовок статті, виводити автора і дату публікації, область для роботи з текстовою частиною основного вмісту.

Варто зазначити, що можна управляти не тільки типами полів, їх особливими форматами і їх вмістом, а й відображенням їх на сторінки

матеріалу. Також можна задавати необхідний порядок розташування полів на сторінки.

Формат поля формує дані поля для перегляду кінцевим користувачем. Формати поля визначаються як плагіни.

Плагін API в Drupal 8 дозволяє модулю або підсистемі забезпечувати функціональність (екземпляри плагінів) об'єктно-орієнтованим способом. Додатки, які виконують подібну функціональність, належать до одного типу плагіна. Розробники модулів записують плагіни для розширення різноманітних систем, таких як блоки, віджети і графічні ефекти, а також додають нові опції для вибору адміністраторів.

Плагіни – це невеликі функціональні можливості, які можна замінювати. Додатки, які виконують подібну функціональність, належать до одного типу плагіна.

Більшість плагінів у Drupal 8 будуть використовувати анотації для реєстрації і опису їх метаданих.

Для більшої частини коду в Drupal Core використовується перегляд анімованих повідомлень [12].

Менеджер плагінів – це центральний клас керування, який визначає, яким чином буде виявлено та інстанційно створено плагіни певного типу.

Нові типи плагінів завжди повинні використовувати спеціальний клас анотацій, тому пізніше можна додавати або змінювати ключі анотацій або документувати їх.

Загалом, плагіни не повинні бути визначені в класі, а в спеціальному модулі.

Наступним кроком у створенні модуля карт є використання плагінів у програмному коді, який представлено нижче:

```
Файл Plugin/Field/FieldFormatter/DrupalGoogleMapFormatter.php:  
namespace Drupal\drupal_google_map\Plugin\Field\FieldFormatter;  
use Drupal\Core\Field\FieldItemListInterface;  
use Drupal\Core\Field\FormatterBase;
```

```

use Drupal\Core\Form\FormStateInterface;

/**
 * Plugin implementation of the 'drupal_google_map' formatter.
 *
 * @FieldFormatter(
 *   id = "drupal_google_map",
 *   label = @Translation("Drupal Google Map"),
 *   field_types = {
 *     "string",
 *     "computed",
 *     "computed_string",
 *   }
 * )
 */

class DrupalGoogleMapFormatter extends FormatterBase
{
  // Методи класу для налаштування відображення поля
}

```

Якщо для форматування потрібні спеціальні налаштування дисплея, для досягнення цієї мети потрібно три етапи:

- а) перевизначити `PluginSettingsBase::defaultSettings()`, щоб встановити типові значення;
- б) створити схему конфігурації для створених налаштувань;
- в) створити форму, щоб дозволити користувачам змінювати налаштування:

```

public static function defaultSettings() {
  return [
    "iframe_height" => "640",
    "iframe_width" => "480",
    "zoom_level" => "1",
  ];
}

```

```

        "langcode" => "ru",
        "apikey" => "",
    ] + parent::defaultSettings();
}

```

Форма, що дозволяє користувачам змінювати значення параметрів, створюється перевизначенням `FormatterBase: settingsForm ()`

```

public function settingsForm(array $form, FormStateInterface $form_state)
{
    $elements = parent::settingsForm($form, $form_state);
    $elements['apikey'] = [
        '#type' => 'textfield',
        '#title' => $this->t('Google Maps API key'),
        '#default_value' => $this->getSetting('apikey'),
        '#description' => $this->t('Static Maps will not work without an API
key. See the <a href="https://developers.google.com/maps/documentation/static-
maps" target="_blank">Static Maps API page</a> to learn more and obtain a
key.'),
    ];
    $elements['iframe_width'] = [
        '#type' => 'textfield',
        '#title' => $this->t('Width of embedded map'),
        '#default_value' => $this->getSetting('iframe_width'),
        '#description' => $this->t('You can set sizes in px or percent (ex: 600px
or 100%). Note that static maps only accept sizes in pixels, without the suffix px
(ex: 600).'),
        '#size' => 10,
    ];
    $elements['iframe_height'] = [
        '#type' => 'textfield',
        '#title' => $this->t('Height of embedded map'),

```



```

    '#default_value' => $this->getSetting('iframe_height'),
    '#description' => $this->t('You can set sizes in px or percent (ex: 600px
or 100%). Note that static maps only accept sizes in pixels, without the suffix px
(ex: 600).'),
    '#size' => 10,
];
$elements['zoom_level'] = [
    '#type' => 'select',
    '#options' => [
        1 => $this->t('1 - Minimum'),
        2 => 2,
        3 => 3,
        4 => 4,
        5 => 5,
        6 => 6,
        7 => 7,
        8 => 8,
        9 => 9,
        10 => 10,
        11 => 11,
        12 => 12,
        13 => 13,
        14 => $this->t('14 - Default'),
        15 => 15,
        16 => 16,
        17 => 17,
        18 => 18,
        19 => 19,
        20 => $this->t('20 - Maximum'),
    ],
];

```

```

    '#title' => $this->t('Zoom level'),
    '#default_value' => $this->getSetting('zoom_level'),
];
$elements['langcode'] = [
    '#type' => 'textfield',
    '#title' => $this->t('Language'),
    '#default_value' => $this->getSetting('langcode'),
    '#description' => $this->t("Enter a two-letter language code that Google
Maps can recognize, or enter 'page' (without the quotes) to use the current page's
language code"),
];
return $elements;

```

У модулі, в папці шаблонів, створюємо шаблон гілки. Ім'я файлу має збігатися з тим, що ввели в `hook_theme()`, де потрібно замінити підкреслення на тире.

Файл `templates/drupal-google-map-output.html.twig`:

```

<iframe
    width="{{ width }}"
    height="{{ height }}"
    frameborder="0" style="border:0"
    src="https://www.google.com/maps/embed/v1/place?key={{ apikey }}
    &q=Space+Needle,Seattle+WA&language={{ langcode }}&zoom={{
zoom }}" allowfullscreen>
</iframe>

```

Особливості використання:

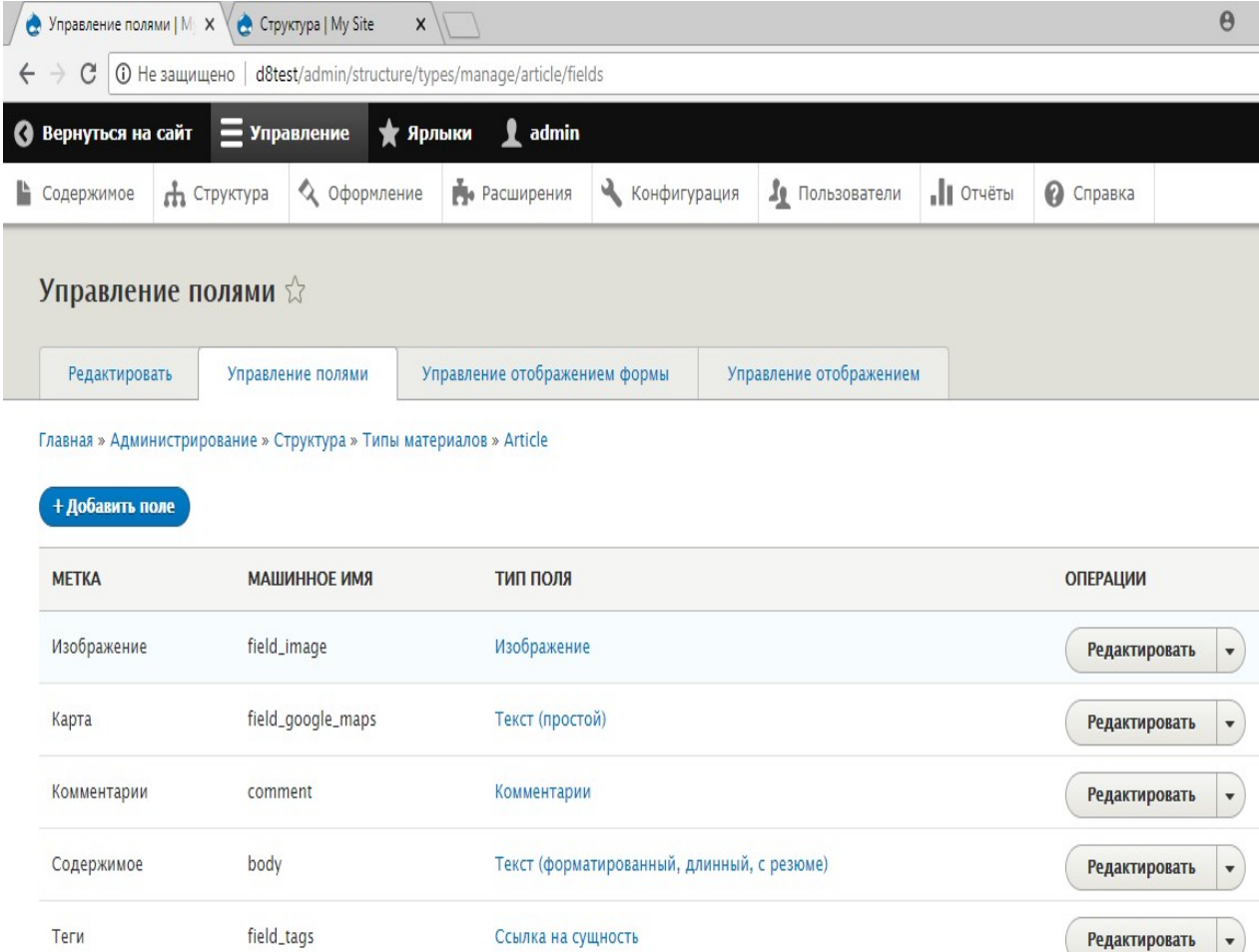
- а) в структурі матеріалу додати нове поле типу «Текст (простий)», що вказує на необхідну мітку і машинне ім'я;
- б) в управлінні відображенням матеріалу для створення поля необхідно вказати формат «Drupal Google Map. У налаштуваннях

відображення поля вказати ключ доступу до API Google Maps, ширину та висоту віджету, рівень масштабу та мову роботи віджету;

в) при створенні або редагуванні матеріалу у налаштованому полі необхідно ввести потрібну адресу для пошуку.

Для підключення модуля карт необхідно зробити наступне: додати поля, відредагувати їх, задати назви кожному полю, тип для кожного поля, як зображено вище на рисунку 3.2.

Наступним кроком буде налаштування розмірів карти, яка буде відображатися на нашому сайті. Для цього задаємо масштаб і мову, на якій будемо бачити всі позначення (рис. 3.3). Також необхідно зберегти всі зроблені зміни.



The screenshot shows the Joomla! administration interface. The browser address bar indicates the URL: `admin/structure/types/manage/article/fields`. The navigation menu includes 'Управление полями' (Manage Fields), 'Структура' (Structure), 'Оформление' (Appearance), 'Расширения' (Extensions), 'Конфигурация' (Configuration), 'Пользователи' (Users), 'Отчёты' (Reports), and 'Справка' (Help). The main content area is titled 'Управление полями' and contains a table of fields for the 'Article' type.

МЕТКА	МАШИННОЕ ИМЯ	ТИП ПОЛЯ	ОПЕРАЦИИ
Изображение	field_image	Изображение	Редактировать
Карта	field_google_maps	Текст (простой)	Редактировать
Комментарии	comment	Комментарии	Редактировать
Содержимое	body	Текст (форматированный, длинный, с резюме)	Редактировать
Теги	field_tags	Ссылка на сущность	Редактировать

Рисунок 3.2 – Управління полями матеріалу

Управление отображен: x Структура | My Site x

Не защищено | d8test/admin/structure/types/manage/article/display

Вернуться на сайт | Управление | Ярлыки | admin

Содержимое | Структура | Оформление | Расширения | Конфигурация | Пользователи | Отчёты | Справка

Теги | Сверху | Метка | Ссылка на указанную сущность

Ссылки

Комментарии | Сверху | Список комментариев | Вариант отображения комментария: По умолчанию

Карта | Сверху

Настройки формата: **Drupal Google Map**

Google Maps API key

AlzaSyDzd9fVgoXmxWq-Hal-EEMp011lobbdeq

Static Maps will not work without an API key. See the [Static Maps API page](#) to learn more and obtain a key.

Width of embedded map

640

You can set sizes in px or percent (ex: 600px or 100%). Note that static maps only accept sizes in pixels, without the suffix px (ex: 600).

Height of embedded map

480

You can set sizes in px or percent (ex: 600px or 100%). Note that static maps only accept sizes in pixels, without the suffix px (ex: 600).

Уровень масштаба

14 - Default

Язык

ru

Enter a two-letter language code that Google Maps can recognize, or enter 'page' (without the quotes) to use the current page's language code

Обновить | Отмена

Отключены

Язык | Сверху | Язык

Рисунок 3.3 – Налаштування відображення модуля

Далі, для створення нового матеріалу, який буде відображено на сайті, або при редагуванні його, необхідно ввести дані у всі запропоновані поля. А саме:

а) в полі «Заголовок» написати заголовок інформаційного матеріалу;

б) у полі «Вміст» ввести основний контент, або тіло статті, при цьому можна змінювати розмір шрифту, додавати посилання про більш детальний опис в мережі Інтернет, або посилання на джерело походження інформації;

- в) також можна додати «Теги», розділивши їх комами, якщо більше одного;
- г) завантажити зображення, але для цього є певні обмеження – лише один файл об'ємом до 100Мб, формату png, jpg, gif, jpeg;
- д) в полі «Карта» потрібно ввести адресу для відображення на картах Google вище наведеної інформації, як зображено на рисунку 3.4.

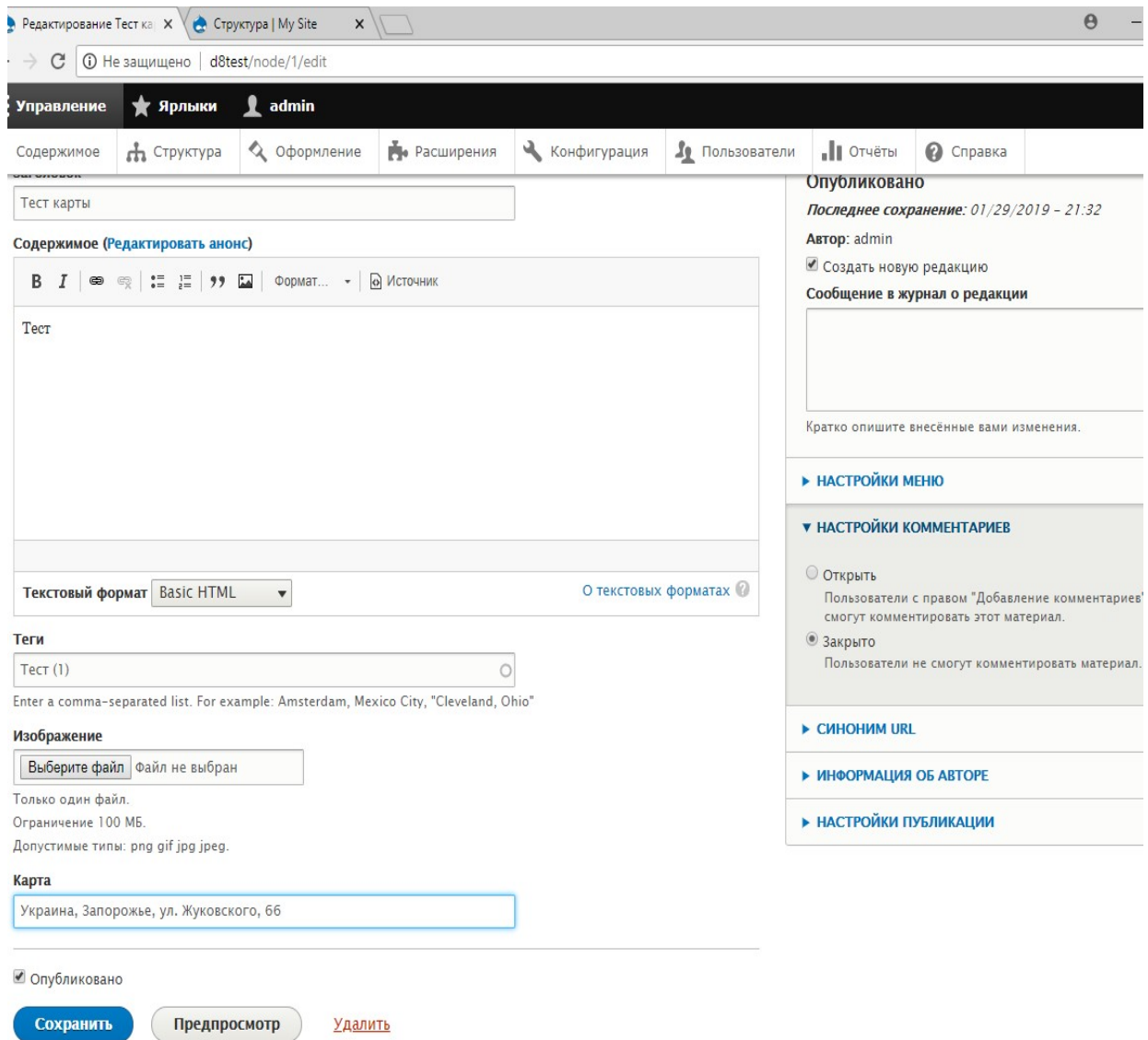


Рисунок 3.4 – Редагування полів матеріалу

По завершенню всіх налаштувань при перегляді матеріалу отримаємо: назву та основний текст, які були введені, зображення до тексту, якщо воно було завантажено та карту з адресою, що була введена у відповідне поле під

час редагування даних, як зображено на рисунках 3.5, 3.6. На правах адміністратора ми можемо цю карту переглядати, редагувати, а також видаляти об'єкти на карті або додавати їх.

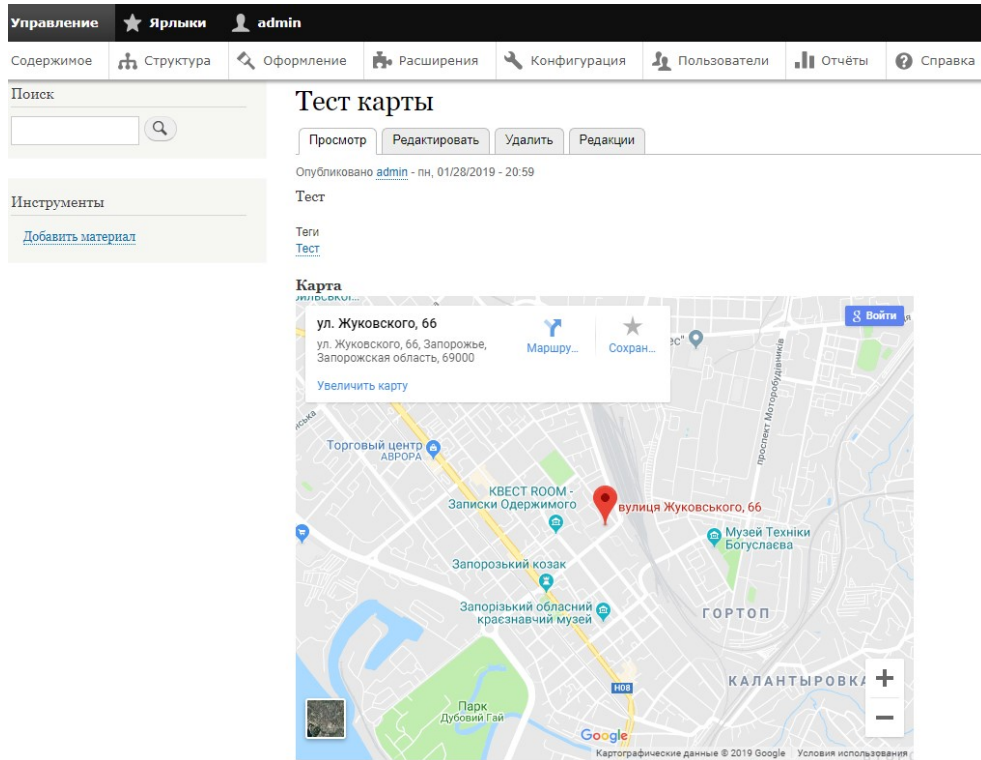


Рисунок 3.5 – Створений модуль карт Google з початковою точкою

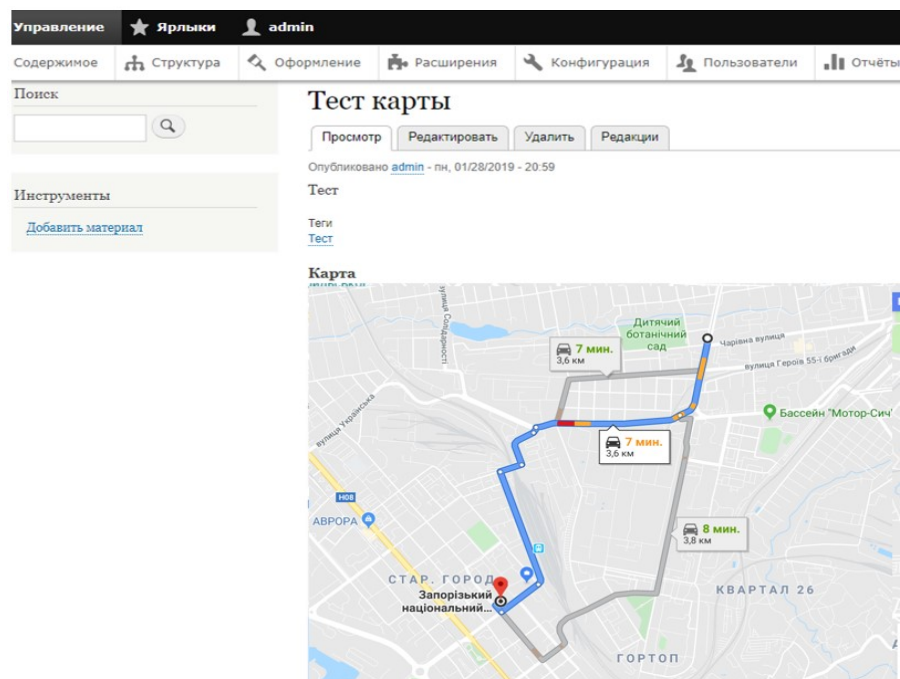


Рисунок 3.6 – Створений модуль карт з маршрутом

ВИСНОВКИ

В даній кваліфікаційній роботі були проаналізовані найпопулярніші CMS та наведено їх позитивні та негативні якості. Здійснено описання та порівняння найпопулярніших CMS, таких як WordPress, Drupal, Joomla. В процесі аналізу функціональності найбільш популярних CMS досліджено, що Drupal якнайбільш підходить для створення модуля карт Google.

Виявлено, що функціональність Drupal – велика добірка модулів, версій дизайнів, тощо. Серед переваг можна виділити наступне: відкритий вихідний код, завдяки якому система постійно перевіряється і поліпшується різними фахівцями; високий рівень безпеки; функціональність на межі універсальності.

Проаналізовано роботу модулів у CMS Drupal. З'ясовано, що вони забезпечують додавання в «тіло» CMS нових можливостей: форми замовлення, відеочату, системи оцінок, медіагалерею і т.д. Доповнення може створити будь-який користувач, але перед публікацією у відкритому доступі модуль обов'язково перевіряється. Проаналізувавши діаграми прецедентів, варіантів та компонентів можна зрозуміти як працює система Drupal.

У результаті виконання кваліфікаційної роботи були вирішені такі задачі: розглянуто основні види існуючих CMS; розглянуто основні переваги і недоліки найпопулярніших CMS; розглянуто інструменти розробки модулів у CMS Drupal; проаналізовано основні програмні засоби для створення модуля карт у Drupal; спроектовано модуль карт Google для CMS Drupal; розроблено модуль карт Google для прокладання маршруту таксі для CMS Drupal.

Отримана у результаті інформаційна система разом зі створеним модулем відображення матеріалів на картах допоможе службам таксі створювати мітки у місті за допомогою Google карти з можливістю вказувати свої параметри для планування маршруту.

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. Створюємо свій сайт. *Навчально-методичний ресурс циклової комісії обліково-економічних дисциплін*. URL:<http://aek-oed.pl.ua/stvoryuyemo-sviy-sayt/> (дата звернення 13.10.2019)
4. Створення Web-сторінок. *Інструментарій для створення Web-сторінок*. URL: http://gymlit.in.ua/re_%D0%A1%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F_Web-%D1%81%D1%82%D0%BE%D1%80%D1%96%D0%BD%D0%BE%D0%BA (дата звернення 16.10.2019)
5. Працюємо над створенням якісного web-сайту. *Веб студія*. URL: <http://www.web-master.lviv.ua/blog/pratsyujemo-nad-stvorennjam-yakisnoho-web-sajtu.html> (дата звернення 16.10.2019)
6. Офіційний сайт WordPress. *Веб розробка*. URL: <https://ru.wordpress.org/> (дата звернення 15.10.2019)
7. Візуальний HTML-редактор DreamWeaver. *Розробка Web-дизайну*. URL: <http://uadoc.zavantag.com/text/22561/index-1.html> (дата звернення 16.11.2019)
8. Найнеобхідніші плагіни для WordPress. *Плагіни для Wordpress*. URL: <https://olegvoloshchuk.com/plaginy-dlya-wordpress/najneobhidnishi-plaginy-dlya-wordpress.html> (дата звернення 18.11.2019)
9. Що таке внутрішня структура сайту. *Веб розробка*. URL: <http://korusno-znatu.in.ua/internet/web-rozrobka/vnytrinnya-stryktyra-sajty.php> (дата звернення 19.11.2019)

10. Створення сайтів за допомогою CMS WordPress. *Створення та просування сайтів*. URL: <http://repair.lviv.ua/stvorenniya-sajtu-za-dopomogoyu-wordpress/> (дата звернення 21.11.2019)
11. Офіційний сайт Drupal. *Веб розробка*. URL : <https://www.drupal.org/> (дата звернення 13.11.2019)
12. Drupal – википедія. *Електронний ресурс*. URL: <http://ru.wikipedia.org/wiki/Drupal> (дата звернення 13.11.2019)
13. Джон К. Вандюк, Мэтт Вестгейт. CMS Drupal. Руководство по разработке системы управления сайтом. Вильямс, 2009. 576 с.
14. Справочник API. Drupal API. *Електронний ресурс*. URL: <http://api.drupal.ru/> (дата звернення 13.11.2019)
15. Разработка модулей для CMS Drupal 7. *Практическое руководство*. Русаков Е.М. 2012 <https://drupal.ru/taxonomy/term/12520> (дата звернення 13.11.2019)
16. Плєскач В. Л. Інформаційні системи і технології на підприємствах. *Засоби створення Web-сайтів*. URL: <http://westudents.com.ua/glavy/27290-zasobi-stvorenniya-Web-saytv.html> (дата звернення 29.11.2019)
17. Українська спільнота Joomla! – підтримка, розширення, локалізація. *Електронний ресурс*. URL: <https://joomla-ua.org/> (дата звернення 29.11.2019)
18. CMF MODX Evolution. *Електронний ресурс*. URL: <http://modx.com.ua/> (дата звернення 25.11.2019)