**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**ДЕРЖАВНИЙ ЗАКЛАД**

**«ЛУГАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА»**

|  |
| --- |
| Навчально-науковий інститут фізики, математики та інформаційних технологій |
| (назва факультету, інституту) |
| Кафедра інформаційних технологій та систем |
| (назва кафедри) |

Пояснювальна записка

до дипломного проєкту (роботи(

БАКАЛАВРА

(освітньо-кваліфікаційний рівень)

Клименко Дарій Костянтинович

На тему: **РОЗРОБКА ВЕБ-САЙТУ** **«МОБІЛОГРАФІЯ ДЛЯ ВСІХ» ЗАСОБАМИ CMS DRUPAL**

Виконав: студент 4 курсу

Напряму підготовки (спеціальності)

121 «Інженерія програмного забезпечення»

(шифр і назва напряму підготоки, спеціальності)

Дарій Клименко

Керівник \_\_ \_ Світлана ПЕРЕЯСЛАВСЬКА

Рецензент \_\_\_\_\_\_\_\_ Юрій КОЗУБ

Полтава – 2023

ЗМІСТ

[ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 3](#_Toc137448101)

[ВСТУП 4](#_Toc137448102)

[РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ СТВОРЕННЯ САЙТІВ 6](#_Toc137448103)

[1.1 Поняття веб сайт та основні етапи його створення 6](#_Toc137448104)

[1.2 Базові мови створення веб сайтів 10](#_Toc137448105)

[1.3 Основні технології розробки веб-сайтів 15](#_Toc137448106)

[1.4 Порівняльна характеристика основних CMS на мові PHP 19](#_Toc137448107)

[Висновки до розділу 1 23](#_Toc137448108)

[РОЗДІЛ 2 ОГЛЯД CMS DRUPAL 24](#_Toc137448109)

[2.1 Порівняльна характеристика Drupal 8, Drupal 9, Drupal 10 24](#_Toc137448110)

[2.2 Основні концепції CMS Drupal 25](#_Toc137448111)

[2.3 Темізація в CMS Drupal 33](#_Toc137448112)

[Висновки до розділу 2 38](#_Toc137448113)

[РОЗГЛЯД 3 РОЗРОБКА САЙТУ «МОБІЛОГРАФІЯ ДЛЯ ВСІХ» 40](#_Toc137448114)

[3.1 Налаштування сайту на CMS Drupal 40](#_Toc137448115)

[3.2 Створення основного контенту сайту 44](#_Toc137448116)

[3.3 Темізація сайту 52](#_Toc137448117)

[Висновки до розділу 3 55](#_Toc137448118)

[ВИСНОВКИ 56](#_Toc137448119)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 58](#_Toc137448120)

[Додатки 60](#_Toc137448121)

[Додаток А Процес реєстрації на platform.sh 60](#_Toc137448122)

[Догадок Б Основні команди composer для праці із Drupal 64](#_Toc137448123)

[Додаток В Зміст info.yml 65](#_Toc137448124)

# ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CMS – Content Management Sysynem

CSS – Cascading Style Sheets

FOSS – Free and Open Source Software

HTML – HyperText Markup Language

JS – JavaScript

JSON – JavaScript Object Notation

LAMP – Linux Apache MySQL PHP

PHP – Personal Home Page Tools

WWW – World Wide Web

XML – Extensible Markup Language

# ВСТУП

**Актуальність роботи.** Створення веб-сайтів має велику важливість у сучасному світі існуючих технологій та цифрового спілкування. Інтернет став основним джерелом інформації та комунікації для мільярдів людей по всьому світу. Веб-сайт дозволяє підприємствам, організаціям, навчальним закладам та приватним особам бути присутніми в Інтернеті, щоб забезпечити доступ до своїх послуг, продуктів чи інформації широкому колу користувачів. Веб-сайти дозволяють забезпечити доступ до інформації та послуг у будь-який час та з будь-якого місця, де є доступ до Інтернету. Веб-сайти створюють можливість для взаємодії з користувачами, надання зворотного зв'язку та комунікації з аудиторією. Вони можуть містити контактні форми, коментарі, форуми або чати, що дозволяють взаємодіяти зі споживачами та вирішувати їх запити або питання.

Drupal як потужна CMS: Drupal є однією з найпопулярніших систем управління веб-контентом. Ця CMS надає широкі можливості для розробки веб-сайтів різного рівня складності. Знання Drupal дає можливість отримати цінні навички, які є великою перевагою на ринку праці, отже тема даної роботи є актуальною для вивчення

**Мета роботи** –розробка веб-сайту засобами CMS Drupal.

**Об’єкт дослідження –** розробка веб-сайтів

**Предмет дослідження** – технологія розробки веб-сайтів із використанням CMS Drupal.

Відповідно до предмета дослідження і мети, були виділені основні **завдання дослідження:**

- розглянути поняття веб-сайт, основні етапи їх створення та базові мови, які використовуються для розробки веб-сайтів;

- розглянути основні підходи і принципи створення веб-сайтів;

- дослідити можливості основних CMS на мові PHP;

- дослідити відмінності версій Drupal, та обрати версію для розробки;

- розглянути основні концепції Drupal та його темізації;

- розробити веб-сайт "Мобілографія для всіх" з допомогою CMS Drupal і протестувати його.

Для вирішення завдань дослідження використано такі **методи дослідження**: *теоретичні:* аналіз наукової літератури, узагальнення та систематизація теоретичних положень про створення веб-сайтів; аналіз технологій створення веб-сайтів; *експериментальні:* тестування розробленого сайту.

До складу роботи входять три розділи, у першому розділі розглядаються веб-сайти, основні етапи їх створення, базові мови та основні технології розробки веб-сайтів, надається порівняльна характеристика основних CMS на мові PHP. Другий розділ присвячений CMS Drupal в ньому проведено порівняння версій 8, 9, 10, розглянуто основні концепції, які потрібно знати для розробки будь якого сайту на цій CMS, а також можливості темізації. У третьому розділі описано основні моменти реалізації сайту «Мобілографія для всіх».

Практичне значення розробки – розроблений веб-сайт «Мобілографія для всіх», який може застосовуватися для вивчення основних прийомів мобілографії та підбору відповідного обладнання.

# РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ СТВОРЕННЯ САЙТІВ

## 1.1 Поняття веб сайт та основні етапи його створення

Історія створення сайтів бере свій початок з 90-х років минулого сторіччя та тісно пов'язана із виникненням мережі Інтернет. Першопрохідником у цій сфері був женевський учений із Європейської лабораторії елементарних частинок Тім Бернерс-Лі. Саме він є творцем першого у світі сайту. Але ще до нього в сорокових роках минулого століття Ваннервар Буш розвивав ідею про те, що завдяки спеціальним технічним пристроям можна розширити людську пам'ять і проіндексувати накопичену століттями інформацію.

Це давало б можливість, на його думку, організувати швидкий пошук необхідної інформації. Саме такий принцип лежить в основі сьогоднішніх сайтів. Інформація про дату створення першого сайту в літературних джерелах відрізняється. Одні дослідники стверджують, що це сталося 1990 року, інші ж датують цю подію 6 серпнем 1991 року. [15]

На цьому сайті було розміщено опис розробленої Бернерсом-Лі технології WWW (World Wide Web), що базується на протоколі передачі даних HTTP, системі адресації URL та розмітці гіпертексту HTML. Крім цього, на сайті знаходилася низка посилань на інші сайти, тобто фактично він був першим Інтернет-каталогом у світі. Тут же була викладена докладна інформація про те, як працюють браузери та сервери та як їх необхідно встановлювати, зовнішній вигляд першого сайту представлено на рис. 1 [12]

Для розробки сайту Бернерс-Лі підготував усі необхідні інструменти заздалегідь: гіпертекстовий браузер із функціоналом веб-редактора, веб-сторінки та сервер на базі NeXTcube. Сьогодні Тім Бернерс-Лі стоїть на чолі заснованого ним Консорціуму Всесвітньої мережі (World Wide Web Consortium), що займається розробкою та впровадженням стандартів Всесвітньої мережі. У середині 90-х розробка сайтів стала однією з галузей у світі, що найбільш динамічно розвиваються.

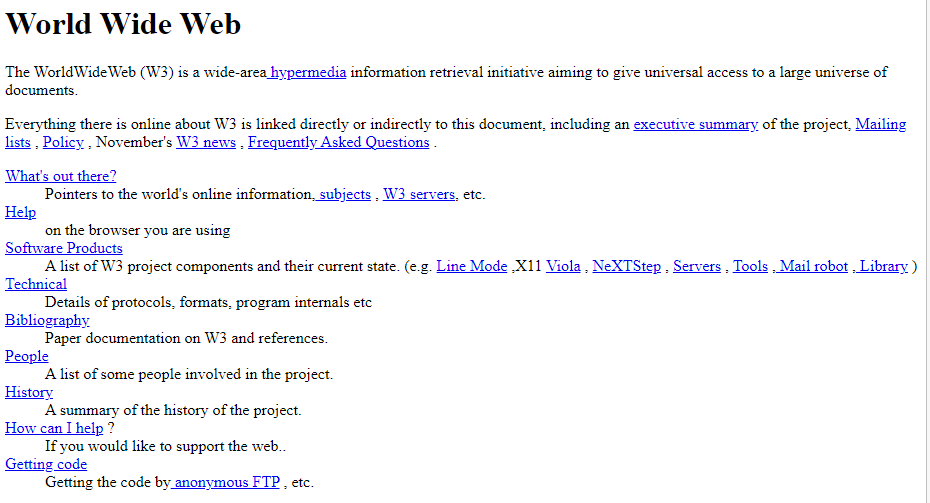


Рис.1. Перший веб-сайт http://info.cern.ch/ (http://info.cern.ch/hypertext/WWW/TheProject.html)

Згідно Wikipedia [15] «Вебсайт або сайт – сукупність веб сторінок та залежного вмісту, доступних у мережі Інтернет, які об'єднані як за змістом, так і за навігацією під єдиним доменним ім'ям. Фізично сайт може розміщуватися як на одному, так і на кількох серверах. Сайтом також називають вузол мережі Інтернет, комп'ютер, за яким закріплена унікальна ІР-адреса».

Класифікація сайтів не є єдиною, оскільки сайти можуть відрізнятися один від одного за різними параметрами. Експерти відносять сайти до різних категорій на основі характерних рис, таких як тематика, структура, дизайн, мета створення, але аналіз існуючої літератури [gni1, ivanova] дозволяє виділити наступні класифікації:

1. За типом контенту:

- Інформаційні сайти: надають інформацію про певну тему, продукт або послугу.

- Електронні магазини (інтернет-магазини): дозволяють користувачам купувати товари або послуги онлайн.

- Блоги: місце для публікації статей, думок.

- Форуми: дозволяють користувачам обмінюватися думками, ідеями або запитаннями.

- Соціальні мережі: платформи для спілкування та обміну інформацією між користувачами.

- Медіа-сайти: включають новинні портали, відеохостинг, аудіо-платформи тощо.

2. За цільовою аудиторією:

- Корпоративні сайти: представляють компанії або бізнеси та надають інформацію про їх продукти, послуги та контактну інформацію.

- Освітні сайти: присвячені навчанню, освіті або академічним установам.

- Урядові та некомерційні сайти: пов'язані з урядовими організаціями або неприбутковими організаціями.

- Особисті сайти: відображають інформацію про індивідуальну особу, її професійні досягнення, інтереси тощо.

- Сайти спеціального призначення: такі як сайти для подій, туристичні сайти, сайти для ресторанів тощо.

За функціональністю та складністю:

Статичні сайти: мають фіксований контент, який змінюється рідко.

Динамічні сайти: мають можливість взаємодії користувачів зі сторінками, наприклад, за допомогою коментарів, форм зворотного зв'язку тощо.

Веб-додатки: мають складні функції та можливості, такі як обробка оплати, управління контентом, системи керування вмістом (CMS) тощо.

Аналіз джерел [14], показує, що основні етапи створення веб-сайту включають наступні кроки:

1. Аналіз та планування:

- Визначення мети та цілей веб-сайту.

- Дослідження цільової аудиторії та її потреб.

- Вибір платформи для розробки (наприклад, CMS, фреймворки).

- Визначення структури веб-сайту та його навігації.

- Розробка контент-плану та визначення необхідного функціоналу.

2. Дизайн:

- Створення веб-дизайну, включаючи кольорову схему, шрифти, макети сторінок.

- Розробка адаптивного дизайну для оптимального відображення на різних пристроях (комп'ютери, планшети, мобільні телефони).

- Вибір та оптимізація зображень та мультимедійного контенту.

3. Розробка:

- Розробка фронтенду: створення HTML-шаблонів, CSS-стилів, розробка інтерфейсу користувача (UI) з використанням HTML, CSS та JavaScript.

- Розробка бекенду: створення серверної логіки, робота з базами даних, розробка функціоналу, який обробляє запити користувачів.

- Інтеграція сторонніх сервісів та додатків, якщо потрібно.

4. Контент:

- Створення та редагування текстового контенту для веб-сторінок.

- Оптимізація контенту для пошукових систем (SEO): використання ключових слів, мета-тегів, описів сторінок.

- Розробка та розміщення графічного контенту, включаючи зображення, відео, ілюстрації.

5. Тестування:

- Виконання функціонального тестування для перевірки роботи всіх функцій та компонентів веб-сайту.

- Перевірка сумісності з різними веб-браузерами, операційними системами та пристроями.

- Виявлення та виправлення помилок, забезпечення безпеки та надійності веб-сайту.

6. Розгортання та публікація:

- Вибір хостингу та розгортання веб-сайту на сервері.

- Конфігурація доменного імені та налаштування DNS.

- Перенесення веб-сайту з розробки до публічного доступу.

- Перевірка працездатності та налаштування аналітичних інструментів.

7. Підтримка та оновлення:

- Підтримка та відлагодження веб-сайту.

- Регулярне оновлення контенту, додавання нових функцій.

- Моніторинг продуктивності та безпеки веб-сайту.

- Взаємодія з користувачами та відповідь на їх запити.

Зауважимо, що це загальний огляд основних етапів створення веб-сайту. За необхідності, кожен етап може бути розширений або деталізований в залежності від конкретних вимог та потреб сайту.

Далі розглянемо основні мови, які використовуються при створенні веб сайтів.

## 1.2 Базові мови створення веб сайтів

На початку XXI сторіччя основою будь якого сайту є HTML (HyperText Markup Language — «мова гіпертекстової розмітки»). Він визначає зміст та структуру веб-контенту. Інші технології, крім HTML, зазвичай використовуються для опису зовнішнього вигляду/подання (CSS) або функціональності/поведінки (JavaScript) веб-сторінки. Мова HTML описується з допомогою стандарту[7].

Мова розмітки має багатолітню історію, і основні етапи її розвитку такі:

HTML 1.0 (1993): Перша версія HTML була простою мовою розмітки, яка дозволяла створювати гіпертекстові документи з посиланнями та заголовками.

HTML 2.0 (1995): Ця версія включала розширення, такі як підтримка таблиць і зображень, а також форми для взаємодії з користувачем.

HTML 3.2 (1997): Була додана підтримка таблиць більшого розміру і вирівнювання в тексті, а також підтримка каскадних таблиць стилів (CSS).

HTML 4.01 (1999): Ця версія включала багато нових можливостей, таких як фрейми (frames), форматування тексту, підтримка мови скриптів JavaScript і багато інших функцій.

XHTML (2000): XHTML (Extensible HTML) була перехідною версією між HTML 4 і HTML5. Вона була більш строгою версією HTML, яка використовувала синтаксис XML.

HTML5 (2014): HTML5 є найсучаснішою версією HTML і включає багато нових можливостей. Вона має покращену підтримку мультимедіа, графіки, геолокації, локального сховища даних та багато інших функцій.

HTML5 також включає підтримку багатьох API, таких як WebSocket API для взаємодії в режимі реального часу, Geolocation API для отримання географічних координат, Canvas API для рендерингу графіки на сторінці та багато інших.

HTML продовжує розвиватися, і нові версії та функції постійно додаються до стандарту.

Під гіпертекстом ("hypertext") розуміються посилання, які з'єднують веб-сторінки один з одним або в межах одного веб-сайту, або між веб-сайтами. Посилання є фундаментальним аспектом Інтернету. Завантажуючи контент в Інтернет і пов'язуючи його зі сторінками, створеними іншими людьми, ви стаєте активним учасником Всесвітньої мережі.

HTML не є мовою програмування; це мова розмітки, і використовується, щоб повідомляти ваш браузер, як відображати веб-сторінки, які ви відвідуєте. Він може бути складним чи простим, залежно від того, як хоче веб-дизайнер.

HTML використовує розмітку ("markup") для відображення тексту, зображень та іншого контенту у веб-браузері. HTML-розмітка включає спеціальні "елементи", такі як <head>, <title>, <body>, <footer>, <header>, ,<article>, <section>, <p>, <div>, <img>, <aside>, <audio>, <canvas>, <datalist>, <details>, <embed>, <nav>, <output>, <progress>, <video> та багато інших.

HTML-елемент виділяється з іншого тексту в документі за допомогою "тегів", які складаються з імені елемента оточеного "<" та ">". Ім'я елемента всередині тега не чутливе до регістру. Тобто воно може бути написане у верхньому або нижньому регістрі, або змішане. Наприклад, тег <title> може бути записаний як <Title>, <TITLE> або будь-яким іншим способом.

Кожен елемент містить наступні елементи дивиться на рис. 2.



Рис.2. Структура html тегу.

Відкриваючий тег (Opening tag): Складається з імені елемента, укладеного в кутові дужки, що відкривають і закривають. Відкриваючий тег вказує, де елемент починається чи починає діяти.

Закриваючий тег (Closing tag): Це те саме, що і тег, що відкриває, за винятком того, що він включає в себе косу межу перед ім'ям елемента. Закриваючий елемент показує, де елемент закінчується. Відсутність тега, що закриває, є однією з найпоширеніших помилок початківців і може призводити до дивних результатів.

Контент (Content): Це контент елемента, який у разі є просто текстом.

Елемент(Element): Відкриваючий тег, що закриває тег і контент разом складають елемент.

Елементи також можуть мати атрибути, які містять додаткову інформацію про елемент, яку не потрібно показувати у фактичному контенті. Наприклад, class це ім'я атрибута, а main-text – це значення атрибута. Клас дозволяє дати елементу ідентифікаційне ім'я, яке може пізніше використовуватися, щоб звертатися до елемента з інформацією про стиль та інші речі.

Атрибут завжди повинен мати: Пробіл між ним та ім'ям елемента (або попереднім атрибутом, якщо елемент вже має один або кілька атрибутів). Ім'я атрибута, за яким слідує знак рівності. Значення атрибута, укладене з двох сторін у лапки.

З допомогою HTML є можливість вставляти у документи текст, гіперпосилання, таблиці, зображення, відео та аудіо, форми, можливо навіть задавати простий дизайн безпосередньо в HTML - наприклад, встановлювати колір і шрифт тексту або колір блоку фону. Однак професіонали не рекомендують так робити – краще вказувати стилі у файлі CSS.

CSS (Cascading Style Sheets) – це код, який використовується для стилізації веб-сторінки. Як і HTML, CSS не є мовою програмування. Це не мова розмітки – це мова таблиці стилів. Це означає, що він дозволяє застосовувати стилі вибірково до елементів у документах HTML[2].

CSS1 (1996): Перша версія CSS вводила основні можливості для стилізації веб-сторінок, такі як зміна кольору, шрифту, розташування елементів та вирівнювання.

CSS2 (1998): Ця версія включала розширення, такі як підтримка плаваючих елементів, позиціонування, фонових зображень та багато інших можливостей для розмітки та стилізації.

CSS2.1 (2004): CSS2.1 була виправленою версією CSS2, яка включала усунення помилок та додаткові розширення, такі як підтримка класів і ідентифікаторів у селекторах.

CSS3: CSS3 є набором модулів, які додають нові можливості до CSS. Він включає модулі для анімації, градієнтів, тіней, трансформацій, медіазапитів, гнучкого розмітку та багато інших.

Із цими стандартами можливо ознайомитися в [2]

Крім офіційних версій CSS, були також випущені препроцесори CSS, такі як Sass, LESS і Stylus, які надають розширений синтаксис та додаткові функції для стилізації веб-сторінок.

CCS оперує особливими структурами, які називаються наборами правил (але найчастіше для стислості "правило"). Кожна з них складається з окремих частин:

Селектор (Selector) Ім'я HTML-елемента чи стилю, чи якийсь спеціальний символ на початку набору правил. Він вибирає елементи для застосування стилю.

Оголошення (Declaration) показує, які з властивостей елемента необхідно стилізувати.

Властивості (Properties) Способи, якими можна стилізувати певний HTML-елемент. Значення властивості (Property value) праворуч від властивості, після двокрапки, ми маємо значення властивості, яке вибирає одну з безлічі можливих ознак для даної властивості.

Повний огляд можливостей HTML та CSS не є предметом даної роботи, можливо ознайомитися із ними більше докладно, наприклад, у [8].

Третій стовп більшості сайтів JavaScript (часто просто JS) – це легковажна, інтерпретована або JIT-компілювана, об'єктно-орієнтована мова з функціями першого класу. Найширше застосування знаходить мову сценаріїв веб-сторінок, але також використовується і в інших програмних продуктах, наприклад node.js. JavaScript це прототипно-орієнтована, мультипарадигменна мова з динамічною типізацією, яка підтримує об'єктно-орієнтовані, імперативні та декларативні (наприклад, функціональне програмування) стилі програмування.

JavaScript була створена Бренданом Айком у 1995 році в рамках проєкту Netscape Navigator. Вона була спочатку названа Mocha, а пізніше перейменована на LiveScript і, нарешті, на JavaScript.

Стандартизація ECMAScript: У 1997 році JavaScript була стандартизована організацією Ecma International, і назва мови була змінена на ECMAScript. Перша стандартна версія називалася ECMAScript 1.

ECMAScript 3 (1999): Ця версія включала багато покращень, включаючи винятки, регулярні вирази, нові методи для роботи зі стрічками та масивами.

ECMAScript 5 (2009): В цій версії було додано багато нових можливостей, таких як об'єкти JSON, строгий режим (strict mode), методи масивів, методи для роботи з об'єктами, властивості і дескриптори.

ECMAScript 6 (ES6) або ECMAScript 2015: Ця версія включила значні зміни в мові, включаючи стрілкові функції, класи, блочні області видимості, розширену роботу зі стрічками та масивами, нові структури даних, генератори, модулі та імпорт/експорт функціональності.

ECMAScript 2016+: Після ES6 стали випускатися нові версії ECMAScript щорічно. Кожна версія включає нові можливості та покращення мови JavaScript. Деякі з них включають проміси (Promises), async/await, розпорядження ітераторів та інші функціональність.

JavaScript продовжує активно розвиватися, і нові функції та можливості додаються до стандарту ECMAScript.

Мова JS має велику кількість можливостей із якими можливо познайомитися наприклад в [6].

## 1.3 Основні технології розробки веб-сайтів

На основі вивчення джерел [16], [17] ми можемо вказати такі основні способи створення сайтів див. рис. 3

Найпоширеніший спосіб створення веб-сторінок з появою стандарту HTML - це ручна розробка за допомогою HTML. Основним інструментом для створення є прості текстові редактори, а також "code-based editor", які надають додаткові засоби для автоматичного створення коду (наприклад, Visual Studio Code, PhpStorm, Brackets та інші). Використання HTML, CSS та JavaScript є достатнім для створення статичних веб-сайтів. Однак, створення динамічних сайтів за допомогою цього методу дещо складніше, оскільки потрібно використовувати серверні скрипти, що вимагають додаткового програмного забезпечення для їх роботи. Незаперечною перевагою цього методу є те, що веб-розробники отримують саме той сайт, який планували, без зайвих тегів тому багато з них віддають перевагу цьому способу створення. Цей спосіб в сучасних умовах.

Рис. 3 Візуалізація основних способів створення веб-сайтів.

За допомогою програмних інструментів для розробки веб-сайтів, таких як візуальні редактори HTML-коду, спрощується процес розробки веб-сторінок, оскільки код сторінки може бути доволі об'ємним. В цих редакторах використовуються готові шаблони оформлення та можливість генерувати HTML-код. Існують два типи програм для редагування: "design-based editor", які містять візуальні редактори та автоматично генерують необхідний код і працюють у режимі WYSIWYG (наприклад, Adobe DreamWeaver, Microsoft FrontPage та інші).

Інструментальні системи (CMS) є готовими візуальними програмними оболонками, які можна заповнити контентом і, за необхідності, змінити їх налаштування. В сутності, вони є системами керування контентом. Цей метод створення є найбільш популярним, оскільки він зручний та практичний, дозволяючи економити час і зусилля, оскільки не потребує спеціальних навичок веб-програмування. Особливість роботи з CMS полягає в тому, що контент повністю відокремлений від візуального представлення веб-сайту, що спрощує процес редагування його вмісту. Найбільш популярними CMS є WordPress, Joomla, Drupal і т. д.

З використанням фреймворків створюються динамічні веб-сайти, в яких фреймворк виступає як основа або каркас. Він містить передбачені заздалегідь реалізовані класи та функції. Це означає, що до існуючого фреймворку можна додавати власний код для вирішення конкретних завдань. Використання фреймворків дозволяє уникнути необхідності писати повторний рутинний код і спрощує доступ до бази даних та розробку інтерфейсу, що допомагає уникнути дублювання коду. Найбільш популярними фреймворками в цій сфері є Lavarel, Uikit, Django, Tornado і т. д. Їх використання сприяє більш ефективному та швидкому розробленню веб-сайтів з використанням передових інструментів та стандартів.

Хмарні конструктори. На основі SaaS-платформ, що працюють у хмарному середовищі Cloud, існують спеціальні конструктори, які дозволяють користувачам збирати свої власні веб-сайти в онлайн-режимі за допомогою готових модулів та компонентів. Цей метод зазвичай застосовується для простих веб-проєктів з невеликими вимогами до розробників, оскільки він дозволяє швидко створити сайт без значних зусиль. Основною перевагою використання таких платформ є відсутність потреби в розробці проєкту, встановленні необхідних програм, налаштуванні хостингу та подальшому супроводі. Проте недоліком цього підходу є обмежений шаблонний дизайн та обмежені можливості розширення функціональності сайту. Найпопулярнішими SaaS-платформами в хмарному середовищі Cloud є UMI, WIX, InSales, Shopify, а також платформи для адміністрування сайтів, такі як sites.google.com та weedly.com. Використання цих платформ дозволяє швидко створювати та підтримувати веб-присутність без значних зусиль і спеціалізованих навичок. У таблиці 1 представлено порівняння.

Таблиця 1. Порівняльна характеристика різних способів створення сайтів

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Ручній/ CodeEditor | Ручній/ WYSIWYG | CMS | фреймворк | Конструктори |
| Складність розробки | складна | середня | легка | середня | легка |
| Функціональність | велика | велика | середня | велика | низька |
| Гнучкість | велика | мала | середня | велика | низька |
| Швидкість розробки | мала | середня | швидка | середня | швидка |
| Утримання та підтримка | складна | складна | середня | середня | середня |

Однією із головних по сукупності характеристик можливо обрати СМІ та фреймовокі. Розглянемо їх дещо докладніше.

Розробка веб-сайту з використанням фреймворку PHP або створення власної PHP-системи вимагає значно більшого часового і технічного зусилля.

Це надає можливість додати високорухомі функціональні можливості як для інтерфейсу користувача, так і для панелі управління.

Такий підхід також може допомогти прискорити робочий процес та створити абсолютно унікальні та амбітні веб-дизайні. Фреймворк також дозволяє уникнути деяких "стандартних" обмежень, характерних для систем управління контентом. Більш того, можливо вносити зміни до системи під час її роботи, що насправді забезпечує адаптабельність вашого сайту до зростання вашого бізнесу. Фреймворки майже не мають обмежень у цьому відношенні.

У деяких випадках системи управління контентом, такі як WordPress, також відомі своїми потенційними ризиками безпеки, що, в певній мірі, є наслідком їх популярності. Використання фреймворків PHP, з іншого боку, потенційно забезпечує більшу рівень безпеки, хоча, звичайно, це залежить від якості програмного кодування.

Опублікування в системі управління контентом (CMS) не вимагає втручання веб-розробника та не потребує глибоких знань HTML, PHP, JavaScript або CSS. Крім управління вмістом, система управління контентом також вирішує ряд важливих завдань, включаючи забезпечення конфіденційності даних, допомогу у виконанні адміністративних функцій, форматування подання даних у зручний вигляд та SEO-оптимізацію.

Варіант з використанням фреймворків підходить для тих, хто володіє програмувальними навичками, тоді як стандартний варіант системи управління контентом (CMS) "з коробки" надає можливість людям, які далекі від веб-розробки, використовувати широкий спектр інструментів. За допомогою стандартних CMS можна, наприклад, швидко розгорнути інтернет-магазин на хостингу або створити соціальну мережу, або ж створити базу знань у форматі wiki всього за кілька хвилин.

## 1.4 Порівняльна характеристика основних CMS на мові PHP

У 2023 році на ринку існує багато CMS, які створені на мові PHP одні із найбільш популярних є Joomla!, WordPress, Drupal, Magento [1]

На сьогоднішній день WordPress має репутацію найбільш використовуваної у світі CMS з відкритим кодом для веб-розробки. Ви можете створювати багато додатків на будь-якому хостингу PHP за допомогою WordPress. Незалежно від того, чи потрібен вам простий блог чи магазин електронної комерції, у WordPress є всі рішення для вас. Він має найбільшу частку ринку з приблизно 75 мільйонами активних веб-сайтів в Інтернеті.

WordPress є платформою з відкритим кодом і дозволяє тисячам розробників з усього світу робити свій внесок у її вдосконалення. Ось чому платформа так швидко розвивалася протягом багатьох років. Він став найкращим вибором для всіх видів веб-додатків. Платформа дала початок багатьом новаторським ідеям у сфері розробки. Він має найбільший запас додаткових плагінів, тем і віджетів, який можна легко налаштувати для будь-яких бажаних результатів.

плюси

Зручна платформа; Найбільше сховище плагінів/тем; Надійно захищена платформа; Вбудований механізм SEO

мінуси

Оновлення тем і плагінів WordPress надходять часто; Трохи вразливий для хакерів через відкритий вихідний код; Потрібні знання для налаштування

Magento

Magento, створений спеціально лише для додатків електронної комерції, є найпопулярнішим вибором для розробки просунутих онлайн-магазинів. Багато провідних брендів, зокрема Ford, Nike, Foxconnect та багато інших, використовують Magento для своїх онлайн-операцій. Ці бренди значною мірою покладаються на Magento через його оптимізовані та вдосконалені функції електронної комерції. Основна перевага використання Magento полягає в тому, що він дає змогу легко розробляти індивідуальні програми електронної комерції.

Magento також відомий своїми бездоганними функціями безпеки. Платформа посилена за допомогою алгоритмів хешування для безпечного керування паролями та має кілька інших функцій для захисту програм від зловмисних атак. Крім того, Magento має широку спільноту розробників, які регулярно вносять свій внесок у різні виправлення та оновлення на платформі. Випуск Magento 2 приніс багато вдосконалень платформі та зробив її найкращим варіантом для створення великого роздрібного онлайн-магазину.

Плюси

Платформа має багато функцій, які допомагають розвивати модернізовані магазини електронної комерції; Magento надзвичайно зручний і простий у роботі; Він має широку спільноту користувачів, які регулярно роблять внесок у розширення платформи та плагіни; Платформа має високу масштабованість і створена для підтримки масштабованості великомасштабних програм

мінуси

Що стосується корпоративної та преміальної версій, платформа є дещо дорожчою; Magento має трохи повільніший час завантаження порівняно з іншими платформами; Він підтримує лише виділений хостинг; Підтримка продукту досить дорога

Drupal

Drupal — це безкоштовна CMS з відкритим кодом, написана на PHP, яка вважається однією з найкращих платформ для магазинів електронної комерції. Спочатку він був розроблений як рішення для студентської спільноти, але пізніше перетворився на повноцінну PHP CMS з відкритим кодом. Drupal надає розробникам легкість створювати оптимізовані онлайн-магазини з досконалістю. Оскільки платформа захоплює розширеними можливостями сучасної розробки додатків, і тому більшість розробників віддають перевагу їй.

WordPress використовує плагіни, Drupal пропонує модулі для виконання налаштованих функцій на веб-сайтах. Крім того, платформа має широкий набір функцій та інших опцій, які роблять її однією з найкращих PHP CMS на ринку розробки програм. Багато великих організацій у світі, включаючи NBC, Гарвардський університет, Tesla, Princess Cruises, MTV UK та інші, використовують Drupal для своїх веб-операцій. Платформа також підтримується широким колом спільнот, які постійно зростають швидкими темпами по всьому світу.

плюси

Платформа дуже розширювана; Регулярні оновлення та виправлення покращують безпеку платформи; Drupal — це адаптована платформа для порталів електронної комерції; Найкраща CMS для веб-сайтів із високим трафіком

мінуси

Оскільки Drupal є зручною для розробників платформою, звичайним користувачам досить важко зрозуміти технічні особливості платформи; Платформа не рекомендована для блогів/публікацій; Вибіркова установка програм і модулів досить складна.

Joomla

Joomla — ще одна CMS на основі PHP, відома завдяки створенню блогів і портфоліо веб-сайтів. Хоча її порівняння з WordPress не зовсім справедливе, платформа пропонує адекватні функції для створення першокласного блогу та динамічних веб-сайтів. Платформа підтримує основні бази даних, включаючи MySQL, MS-SQL і PostgreSQL, надаючи розробникам можливість легко інтегрувати свої сайти в бажані системи баз даних. Це потужна платформа, яка дозволяє розробникам створювати власні веб-сайти з розширеними функціями.

CMS дозволяє легко інтегрувати ваш веб-сайт із веб-хостингом Joomla лише за один клік. Крім того, за допомогою цієї платформи досить легко створювати веб-сайти, адаптовані до мобільних пристроїв. Він також пропонує тисячі розширень і дизайнів, які ви можете використовувати у своїх веб-програмах і можете покращити його функціональність. Саме тому платформа входить до топ-5 найбільш використовуваних CMS у світі та використовується провідними брендами, включаючи eBay, Barnes & Noble, IKEA та багато інших.

плюси

Простий для розуміння інтерфейс користувача; Забезпечує чудове налаштування інтерфейсу; Платформа досить безпечна завдяки регулярним оновленням; Ідеально підходить для створення блогів і динамічних веб-додатків

мінуси

Платформа менш сприятлива для SEO порівняно з іншими; Платформа є досить складною, коли йдеться про включення нестандартного дизайну без досвіду розробки; Має лише кілька модулів, доступних на ринку; Якщо не змінити код, деякі плагіни мають проблеми з сумісністю.

Виходячи із вимог до сайту та його процесу, найдоцільніше обрати CMS Drupal.

## Висновки до розділу 1

Веб-сайт – це електронна сторінка або колекція сторінок, які доступні через Інтернет і містять різноманітну інформацію, таку як текст, зображення, відео, аудіо та інші елементи. Веб-сайти використовуються для різних цілей, таких як надання інформації, продаж товарів і послуг, комунікація зі спільнотою тощо.

Розробка веб-сайтів включає кілька етапів. Перш ніж розпочати створення веб-сайту, необхідно провести аналіз вимог, визначити цілі та цільову аудиторію. Далі, на етапі планування, визначається структура веб-сайту, його дизайн та технології, що будуть використовуватися.

На етапі розробки контенту створюється текстовий матеріал, зображення, відео та інші елементи, які будуть розміщені на веб-сайті. Потім веб-сайт розробляється та програмується, використовуючи різні мови програмування та технології.

Після завершення розробки проводиться тестування веб-сайту, щоб переконатися, що він працює належним чином, є сумісним з різними браузерами та пристроями, і не містить помилок.

Останнім етапом є розгортання веб-сайту на веб-сервері та його публікація в Інтернеті. Після розгортання веб-сайту необхідно забезпечити його підтримку, оновлювати контент та виправляти помилки, які можуть виникнути в процесі експлуатації.

Розглянуто основні способи створення сайтів (ручній на основі Code, ручній на основі WYSIWYG, CMS, фреймворк, конструктор), а також основні CMS на мові PHP (Joomla!, WordPress, Drupal, Magento). Та визначено, що сайт буде реалізовано з допомогою CMS Drupal.

# РОЗДІЛ 2 ОГЛЯД CMS DRUPAL

## 2.1 Порівняльна характеристика Drupal 8, Drupal 9, Drupal 10

Давайте розглянемо ключові відмінності між Drupal 8, 9 і 10, базуючись на [4] та [10].

Drupal 8 був серйозним оновленням платформи, зосередженим на покращенні взаємодії з користувачем і полегшенні створення адаптивних і зручних для мобільних пристроїв веб-сайтів. Деякі ключові вдосконалення в Drupal 8 включали нову систему тем, покращену підтримку адаптивного дизайну та кращу інтеграцію зі сторонніми бібліотеками та інструментами. Крім того, Drupal 8 представив більш модульну архітектуру, що спрощує розширення та налаштування платформи для задоволення конкретних потреб бізнесу.

Drupal 9 був випущений у червні 2020 року, і його основна увага була спрямована на оновлення базової інфраструктури платформи. Це означало, що багато ключових покращень у Drupal 9 були «під капотом», такі як оновлення до останніх версій Symfony, Twig та інших залежностей. Однак Drupal 9 також представив деякі важливі нові функції, такі як краща підтримка керування медіафайлами та покращена доступність.

Станом на травень 2023 остання версія Drupal – це Drupal 10, яка була випущена в грудні 2022 року. Основна мета Drupal 10 – продовжити роботу, розпочату в Drupal 9, з акцентом на оновленні базової інфраструктури платформи. Однак Drupal 10 також представить важливі нові функції, такі як краща підтримка безголових і роз’єднаних архітектур, покращена продуктивність і покращений досвід розробника.

Однією з найбільших залежностей Drupal 9 є Symfony 4, термін експлуатації якого закінчується в листопаді 2023 року. Це означає, що помилки безпеки в Symfony 4 не будуть виправлені після листопада 2023 року. Drupal 10 оновив Symfony з версії 4 до версії 6.

Drupal 10 використовує CKEditor5 із значно покращеним досвідом створення та набором плагінів.

Оновлення ядра Drupal протягом багатьох років було кошмаром для новачків і розробників сайтів. З Drupal 9 набагато легше оновлювати ядро Drupal через композитор. З Drupal 10 це відбувається автоматично.

Узагальнимо інформацію щодо різних версій Drupal у таблиці 2.

Таблиця 2 Відомості про версії 8, 9, 10

|  |  |  |  |
| --- | --- | --- | --- |
| Ознака | Drupal 8 | Drupal 9 | Drupal 10 |
| Дата випуску | Листопад 2015 | Червень 2020 | Червень 2022 |
| Підтримується до | Листопад 2021 | Листопад 2023 | Листопад 2025 |
| Версія PHP | PHP 7.0 | PHP 7.3 (Drupal 9.5.3 вимагає PHP 8.1) | PHP 8.1 або вище |
| Конструктор макетів | Доступні | Доступні | Доступні |
| JS | JQuery | JQuery | Поліпшені JS компоненти |
| Медіа-менеджмент | Потрібні додані модулі | Потрібні додані модулі | Доступні |
| Покращена міграція | Так | Так | Так |
| Оновлення на основі композитора | Необов'язково | Необов'язково | Необхідний |

Вибір 10 версії обснований тим, що це досить стабільна версія, яка буде підтримуватися до 2025 року та містить важливі оновлення.

## 2.2 Основні концепції CMS Drupal

Як ми вже говорили раніше, Система керування вмістом (CMS) — це програмний інструмент, який дозволяє користувачам додавати, публікувати, редагувати або видаляти вміст із веб-сайту за допомогою веб-браузера на смартфоні, планшеті чи настільному комп’ютері. Як правило, програмне забезпечення CMS написано на мові сценаріїв, а його сценарії виконуються на комп’ютері, де встановлено базу даних і веб-сервер. Вміст і налаштування для веб-сайту зазвичай зберігаються в базі даних, і для кожного запиту сторінки, який надходить на веб-сервер, сценарії поєднують інформацію з бази даних і активів (файли JavaScript, файли CSS, файли зображень тощо, які є частиною CMS або були завантажені) для створення сторінок веб-сайту.

Відома комбінація операційної системи, на якій працює CMS, мови сценаріїв, на якій вона написана, бази даних, у якій зберігається інформація, і веб-сервера, який запускає сценарії для отримання інформації та повернення її у веб-браузер відвідувача сайту. як стек, на якому працює CMS

Поширена комбінація операційної системи Linux, веб-сервера Apache, бази даних MySQL і мови сценаріїв PHP відома як стек LAMP.

Drupal — це гнучка CMS, заснована на стеку LAMP, із модульною конструкцією, що дозволяє додавати та видаляти функції, встановлюючи та видаляючи модулі, а також дозволяє змінювати зовнішній вигляд веб-сайту шляхом встановлення та видалення тем. Базове завантаження Drupal, відоме як Drupal Core, містить PHP-скрипти, необхідні для запуску основних функцій CMS, кілька додаткових модулів і тем, а також багато ресурсів JavaScript, CSS і зображень. Багато додаткових модулів і тем можна завантажити з веб-сайту https://www.drupal.org/project/project\_module [5]

Будь який розробник та навіть звичайний користувач Drupal повинен усвідомлювати наступні концепції

Модуль

Модуль — це набір файлів PHP, JavaScript і/або CSS, який розширює можливості сайту та додає функціональність.

Можливо ввімкнути функції та функції, встановивши модуль, і вимкнути їх, видаливши модуль; перед видаленням може знадобитися видалити дані та конфігурацію, пов’язані з цією функцією чи функціями.

Основне завантаження містить модулі для таких функцій, як:

Керування обліковими записами користувачів (основний модуль користувача User)

Керування базовим вмістом (основний модуль Node) і полями (основні модулі Field і Field UI)

Керування навігаційними меню (основний модуль інтерфейсу користувача меню)

Кожен встановлений модуль збільшує час, необхідний для створення сторінок на вашому сайті, тому непотрібні модулі доцільно видалити.

Теми

Тема — це набір файлів, які визначають візуальний вигляд і відчуття сайту. Основне програмне забезпечення та модулі, які працюють на сайті, визначають, який вміст (включно з текстом HTML та іншими даними, що зберігаються в базі даних, завантаженими зображеннями та будь-якими іншими файлами ресурсів) буде відображатися на сторінках сайту. Тема визначає розмітку HTML і стиль CSS, який обгортає вміст.

Ядро Drupal містить кілька базових тем із основним дистрибутивом. Ці теми в основному були розроблені та створені спільнотою протягом останніх кількох років, і всі вони є гарним вибором для створення перших сайтів і ближчого ознайомлення з тим, як працює основне програмне забезпечення.[9]

Drupal – добре зарекомендувала себе CMS, тому ринок сторонніх тем – як безкоштовних, так і платних – дуже потужний.

Якщо жоден із варіантів сторонніх розробників не відповідає потребам, потрібно буде створити спеціальну тему. Спеціальна тема може бути такою ж простою, як один файл CSS, який додає стиль до розмітки, наданої основним програмним забезпеченням.

Більш докладно про теми буде надано інформацію у подальших розділах.

Проєкт Drupal

Проєкт Drupal — це проєкт FOSS, метою якого є розробка основного програмного забезпечення системи керування вмістом, а також додаткових модулів, додаткових тем, перекладів, документації та дистрибутивів спеціального призначення. Люди, які вкладають свій час і гроші в проєкт Drupal, приїжджають з усього світу та є різноманітною спільнотою, яка об’єднується заради цієї спільної мети.

Спільнота охоплює багато невеликих груп, які виконують багато різних завдань, таких як розробка певної частини програмного забезпечення, пов’язаного з Drupal, написання документації, підтримка безпеки програмного забезпечення Drupal, переклад програмного забезпечення Drupal на певну мову, використання Drupal для певної мети та разом, щоб зустрітися особисто в певному географічному регіоні.

Типи даних

Дані та інформація на сайті поділяються на чотири типи, які редагуються, перекладаються та зберігаються по-різному. Ці чотири типи:

Вміст (Content)

Інформація (текст, зображення), призначена для відображення відвідувачам сайту. Цей тип інформації, як правило, є відносно постійним, але зазвичай його можна редагувати.

Конфігурація (Configuration)

Інформація про сайт, яка не є вмістом, але є відносно постійною та використовується для визначення того, як сайт поводиться або відображається. Іноді він також відображається для відвідувачів сайту, але зазвичай це менші частини тексту (наприклад, мітки полів, назва сайту тощо), а не великі фрагменти, які зазвичай вважаються Вмістом.

Стан (State)

Інформація тимчасового характеру про поточний стан сайту, наприклад, час останнього виконання завдань cron.

Сесія (Session)

Інформація про взаємодію окремих відвідувачів сайту із сайтом, наприклад, чи ввійшли вони в систему, і їхні файли cookie. Технічно це підтип станової інформації, оскільки вона також є тимчасовою.

Дуже пов’язанні із контентом сутність вмісту (або частіше сутність) — це елемент даних вмісту, який може складатися з тексту, розмітки HTML, зображень, вкладених файлів та інших даних, призначених для відображення відвідувачам сайту. Сутності вмісту можуть бути визначені основним програмним забезпеченням або модулями.[11]

Сутності вмісту згруповані в типи сутностей, які мають різні цілі та дуже різними способами відображаються на сайті. Більшість типів сутностей також поділяються на підтипи сутностей, які є підрозділами всередині типу сутності, щоб дозволити менші варіанти використання та відображення сутностей.

Наприклад сутність контенту визначено в модулі Node використовується для вмісту призначеного для основної сторінки для сторінок сайту. Інший приклад коментарі, це сутність яка додається до сутностей контенту і працює з допомогою модуля Comment. Також заслуговують уваги таки сутності: профіль користувача, термін таксономії, файл.

У межах сутностей дані зберігаються в окремих полях, кожне з яких містить один тип даних, наприклад форматований або простий текст, зображення чи інші файли або дати. Типи полів можуть бути визначені основним програмним забезпеченням або модулями.

Адміністратор може додавати поля до підтипів сутностей, щоб усі сутності даного підтипу сутності мали однакову колекцію доступних полів. Наприклад, тип контенту «Мобільний телефон» може мати поля для назви бренду, зображення логотипу, URL-адреси веб-сайту та кількості камер, тоді як тип вмісту сторінки «Новина» може мати поля лише для заголовка та основної частини сторінки. Коли користувач створює або редагує сутності, він вказує значення для полів сутності.

Поле посилання – це поле, яке представляє зв’язок між сутністю та однією або декількома іншими сутностями, які можуть належати до того самого або іншого типу сутності. Три найбільш часто використовувані поля посилання:

Посилання на вміст

Посилання на елемент вмісту. Наприклад, можливо зв’язати мобільні з постачальниками, які їх надіслали, для цього потрібно налаштувати поле посилання на вміст під назвою «Надіслано», посилаючись на елементи вмісту «Постачальник» у типі вмісту «Мобільний».

Посилання на терміни таксономії

Посилання на таксономічний термін. Наприклад, можливо пов’язати рецепти з їхніми інгредієнтами.

Посилання на користувача

Посилання на обліковий запис користувача. Наприклад, можливо зв’язати фотографії із їх авторами.

Більш докладно розглянемо другий тип поля посилання таксономія використовується для класифікації вмісту веб-сайту. Одним із поширених прикладів таксономії є теги, які використовуються для класифікації чи категоризації публікацій на веб-сайті блогу; веб-сайт ринку фотографії може використовувати таксономію фотографій для їх класифікації. Окремі сутності таксономії відомі як терміни (теги блогу); а набір термінів відомий як словник (набір усіх тегів публікацій блогу). Технічно терміни таксономії є типом сутності, а підтипи сутності є словниками. Як і інші сутності, терміни таксономії можуть мати приєднані поля; наприклад, є можливість налаштувати поле зображення, яке містить піктограму для кожного терміна.

Окремий словник може організовувати свої терміни в ієрархію або він може бути плоским. Наприклад, теги блогу зазвичай мають плоску структуру, тоді як словник типів фотографій може бути ієрархічним.

Терміни таксономії зазвичай додаються як довідкові поля до інших об’єктів вмісту, і саме так їх можна використовувати для класифікації вмісту. Коли користувач налаштовує довідкове поле таксономії, він може дозволити користувачам вводити терміни двома способами:

Вільне тегування

Нові умови можна створювати прямо у формі редагування вмісту.

Фіксований список термінів

Список термінів підбирається та керується поза формою редагування вмісту, і користувачі можуть вибирати з наявного списку лише під час редагування вмісту.

Довідкові поля таксономії можна додати до будь-якої сутності, наприклад до облікових записів користувачів, спеціальних блоків або звичайних елементів вмісту, на сайті буде автоматично налаштовано сторінки з таксономічним списком для кожного терміна.

Спосіб відображення сутності (наприклад, вмісту, користувача чи коментаря) залежить від контексту, у якому вона відображається. Цей контекст відомий як режим перегляду. Приклади режимів перегляду:

повна сторінка вмісту з усіма значеннями полів (full page)

тизер вмісту з мініатюрою та посиланням «Докладніше» (teaser)

повний профіль користувача з масштабованою портретною фотографією

аватар користувача з іменем користувача та посиланням на профіль

Кожен режим перегляду можна налаштувати на сторінках адміністрування для цього потрібно перейти у адміністративному меню «Керування» до «Структура» > «Типи вмісту» (адміністрування/структура/типи) і клацнути «Керувати відображенням» (Manage display) у розкривному меню. У кожному режимі перегляду всі поля можна приховати або відобразити, а якщо вони відображаються, можливо вибрати та налаштувати форматування полів.

Форматування полів — це налаштування для відображення значень полів. Наприклад, довгі текстові поля можуть відображатися обрізаними або на повну довжину, а поля посилань на терміни таксономії можуть відображатися у вигляді звичайного тексту або бути пов’язані зі сторінкою термінів таксономії.

Перегляд — це перелік вмісту веб-сайту. Основний модуль Views керує відображенням переглядів, а основний модуль Views UI дозволяє створювати та редагувати їх в адміністративному інтерфейсі. Коли ви визначаєте перегляди, ви зацікавлені в тому, щоб отримати дані з вашого веб-сайту та показати їх користувачеві.

Список, створений представленням даних, може мати будь-яку з таких форм:

* Таблиця з полями, які можна сортувати
* Сіткові макети
* Тизери або картинки, які посилаються на статті
* Блоки
* JSON
* RSS-канали
* Календарі
* Слайд-шоу на екрані

Наостанок дуже стисло розглянемо можливості Drupal, щодо обмеження прав користувачів і керування доступом.

Кожен, хто відвідує веб-сайт, є його користувачем. Існує три групи користувачів:

Користувачі, які не ввійшли в систему, або анонімні користувачі

Користувачі, які ввійшли в систему, або автентифіковані користувачі

Обліковий запис користувача-адміністратора, який було автоматично створено під час встановлення вашого сайту, або Користувач 1.

Можливість виконувати дії на сайті (включаючи перегляд вмісту, редагування вмісту та зміну конфігурації) регулюється дозволами. Кожен дозвіл має назву (наприклад, «Перегляд опублікованого вмісту») і охоплює одну дію або невеликий набір дій. Щоб виконати відповідну дію на сайті, користувач повинен отримати дозвіл; дозволи визначаються модулями, які забезпечують дії.

Замість того, щоб призначати окремі дозволи безпосередньо кожному користувачеві, дозволи групуються в ролі. Можливо визначити одну або кілька ролей на сайті, а потім надати дозволи кожній ролі. Дозволи, надані автентифікованим і анонімним користувачам, містяться в ролях Автентифікований користувач і Анонімний користувач, і залежно від профілю інсталяції, який використовували під час інсталяції сайту, також може існувати роль адміністратора, якій автоматично призначаються всі дозволи на сайті.

Кожному обліковому запису користувача на сайті автоматично призначається роль автентифікованого користувача, і за бажанням йому може бути призначена одна або кілька додаткових ролей. Коли призначаємо роль обліковому запису користувача, користувач матиме всі дозволи ролі після входу в систему.

Хорошою практикою є створення кількох ролей на сайті.

## 2.3 Темізація в CMS Drupal

Окрім основного вмісту, веб-сторінка містить інший вміст, такий як брендинг сайту (назва сайту, слоган і логотип), засоби навігації (меню, посилання та значки), форматований текст і зображення. Кожна тема надає набір іменованих областей, таких як заголовок, вміст і бічна панель, де розробники сайтів можуть розмістити свій вміст. Усі ці регіони за традицією називають англійськими іменами.[3]

Нижче представлено регіони теми за замовченням Olivero дивиться рис. 4

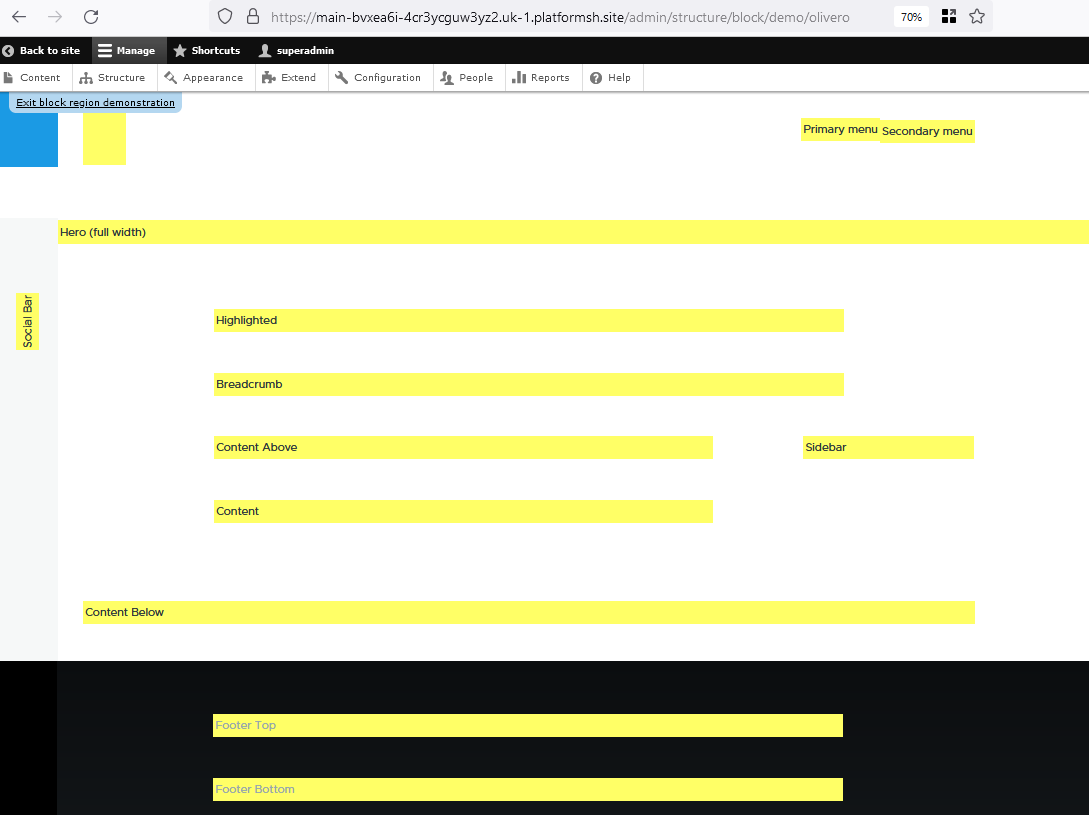


Рис. 4 Регіони теми Olivero

Доступні регіони залежать від дизайну теми. Потрібна лише область вмісту, яка містить основний вміст; інші необов'язкові.

Блоки — це окремі частини макета веб-сторінки сайту. Вони розміщуються в областях теми, і їх можна створювати, видаляти та змінювати на сторінці адміністрування макета блоку (адмін/структура/блок). Приклади блоків включають список Хто онлайн, головне меню навігації та навігаційний шлях. Вміст головної сторінки також є блоком.

Деякі модулі роблять нові блоки доступними для розміщення на сайті. Наприклад, коли основний модуль пошуку встановлено та налаштовано, він надає блок, який містить форму пошуку. Ви також можете створювати та розміщувати власні блоки.

Кожен блок має власні параметри конфігурації, які дозволяють вибрати, на яких сторінках вашого сайту буде відображатися блок. Можна навіть розмістити кілька копій блоку, кожна зі своєю окремою конфігурацією та правилами видимості.

Drupal використовує наступний принцип після того, як дані будуть отримані та оброблені всередині модуля (бізнес-логіка), йому потрібно буде забезпечити реалізацію теми за замовчуванням, щоб загорнути її в розмітку. Іноді конкретна тема потребує заміни цієї реалізації, щоб досягти певної мети дизайну. Якщо тема забезпечує власну реалізацію, Drupal використовуватиме цю реалізацію теми замість стандартної реалізації модуля. Зазвичай це називається перевизначенням. В іншому випадку резервний варіант за замовчуванням все ще буде присутній. Тема також надає можливість застосовувати стилі лише за допомогою CSS і залишати без змін розмітку, надану модулем.

Механізми створення тем відповідають за фактичний вихід через файли шаблонів. Починаючи з Drupal 8, це обробляється двигуном Twig, створеним SensioLabs (люди, відповідальні за проєкт Symfony). Отже, усі файли шаблонів у Drupal мають розширення .html.twig.

Хуки тем визначають, як має відтворюватися конкретна частина даних. Вони реєструються в системі тем за модулями (і темами) за допомогою hooktheme(). При цьому вони отримують ім’я, список змінних, які вони виводять (дані, які потрібно обернути розміткою), та інші параметри.

Модулі та теми, які реєструють перехоплення тем, також повинні забезпечити реалізацію (таку, яка буде використовуватися за замовчуванням).

На додаток до фактичної реалізації, модулі та теми, які реєструють перехоплення теми, також можуть надавати стандартний препроцесор шаблону. Відповідальністю цього є «попередня обробка» (тобто підготовка) даних перед надсиланням до шаблону. Наприклад, якщо хук теми отримує сутність (складний об’єкт даних) як свою єдину змінну, препроцесор може бути використаний, щоб розбити цю сутність на крихітні фрагменти, які необхідно вивести в шаблон (наприклад, заголовок і опис).

Препроцесори шаблонів — це прості процедурні функції, які слідують шаблону іменування та викликаються системою теми перед відтворенням шаблону. Як я згадував раніше, модулі та теми, які реєструють хук теми, також можуть надавати препроцесор за замовчуванням. Отже, для хука теми під назвою componentbox функція препроцесора за замовчуванням виглядатиме так: templatepreprocesscomponentbox(&$variables)

Щоб створити тему для Drupal 8 або новішої версії, вам потрібно спочатку створити файл THEMENAME.info.yml, який надає метадані про вашу тему для Drupal. Це подібно до того, як визначаються модулі та профілі інсталяції, тому важливо встановити ключ «тип» у файлі file.info.yml на «тему», щоб відрізнити його.

Створіть файл .info.yml у корені папки тем. Ім’я папки – це ім’я комп’ютера, а ім’я інформаційного файлу – ім’я\_машини.info.yml. Для решти цієї сторінки ми створимо тему із зрозумілою для людини назвою «Fluffiness», а назвою машини — «fluffiness», тому папку буде названо «fluffiness/», а файл .info.yml — «fluffiness». /fluffiness.info.yml". Якщо файл містить мінімально необхідні властивості (ім’я, тип, базова тема та вимоги до ядра/версії ядра), ваша тема буде видима на вашому веб-сайті в розділі «Керування» > «Вигляд» (http://example.com/admin/appearance).

Із самим форматом YAML можливо знайти в [13].

Основними ключовими словами є

name (обов'язково)

Ім'я, зрозуміле людині. Це з’явиться на сторінці «Вигляд», де активована тема.

type (обов'язково)

Вказує тип розширення, тобто «модуль», «тема» або «профіль». Для тем завжди має бути встановлено значення "тема". Це значення чутливе до регістру.

base theme (обов'язково)

Тема може успадкувати ресурси від іншої теми, вказавши її як базову тему.

core\_version\_requirement (обов'язково)

Дозволяє модулям, темам і профілям вказувати, що вони сумісні з кількома основними версіями ядра Drupal.

Розробник повинен розмістити теми в папці «themes» інсталяції Drupal. Зауважимо, що основні теми Drupal, такі як Bartik і Seven, розташовані в папці core/themes встановлення див. рис. 5

Рекомендується розміщувати надані теми у підпапці під назвою «contrib», а власні теми — у папці під назвою «custom».

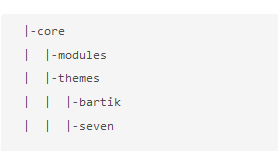


Рис. 5 Стуктура тек для розміщення тем

На рис. 6 представлена типова структура файлів у теці теми.



Рис. 6 Структура теці

\*.info.yml

Тема повинна містити файл .info.yml для визначення теми. Серед іншого файли .info.yml визначають метадані, бібліотеки та блокові області. Це єдиний необхідний файл у темі.

\*.libraries.yml

Файл .libraries.yml використовується для визначення бібліотек JavaScript і CSS, які може завантажувати тема.

\*.breakpoints.yml

Точки зупину визначають, де адаптивний дизайн має змінитися, щоб реагувати на різні пристрої. Точки зупину визначаються у файлі .breakpoints.yml.

\*.themes

Файл .theme – це файл PHP, який містить усю умовну логіку та (попередню) обробку виведених даних. Він також може розширити основні налаштування теми.

css/

Рекомендовано зберігати файли .css у підпапці «css».

js/

Спеціальні файли JavaScript для теми зберігаються в папці «js». Щоб тема завантажувала файли JavaScript, вони повинні бути визначені у файлі .libraries.yml.

screenshot.png

Якщо файл screenshot.png знайдено в папці теми, він використовуватиметься на сторінці «Вигляд». Крім того, можливо визначити знімок екрана у файлі .info.yml.

logo.svg

Якщо векторний файл SVG із логотипом теми знайдено в папці теми, його можна використовувати в заголовку веб-сайту.

Drupal дозволяє замінити всі шаблони, які використовуються для створення розмітки HTML, щоб ви могли повністю контролювати розмітку, яка відображається як результат у користувацькій темі. Існують шаблони для кожного елемента сторінки, починаючи від високорівневого HTML і закінчуючи невеликими полями.

## Висновки до розділу 2

Проведений порівняльний аналіз версій Drupal 8, 9, 10.

Drupal 10 є останньою версією популярної відкритої платформи управління контентом (CMS) станом на час написання цієї роботи. З поліпшеними заходами безпеки, оптимізацією продуктивності та інструментами для зручності використання, Drupal 10 надає бізнесам усі необхідні переваги для створення веб-сайтів високої якості, які залучають своїх користувачів.

Основними концепціями є: модуль, тема, проєкт, типи даних, сутність, таксономія, поле, перегляд, користувачі, дозволи, ролі.

Розглянуто теоретичні основи темізаціїї Drupal.

# РОЗГЛЯД 3 РОЗРОБКА САЙТУ «МОБІЛОГРАФІЯ ДЛЯ ВСІХ»

## 3.1 Налаштування сайту на CMS Drupal

Налаштування будь якого сайту починається із конфігурування робочої середи, як вже говорилось у підрозділі 2.2 для праці Drupal на сервері потрібно мати LAMP або аналогічний йому по функціональності стек.

Для створення веб сайту обрано platform.sh на рис. 7 представлено початок реєстрації на цій платформі, а додатку А представлено опис основних кроків.

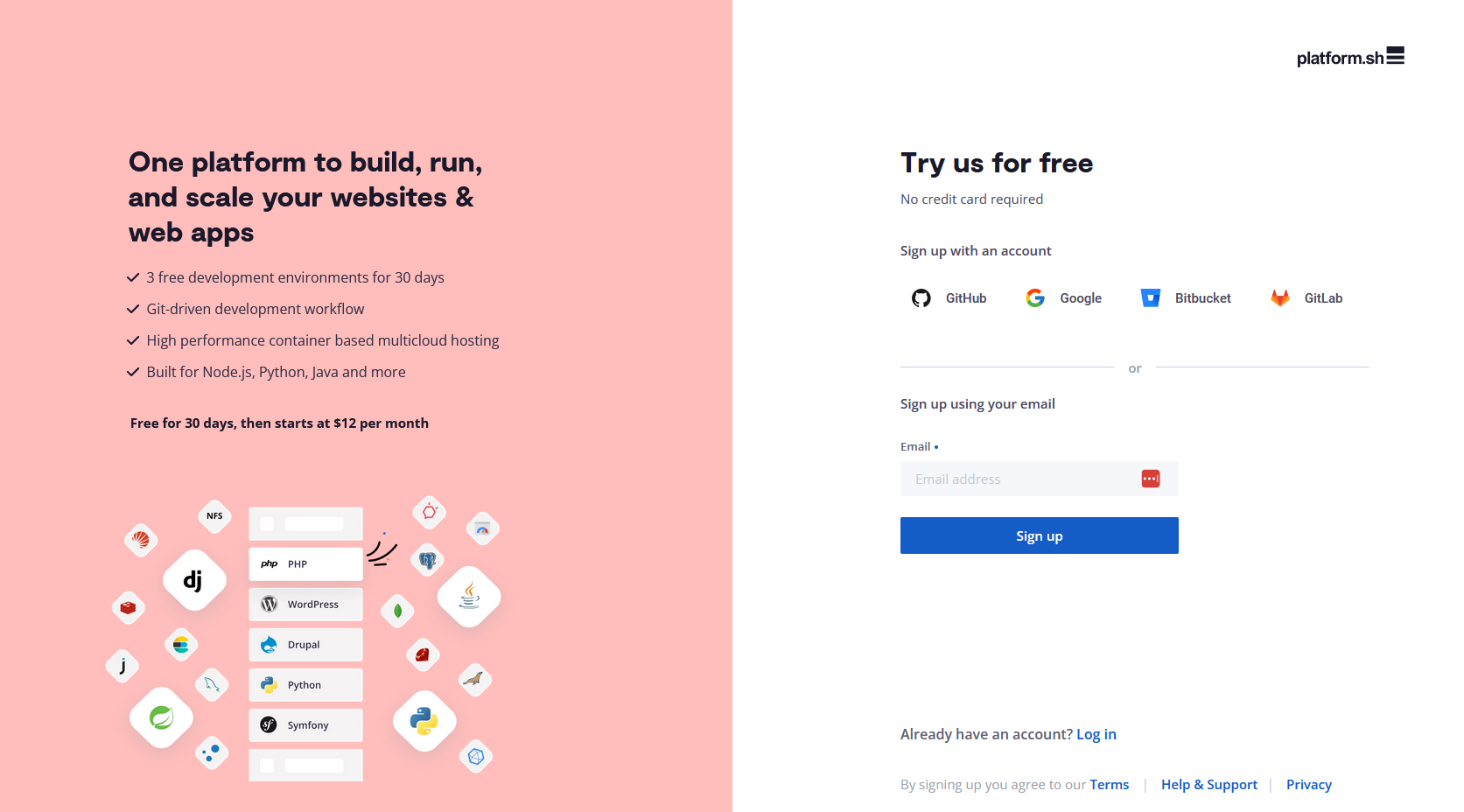


Рис. 7 Початок реєстрації на platform.sh

Далі безпосереднє встановлення Drupal 10 відбувається із допомогою Composer – менеджера пакетів для PHP. Він відстежує версії та залежності ядра Drupal, модулі, теми та бібліотеки і гарантує, що користувач не встановить щось, що є не сумісним. На додаток до ядра drupal установки, кожен модуль, або тема може додавати вимоги та залежності з допомогою файлу composer.json. Основні команди composer вказані у додатку Б.

На обраній платформі встановлення Drupal відбувається автоматично. Якщо б обрали локальне встановлення, то потрібно використати composer.

Далі дуже важливим є перший захід на сайт, з початку обирається мова дивиться рис. 8.

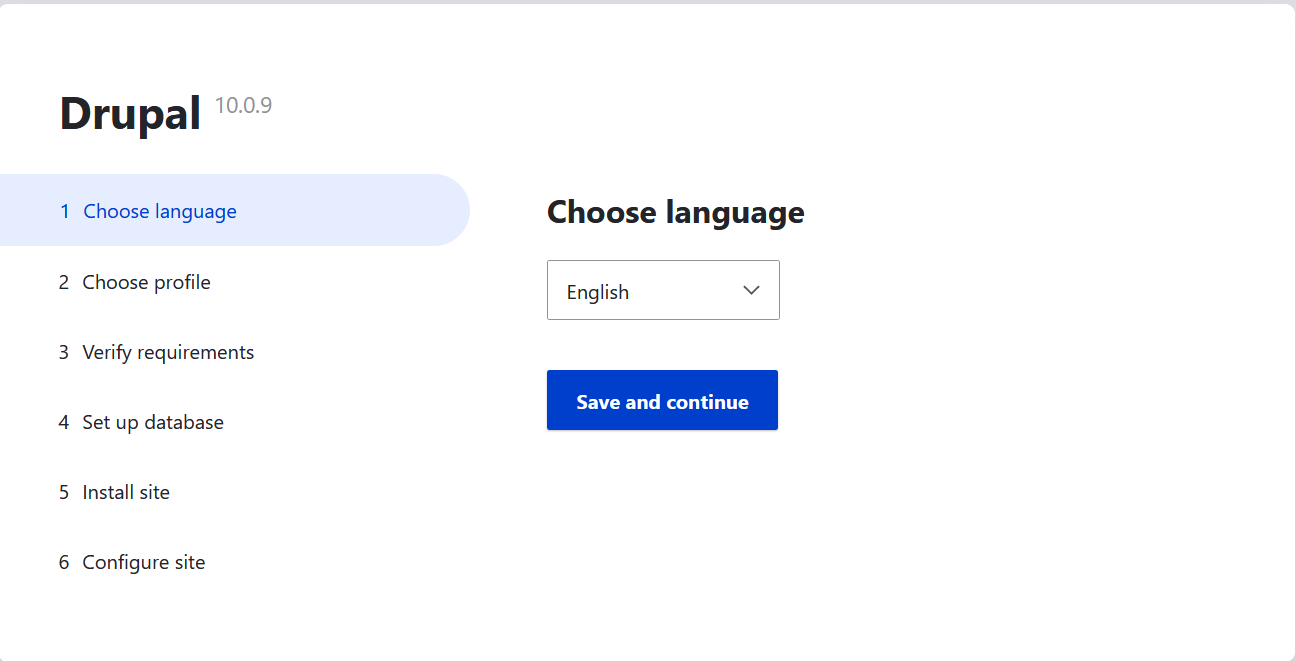


Рис. 8 Початок створення сайту крок Вибір мови

Далі обираємо тип профайлу.

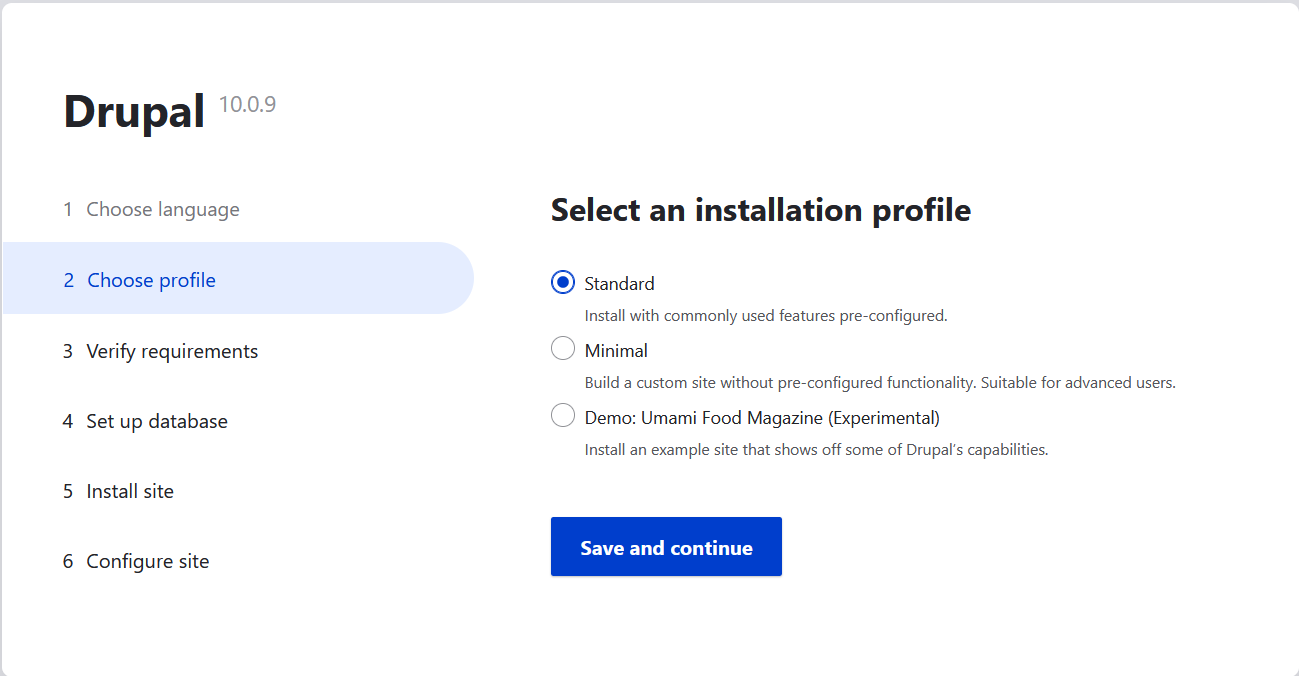


Рис. 9 Початок створення сайту крок Вибір профайлу

На наступних кроках відбувається перевірка вимог, налаштування бази даних завершується створенням першого користувача див. рис. 10, 11

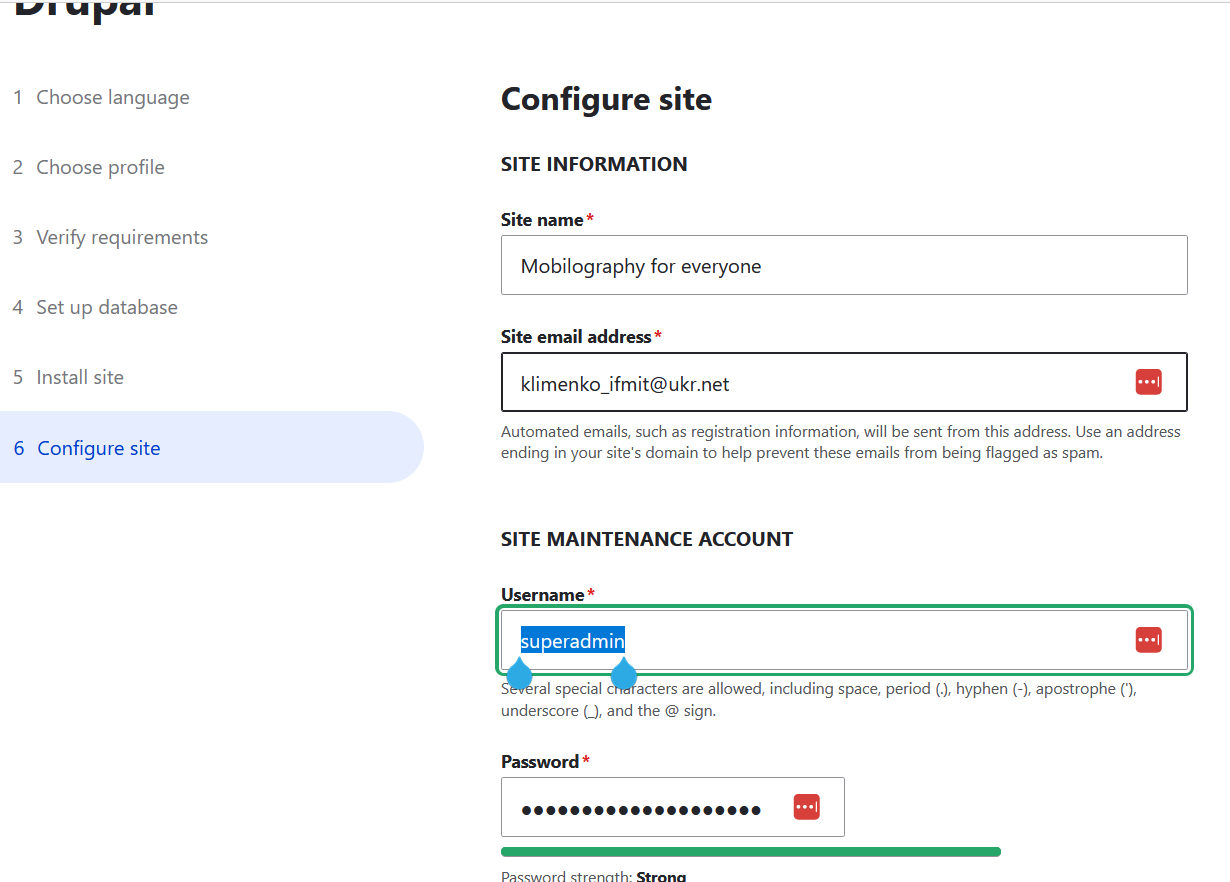


Рис. 10 Завершення створення сайту крок «Налаштування сайту 1»

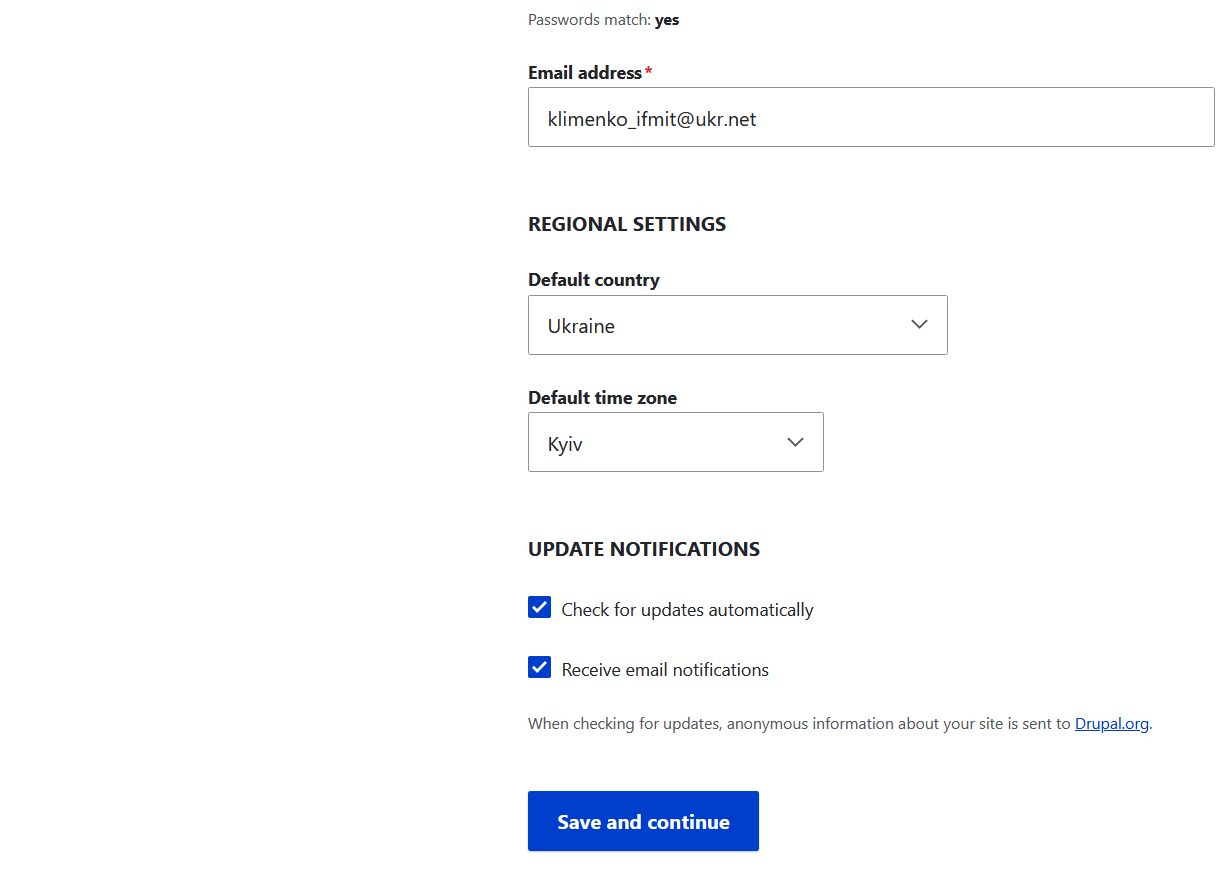


Рис. 11 Завершення створення сайту крок «Налаштування сайту 2»

По завершенню встановлення ми переходимо на головну сторінку сайту.

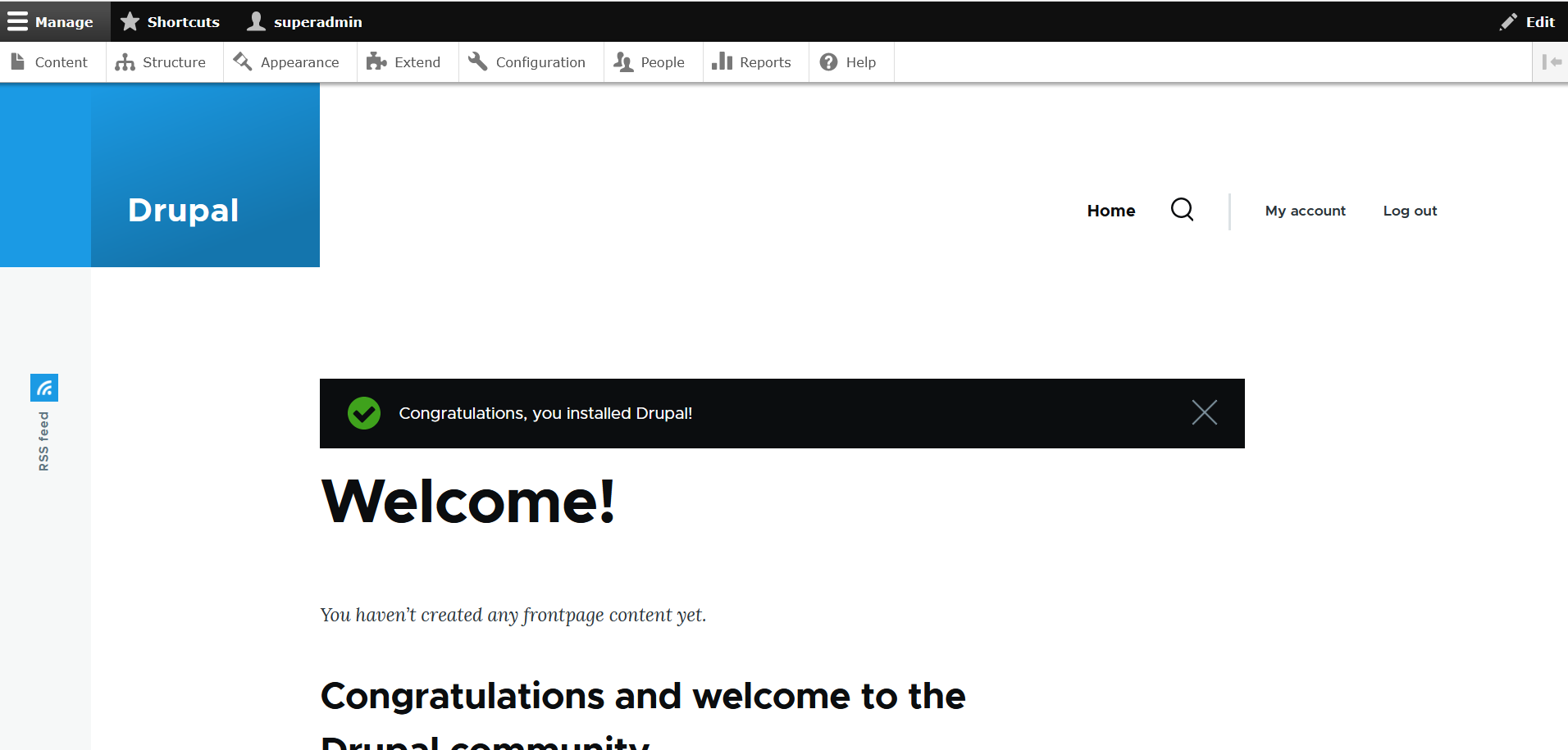


Рис. 12 Вигляд тільки що встановленого Drupal

Вимоги до сайту включають мультимовність (українська, англійська), тому включення модулів Language, Interface Translation, Content Translation, Configuration Translation потрібно зробити найшвидшим чином.

Для цього переходимо на /admin/modules див. рис. 13

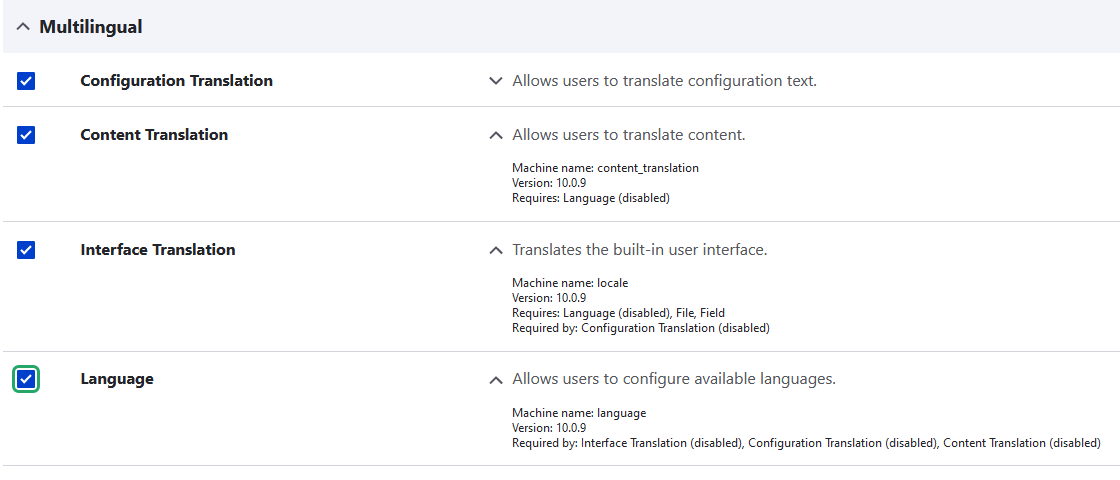


Рис. 13 Активізація мультимовних модулів

Після встановлення модулів потрібно перейти за шляхом /admin/config/regional/language див. рис. 14 та встановити нову мову, імпорт перекладу відбудеться автоматичним чином.

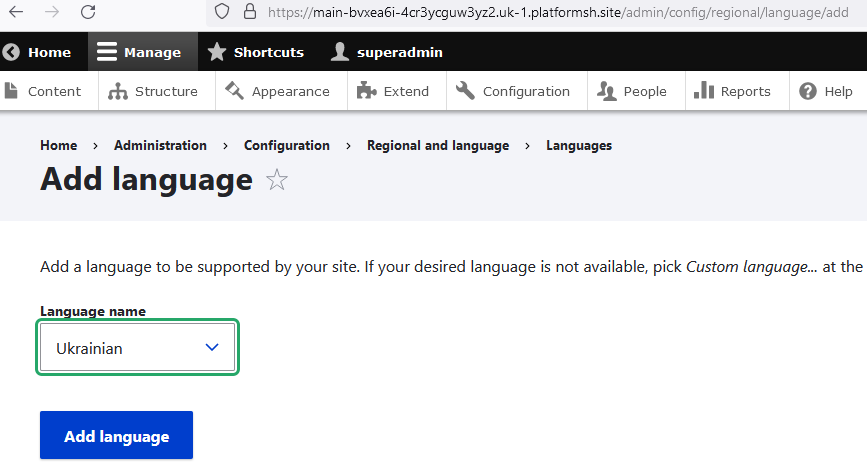


Рис. 14 Додавання мови

Також потрібно додати кастомний блок Language switcher перехід за шляхом /admin/structure/block сам процес представлено в рис. 15, 16

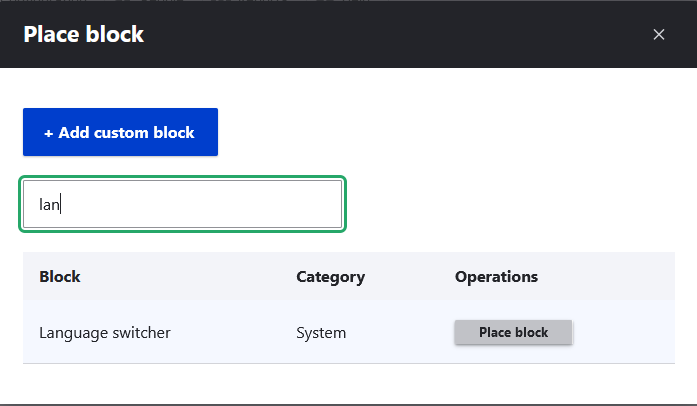


Рис. 15 Додавання блоку перемикач мов

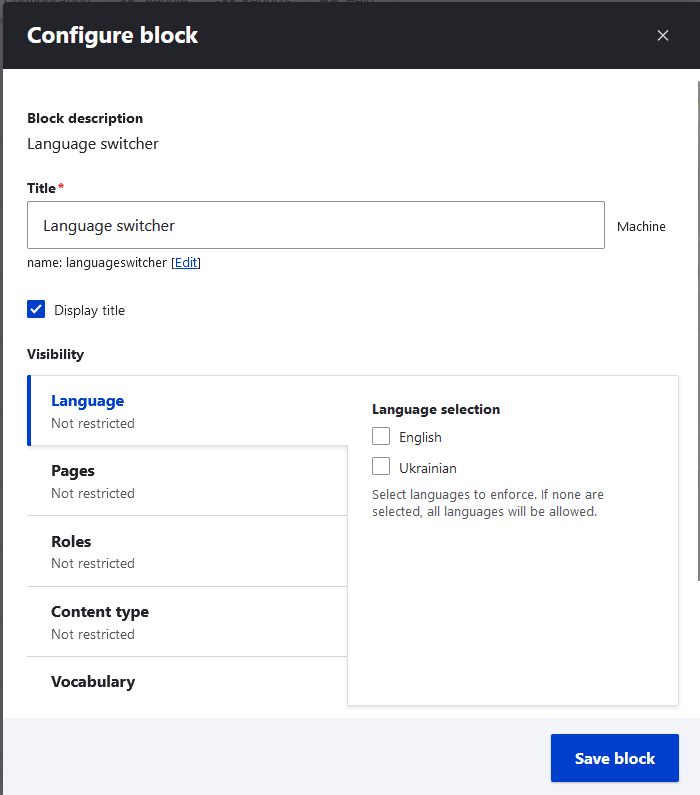


Рис. 16 Налаштування блоку перемикач мов

Також потрібно вказати, які типи матеріалів можливо перекладати за шляхом admin/config/regional/content-language

Далі налаштовуємо основні конфігурації переходимо на admin/config/system/site-information і там встановлюємо ім’я сайту, слоган та емайл рис. 17, а також встановлюємо переклад.

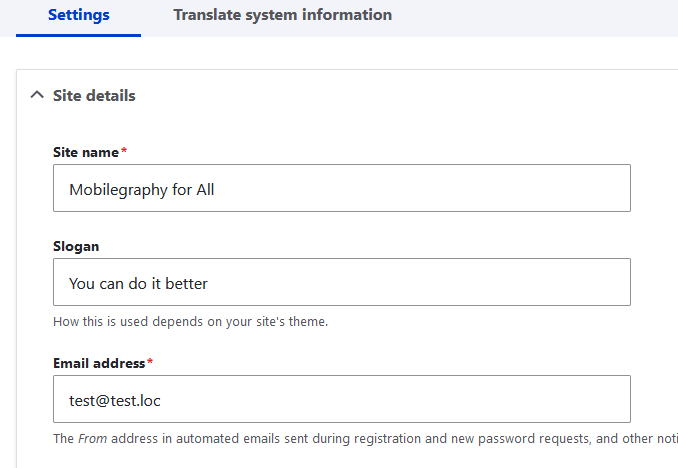


Рис. 17 Налаштування ім’я сайту та слогану

Далі переходимо до опису створення контенту сайту

## 3.2 Створення основного контенту сайту

З початку створімо сторінку сайту «Що таке мобілографія», для цього переходимо по шляху admin/content та клацаємо кнопку «Add Content» та обираємо тип «Page» рис. 18 обираємо Menu Settings і вказуємо, що посилання на цю сторінку будуть в головному меню, також задаємо URL alias /about

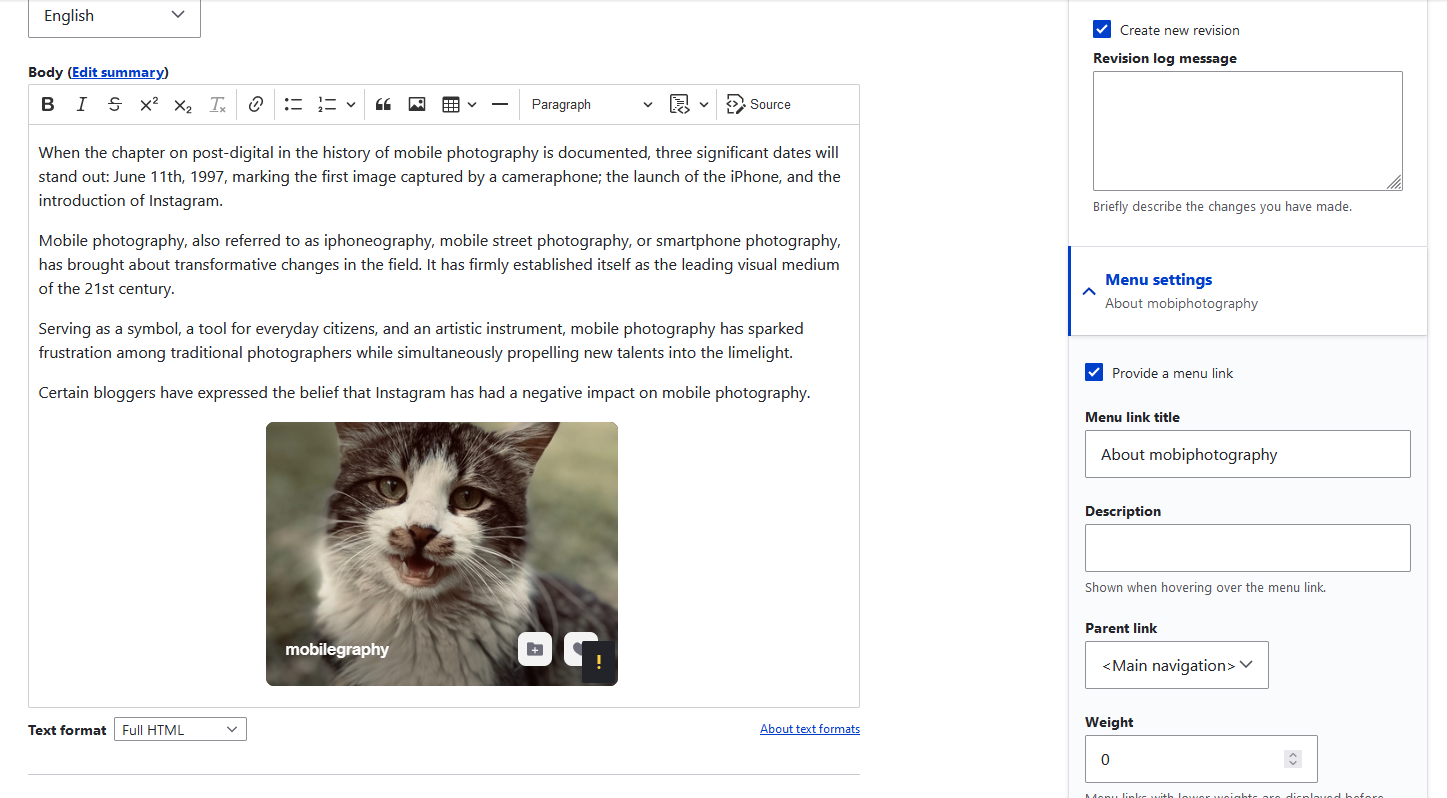


Рис. 18 Створення сторінки про мобілографію

Після створення додаємо переклад використовуємо посилання /node/1/translations/add/en/uk

Згідно вимог до сайту на ньому потрібно відобразити обладнання різних типів, яке потрібно для мобіографії, яке буває різних видів, для цієї задачі найбільш усього підходить тахномія. Розпочинаємо із створення словника термінів для цього переходимо Structure > Taxonomy (admin/structure/taxonomy) та додаємо новий словник (+Add vocabulary) далі потрібно додати ім’я словника, його опис, вказати, що його будуть перекладати дивиться рис. 19

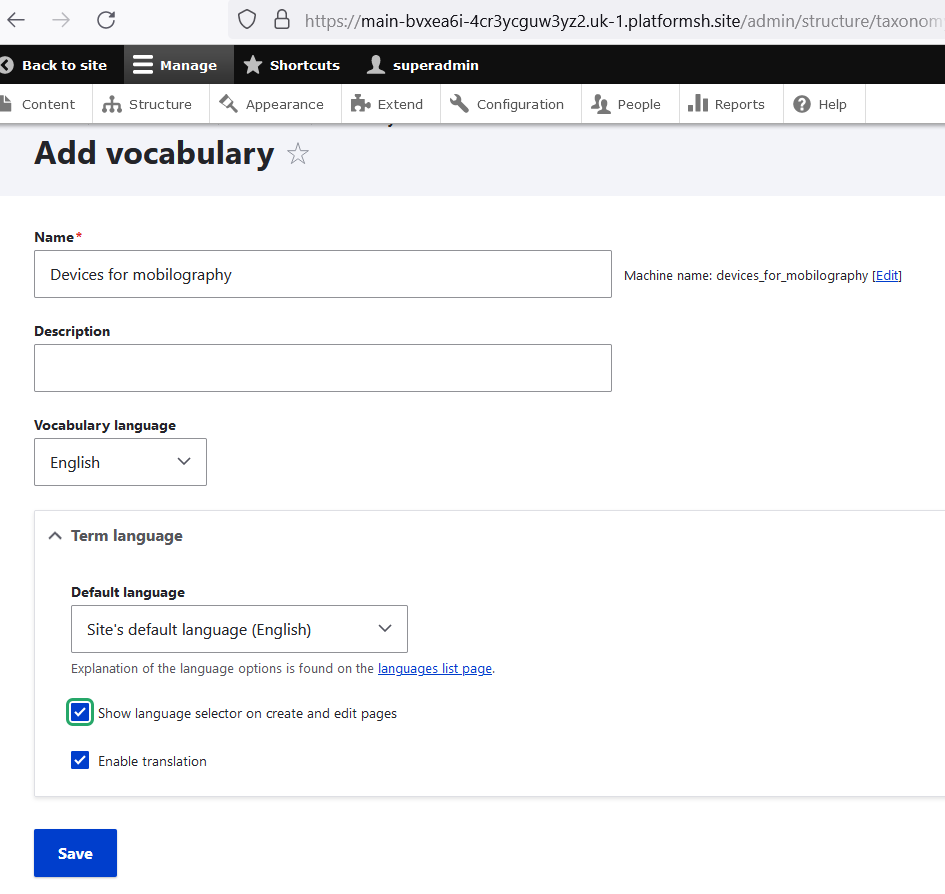


Рис. 19 Створення словника для різних типів пристроїв

Далі додаємо нові терміни в цей словник, та вказуємо взаємовідносини між ними рис. 20 використовуємо шлях admin/structure/taxonomy/manage/devices\_for\_mobilography/add

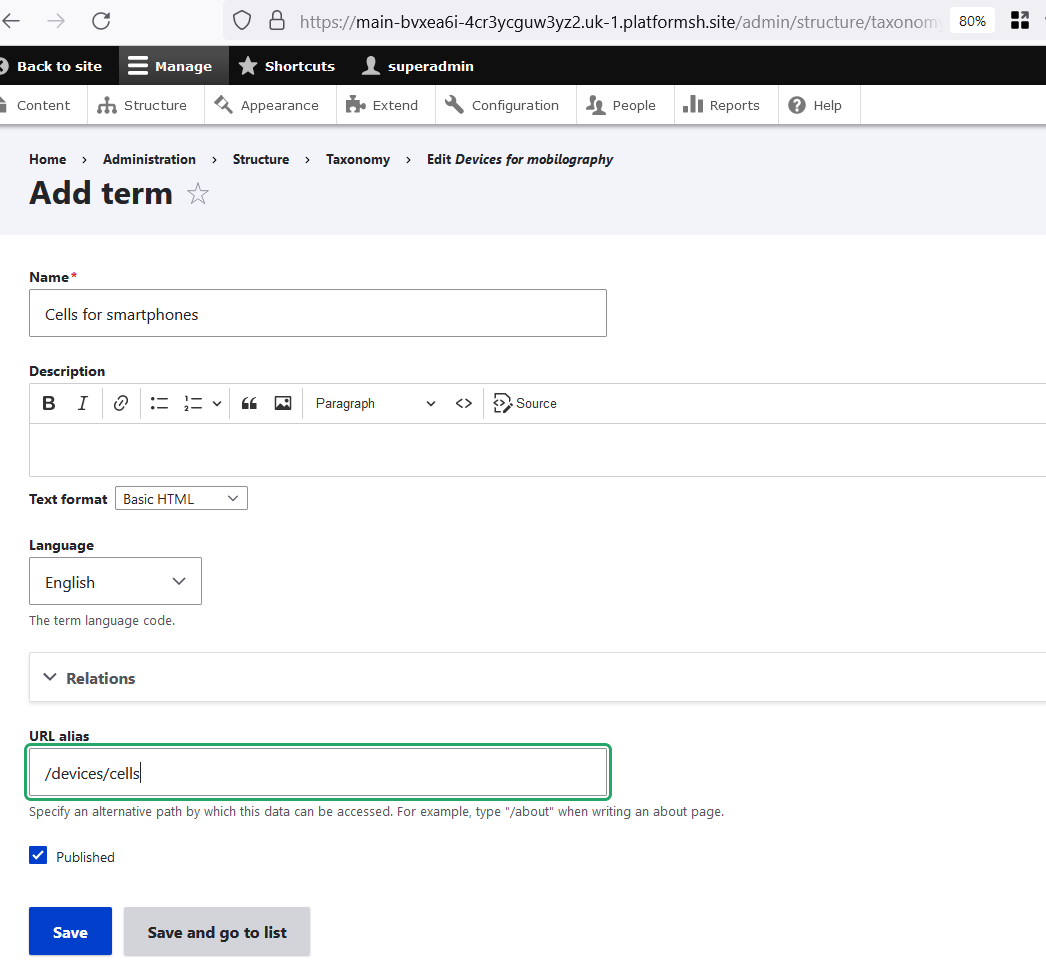


Рис. 20 Створення терміну

Аналогічним чином робили для інших термінів, та їх перекладів.

Далі переходимо до створення типу контента для зберігання інформації про пристрої, із вимог отримаємо, що кожний пристрій має: назву, ціну, опис, характеристики, фото, та тип. переходимо Structure > Content types (admin/structure/types) там є можливість подивитися на перелік існуючих типів контенту додаємо новий контент (+Add content type) потім переходимо на сторінку (дивись рисунок 21) де можливо вказати ім’я типу контенту, опції публікації (чи з’явиться матеріал на головній сторінці, чи буде матеріал попереду усіх списків), також можливо вказувати настроювання мов (мова за замовченням, чи можливо перекладати) та інші параметри.

Далі ми переходимо до додавання полів до нашого типу, за замовченням кожен матеріал має Title, це поле будемо використовувати для назви та Body це поле буде використовувати для опису.

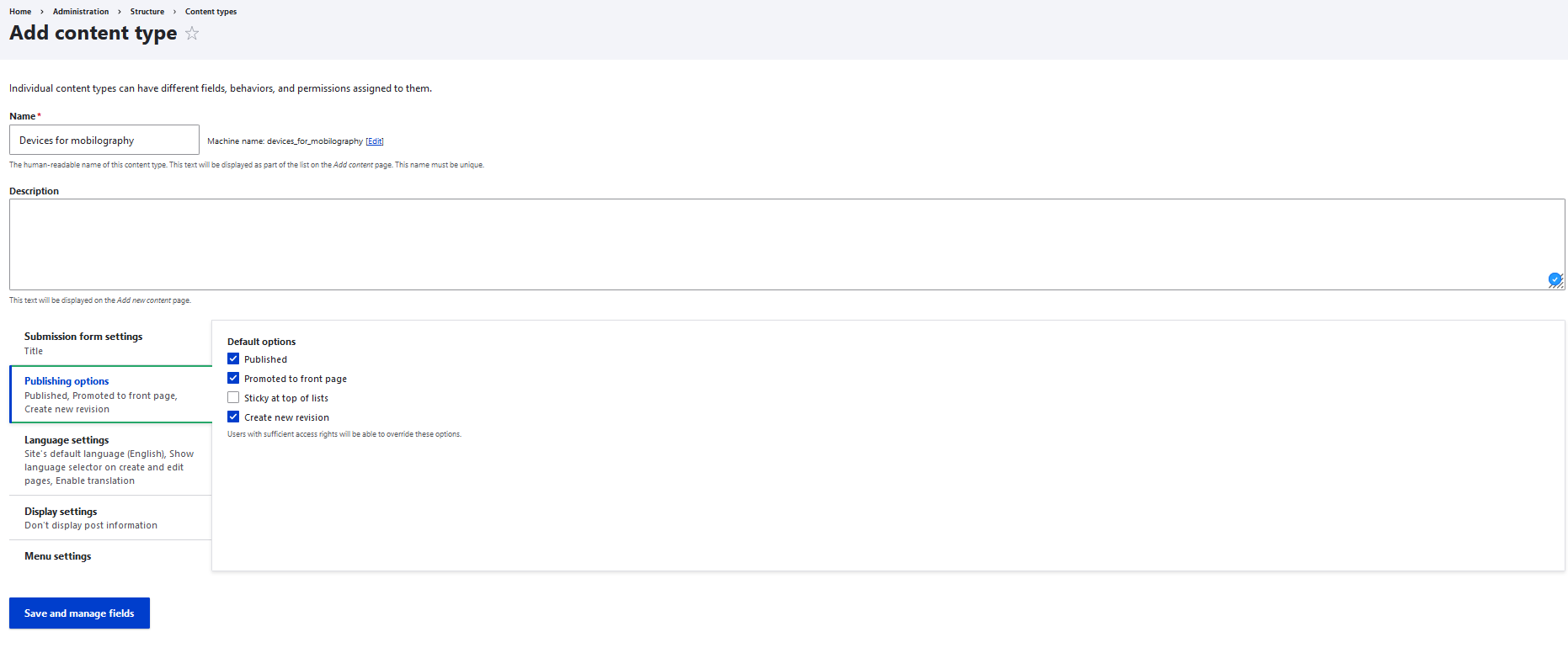


Рис. 20 Створення нового типу контенту

У таблиці 3 представимо назву полів та їх тип в Drupal

Таблиця 3 Назви полів та їх типи для типу Пристрої для мобілографії

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Назва | Name | Plaine text |
| Ціна | Price | Number (decimal) |
| Опис | Description | Text |
| Характеристики | Characteristics | Text |
| Фото | Photo | Media |
| тип | type | refrence |

На рисунку 21 представлено перший крок додавання поля Назва до типу «Devices for mobiligraphy» на рис. 22 представлено які властивості можливо задати, наприклад, довжина не більше 240 символів, і що значення може бути лише одне.

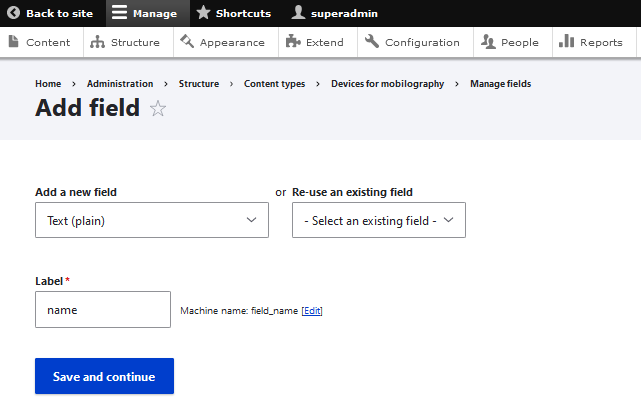


Рис. 21 Додавання поля до типу контенту крок 1

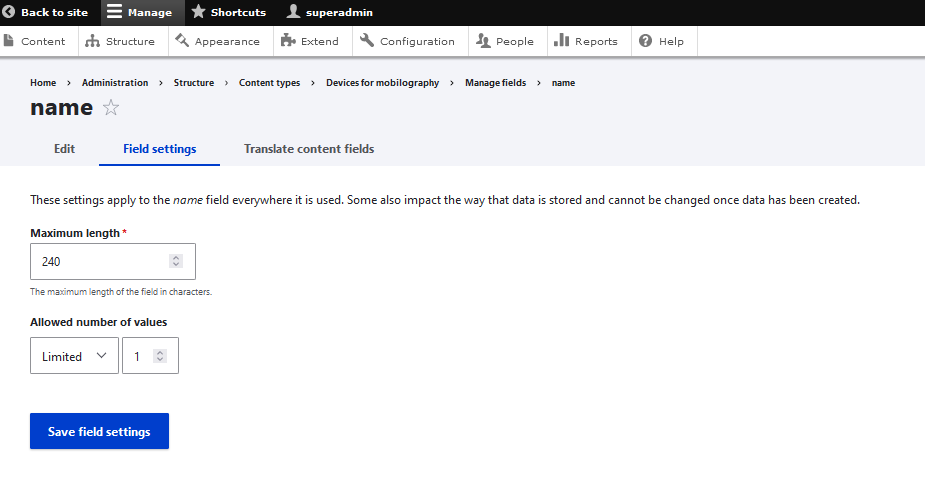


Рис. 22 Додавання поля до типу контенту крок 2

На останньому кроці додавання поля, можливо вказати чи є воно обов’язковим, та налаштування перекладу див. рис. 23

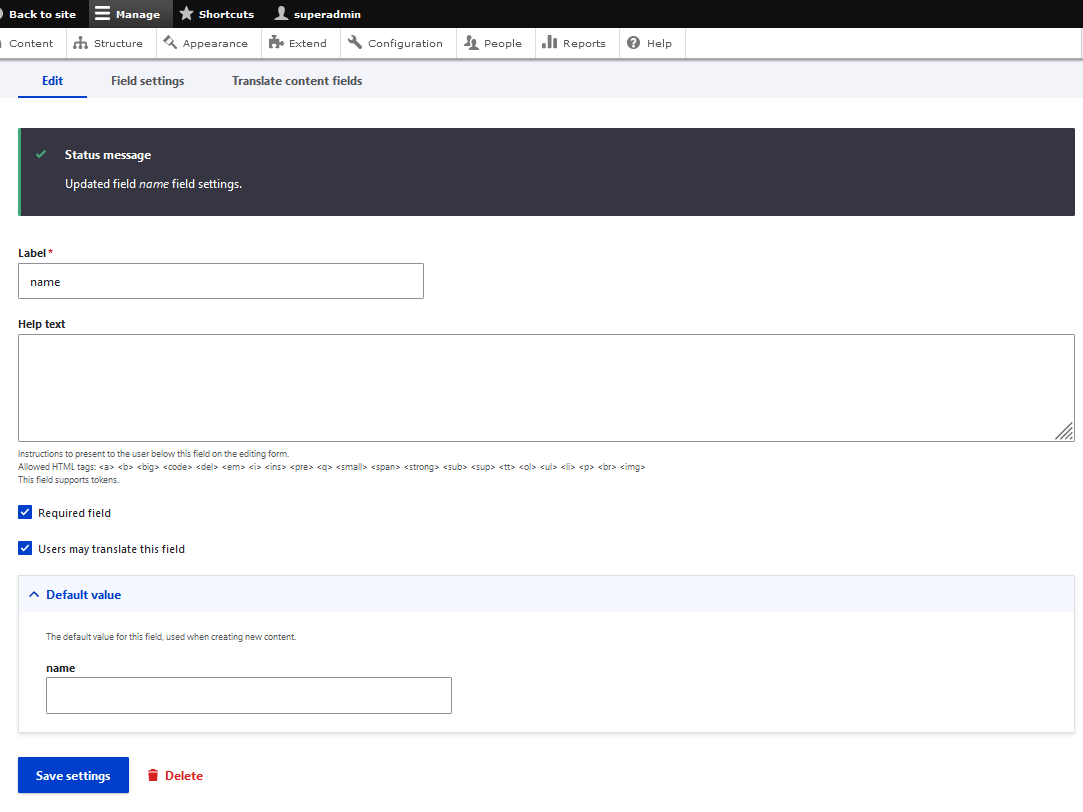


Рис. 23 Додавання поля до типу контенту крок 3

Аналогічним чином додаємо усі інші поля. За виключенням типу фото, яке вже будується на основі існуючого поля рис. 24

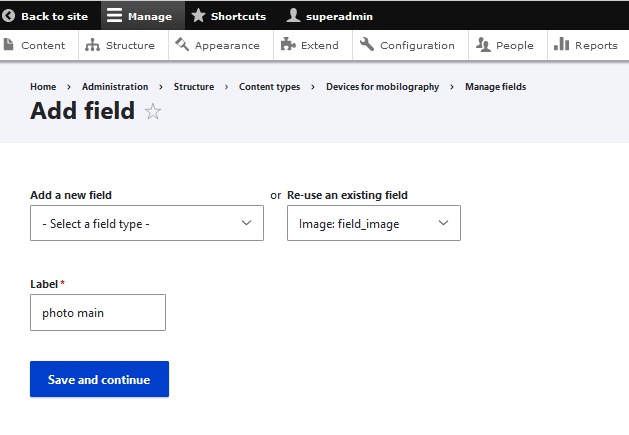


Рис. 24 Додавання існуючого поля.

Додавання поля типу "Media" для вставки зображень відбувається аналогічним чином, з урахуванням переваг, які поля посилань на медіа (Media) мають порівняно з основними посиланнями на файли та зображення. У полях посилання на медіа (Media) можна вказати посилання на кілька типів носіїв в одному полі. Крім того, до самого типу носія можна додавати поля, що дозволяє додавати власні метадані, такі як описи та теги таксономії, для посиланих носіїв. Типи медіафайлів для аудіо- та відеофайлів вже є встановленими за замовчуванням, тому для завантаження цих носіїв не потрібна додаткова конфігурація. Для спільних або спеціальних проєктів можуть бути встановлені додаткові джерела мультимедіа, такі як сторонні веб-сайти або Twitter. Крім того, існуючі медіа-елементи можуть бути використані повторно на будь-яких інших елементах вмісту за допомогою поля посилання на медіа (Media). При виборі цих налаштувань, можна обмежити кількість фотографій до нескінченності та вибрати, що медіа можуть бути лише зображеннями.

Після натискання кнопки "Додати поле" ми продовжуємо з додаванням полів для класифікації (Taxonomy). Вибравши тип поля та його назву, ми переходимо до наступного етапу. Деталі представлені на зображенні 25.

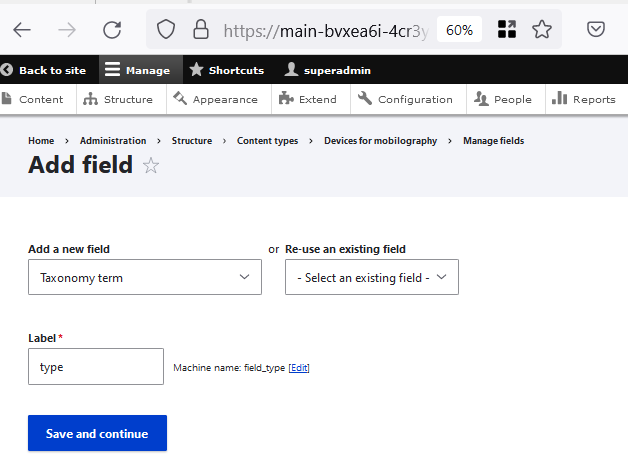


Рис. 25 Додавання посилаючого поля крок 1

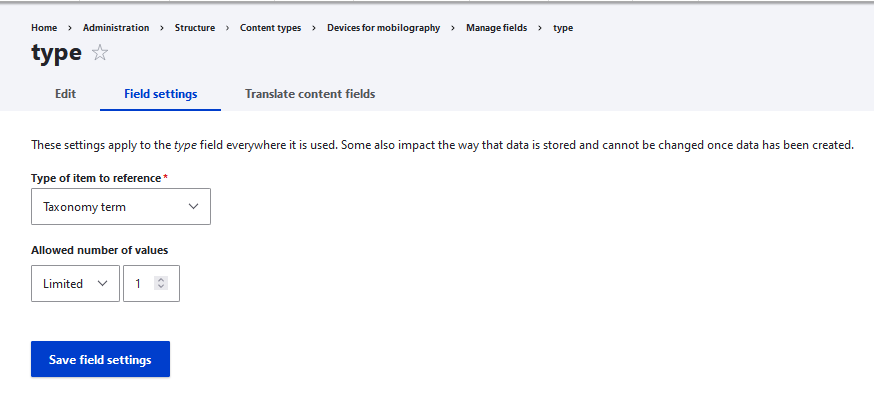


Рис. 26 Додавання посилаючого поля крок 2

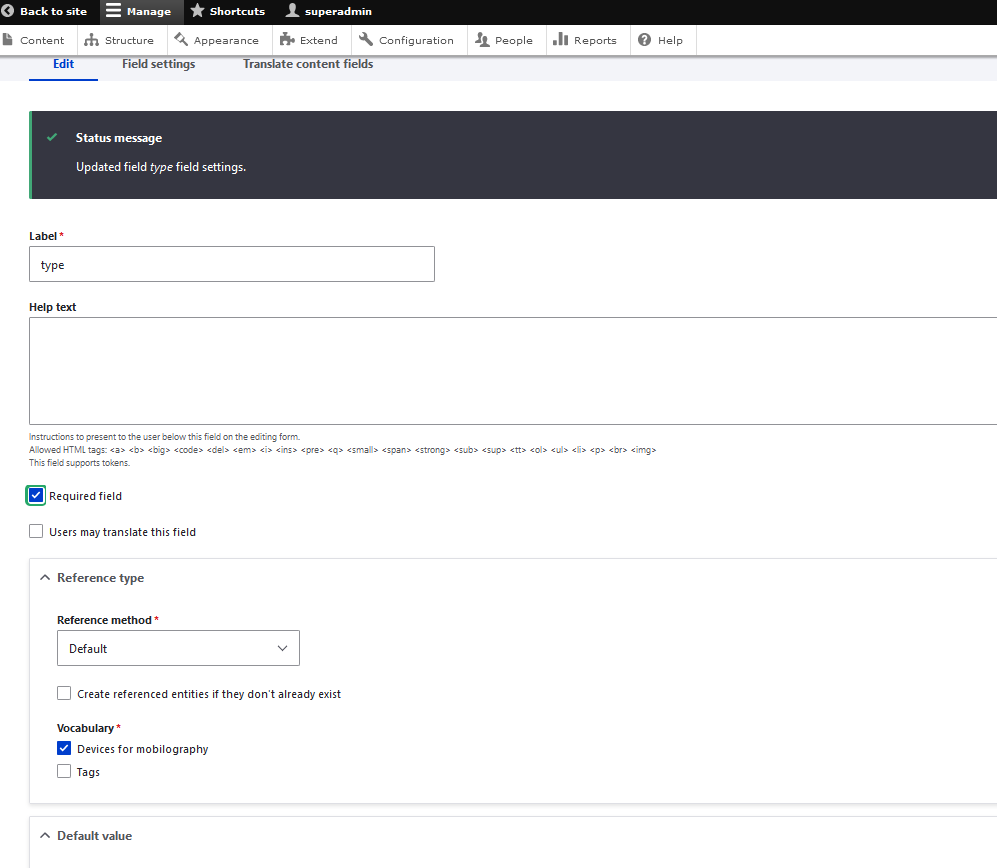


Рис. 27 Додавання посилаючого поля крок 3

Далі Drupal запитує до яких словників відноситься цей тип і яким чином він буде відображатися рис. 26, 27

Після завдання усіх полів типу матеріалів, переходимо для створення відповідних елементів шлях node/add/devices\_for\_mobilography див. рис. 28

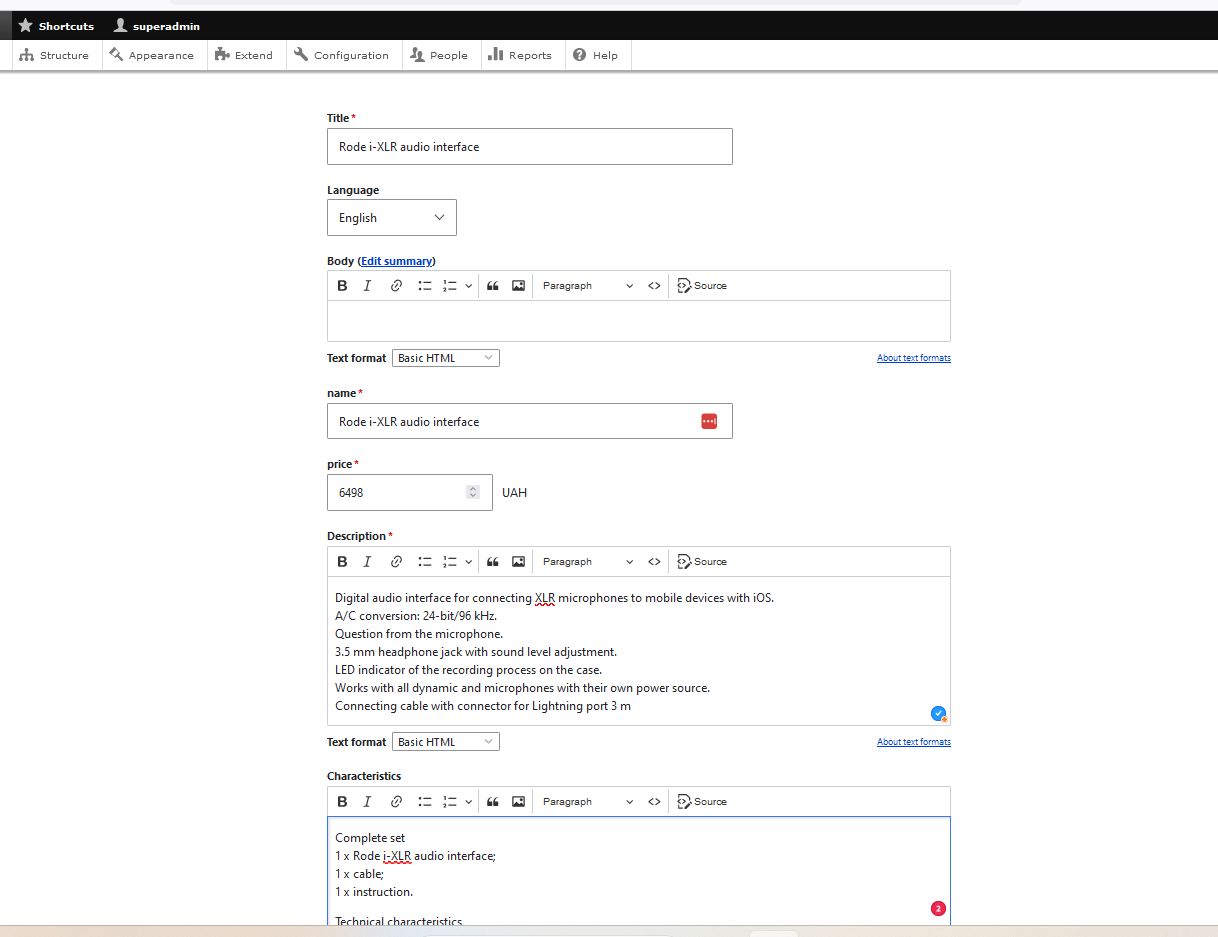


Рис. 28 Приклад додавання нового пристрою крок 1

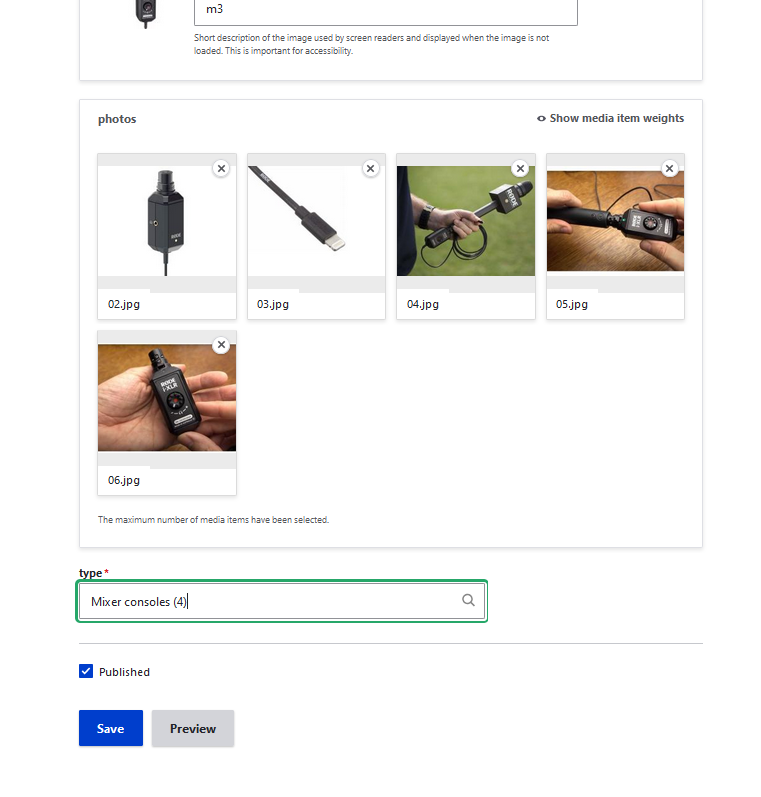


Рис. 29 Приклад додавання нового пристрою крок 2

## 3.3 Темізація сайту

Почнемо із зміни візуалізації певного типу даних. Перевпорядкування полів змінює їх порядок, який відображається, коли користувач створює або редагує вміст за допомогою цих полів. На наступному знімку екрана показано сторінку «Керування полями». admin/structure/types/manage/devices\_for\_mobilography/display/teaser рис. 30.

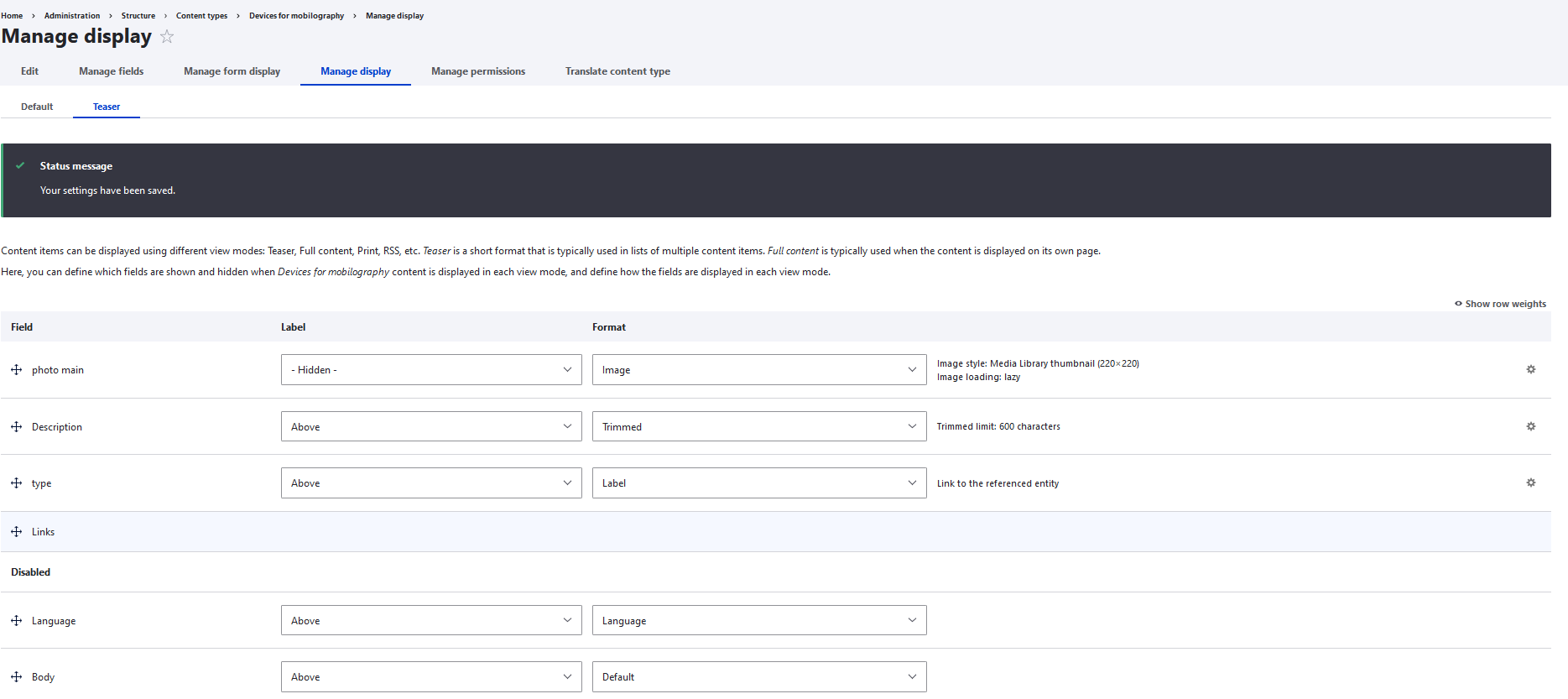


Рис. 30 Керування відображенням полів певного типу.

Елементи змісту можна відображати в різних режимах перегляду: тизер, повний вміст, друк, RSS тощо. Тизер — це короткий формат, який зазвичай використовується у списках із кількома елементами вмісту. Повний вміст зазвичай використовується, коли вміст відображається на окремій сторінці.

Тут можливо визначити, які поля відображатимуться та приховуватимуть, коли вміст Пристроїв для мобільної роботи відображається в кожному режимі перегляду, а також визначити, як поля відображатимуться в кожному режимі перегляду.

Для кожного режиму перегляду Drupal дозволяє: показувати та приховувати поля, змінити порядок відображення полів, приховати або показати мітку поля та, змінити спосіб відображення певних типів полів (наприклад, чи обрізається вміст поля).

Наприклад на сайті наголоджено відображення тільки, назви, опису, типу рис. 31

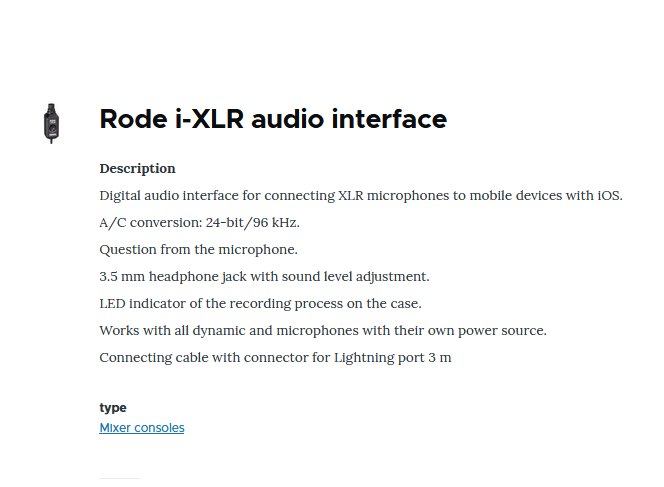


Рис. 31 Відображення пристрою в режимі Teaser

Як вже говорилось вище – це те, що робить сайт Drupal таким, яким він виглядає. Themers, або розробники тем, використовують HTML, CSS, JavaScript та інші зовнішні ресурси, щоб реалізувати дизайн для свого сайту. Кожна окрема тема — це набір файлів, які визначають рівень презентації для вашої програми. Теми, як правило, є одним із перших місць, де код налаштовується для сайту Drupal, і в багатьох випадках вони унікальні для конкретного сайту, для якого вони були створені.

Замість того, щоб починати з нуля, теми Drupal починаються з існуючої структури HTML і вносять зміни за потреби, замінюючи та змінюючи лише необхідні шаблони. Деякі теми потребують зміни лише кількох вибраних бітів, тоді як інші можуть змінити майже все. У будь-якому випадку, якщо це HTML, ви можете змінити його за допомогою теми.

Для того, щоб це працювало, кожен компонент Drupal, який повинен відобразити щось у браузері, надає простий мінімальний шаблон HTML для цього елемента. Будь то вміст вузла, логотип сайту, що відображається в заголовку, або навіть сама область заголовка, необхідний HTML відображається з шаблону. Ці шаблони можуть бути перевизначені темою, щоб змінити розмітку, яку вони створюють.

Теми використовуються для:

Змініть HTML-розмітку будь-чого в Drupal

Додайте стилі CSS, щоб змінити макет, колір або типографіку на одній або кількох сторінках

Використовуйте JavaScript для покращення взаємодії з користувачем

Більшість тем поєднуватимуть зміни в розмітці HTML із новими файлами CSS, які забезпечують макет і загальну графічну обробку сайту, і JavaScript, який змінює способи взаємодії користувачів із вмістом сторінки. Поєднайте все це, і ви зможете зробити так, щоб Drupal виглядав і відчувався як усе, що ви можете собі уявити.

На базі існуючою теми було зроблено нову тему, починається створення теми із створення теці[3], а в ній файлу info.yml де вказуються параметри теми, його зміст представлено у Додатку В.

А, наприклад, файл block.html.twig містить команди, які визначають відображення блоку див. рис. 32



Рис. 32 Основний зміст block.html.twig

Якщо зміна label істина, то тоді відображається div із класом block\_heading в якому міститься мітка блоку, тобто в залежності від конфігурування мітка відображається, чи не відображається на сторінці.

Також всередині блоку content відображається основний контент блоку у тегу div із класом block\_content.

Структура сторінки задається у файлі layout/html.html.twig рис. 33

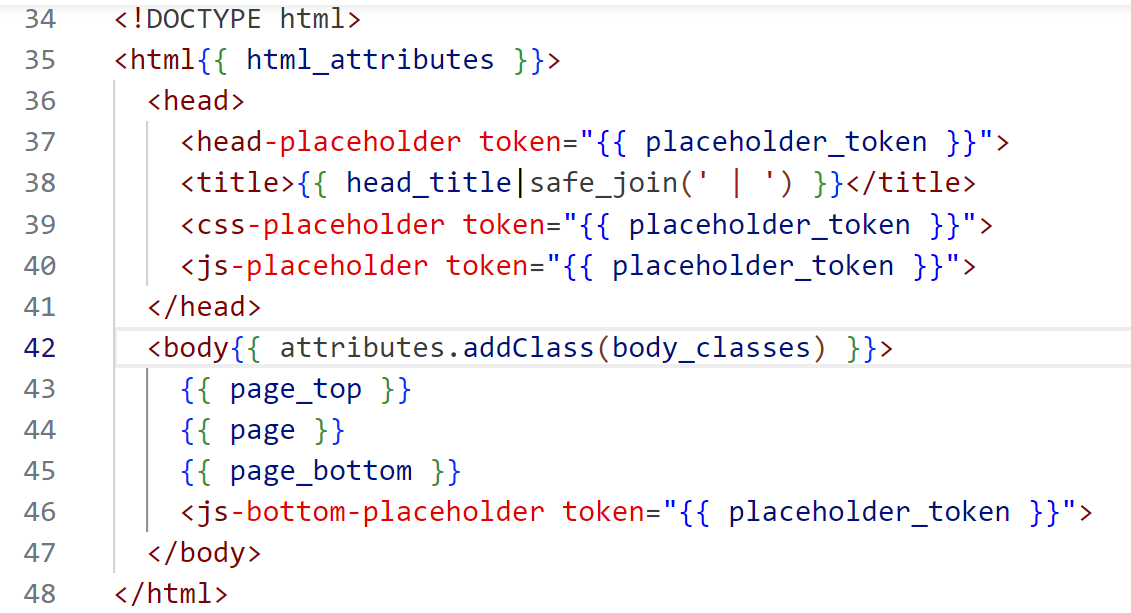


Рис. 33 Основний зміст html.html.twig

Рядки 35-41 відповідають за відображення частини head в якій відображається заголовок сторінки, під’єднуються стилеві файли. А рядки 42-47 основної частини, яка складається із верхньої частин, основного змісту, та нижньої частини.

## Висновки до розділу 3

Описано основні кроки створення сайту з допомогою CMS Drupal.

Для реалізації сайту та тестування сайту було створено сайт у провайдера platform.sh .

Сайт наповнений графічним і текстовим контентом. Створений сайт готовий до запуску в інтернет і відповідає усім вимогам.

# ВИСНОВКИ

Під час дослідження було проаналізовано науково-технічний матеріал, аналіз якого дозволив розробити веб-сайт «Мобілографія для всіх» При цьому було отримано наступні висновки та результати:

Веб-сайт – це електронна сторінка або колекція сторінок, які доступні через Інтернет і містять різноманітну інформацію, таку як текст, зображення, відео, аудіо та інші елементи. Веб-сайти використовуються для різних цілей, таких як надання інформації, продаж товарів і послуг, комунікація зі спільнотою тощо.

Розробка веб-сайтів включає кілька етапів. Перш ніж розпочати створення веб-сайту, необхідно провести аналіз вимог, визначити цілі та цільову аудиторію. Далі, на етапі планування, визначається структура веб-сайту, його дизайн та технології, що будуть використовуватися.

Розглянуто основні способи створення сайтів (ручній на основі Code, ручній на основі WYSIWYG, CMS фреймворк Конструктори), а також проведено порівняльний аналіз основних CMS на мові PHP (Joomla!, WordPress, Drupal).

Проведені дослідження надали можливість визначити, що сайт «Мобілографія для усіх» буде реалізовано з допомогою CMS Drupal.

В другому розділі роботи було проведено порівняльний аналіз останніх трьох версій CMS Drupal і для реалізації обрано 10, бо вона є останньою версією популярної відкритої платформи управління контентом (CMS) станом на час написання цієї роботи. З поліпшеними заходами безпеки, оптимізацією продуктивності та інструментами для зручності використання, Drupal 10 надає бізнесам усі необхідні переваги для створення веб-сайтів високої якості, які залучають своїх користувачів.

Розглянуто основні концепції Drupal: модуль, тема, проєкт, типи даних, сутність, таксономія, поле, перегляд, користувачі, дозволи, ролі.

Розглянуто теоретичні основи темізаціїї Drupal.

У третій частині було розглянуто основні кроки для реалізації сайту та тестування сайту «Мобілографія для всіх», а само налаштування при першому запуску, мультимовності, включення блоків, показано основні етапи створення простого контенту, нового типу контенту (моделі даних).

Для реалізації сайту та тестування сайту було створено сайт у провайдера platform.sh .

Сайт наповнений графічним і текстовим контентом. Створений сайт готовий до запуску в інтернет і відповідає усім вимогам.

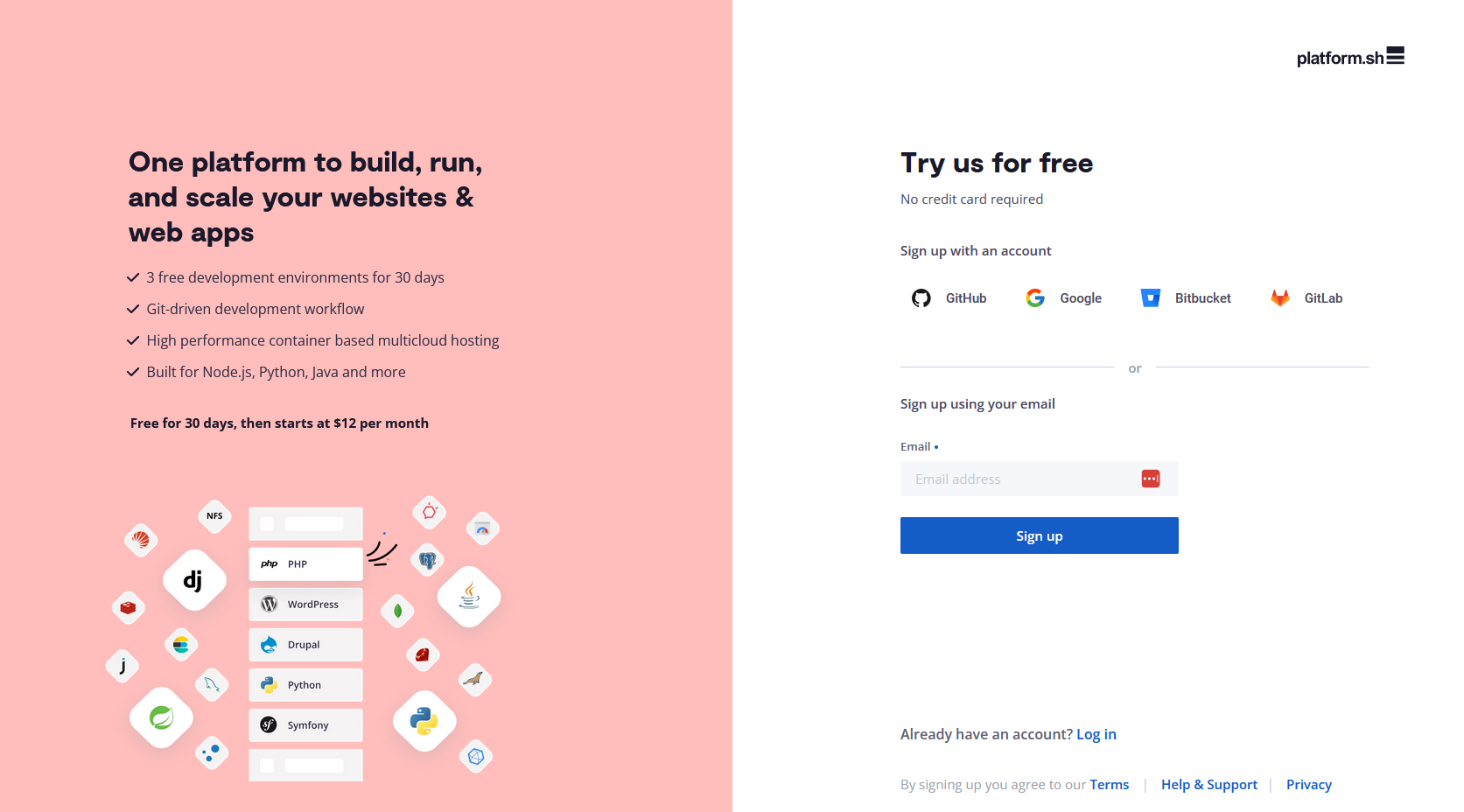
# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

* 1. Ahmed S. Best PHP CMS For Developers in 2023. The Official Cloudways Blog. URL: https://www.cloudways.com/blog/best-php-cms/ (дата звернення: 10.05.2023)
  2. Cascading Style Sheets home page: веб-сайт. URL: https://www.w3.org/Style/CSS/. (дата звернення 13.05.2023)
  3. Chumley, C. Drupal 8 theming with Twig / C. Chumley. – Birmingham, UK: Packt Publishing, 2016.
  4. Drupal 9 VS Drupal 10 – Top difference between Drupal 9 & 10. Web Development Agency Las Vegas - Skynet Technologies. URL: https://www.skynettechnologies.com/blog/drupal-9-vs-drupal-10 (дата звернення: 2.04.2023)
  5. Glaman, M. Drupal 10 development cookbook / M. Glaman, K. Quillen. – Birmingham: Packt Publishing, 2023.
  6. Haverbeke M. Eloquent javascript: a modern introduction to programming. 3rd ed. San Francisco, California, United States of America : No Starch Press, 2018. 472 p.
  7. HTML Living Standard: веб-сайт. URL: https://html.spec.whatwg.org/multipage/. (дата звернення 13.05.2023)
  8. HTML і CSS довідник українською. Html CSS довідник. URL: https://html-css.co.ua/ (дата звернення: 06.05.2023)
  9. Mayekar, D. Decoupling Drupal / D. Mayekar. – Berkeley CA: Apress; Imprint: Apress, 2017.
  10. Navin I. Drupal 9 vs. Drupal 10: Which is best for your website?. drupalpartners. URL: https://www.drupalpartners.com/blog/drupal-9-vs-drupal-10-which-best-your-website (дата звернення: 22.04.2023).
  11. Sipos, D. Drupal 10 Module Development / D. Sipos. – Birmingham: Packt Publishing, 2023.
  12. The birth of the Web: веб-сайт. URL: https://home.cern/science/computing/birth-web. (дата звернення 12.05.2023)
  13. YAML Ain't Markup Language. The Official YAML Web Site. URL: https://yaml.org/ (дата звернення: 15.05.2023)
  14. Аронов А. О. Аналіз підходів до створення сайтів та вимог щодо їх інформаційного наповнення. Зв'язок. 2018. № 2. С. 40–43. URL: http://con.dut.edu.ua/index.php/communication/article/view/1991 (дата звернення: 02.05.2023).
  15. Вебсайт :веб-сайт. URL: https://uk.wikipedia.org/wiki/Вебсайт (дата звернення 12.05.2023)
  16. Гніденко І.А. Дослідження засобів для створення та супроводження WEB-сайтів / Гніденко І.А. Воробйов І.Є. / Проблеми інформатизації та управління. - К.: НАУ, 2019. - 2 (62). - С.31–36.
  17. Іванова І. Засоби створення веб-ресурсів та їх класифікація. Актуальні питання сучасної інформатики: матеріали доп. VІ Всеукр. наук.-практ. конф. з міжнар. участю, 18-19 листоп. 2021 р... 9. 2022

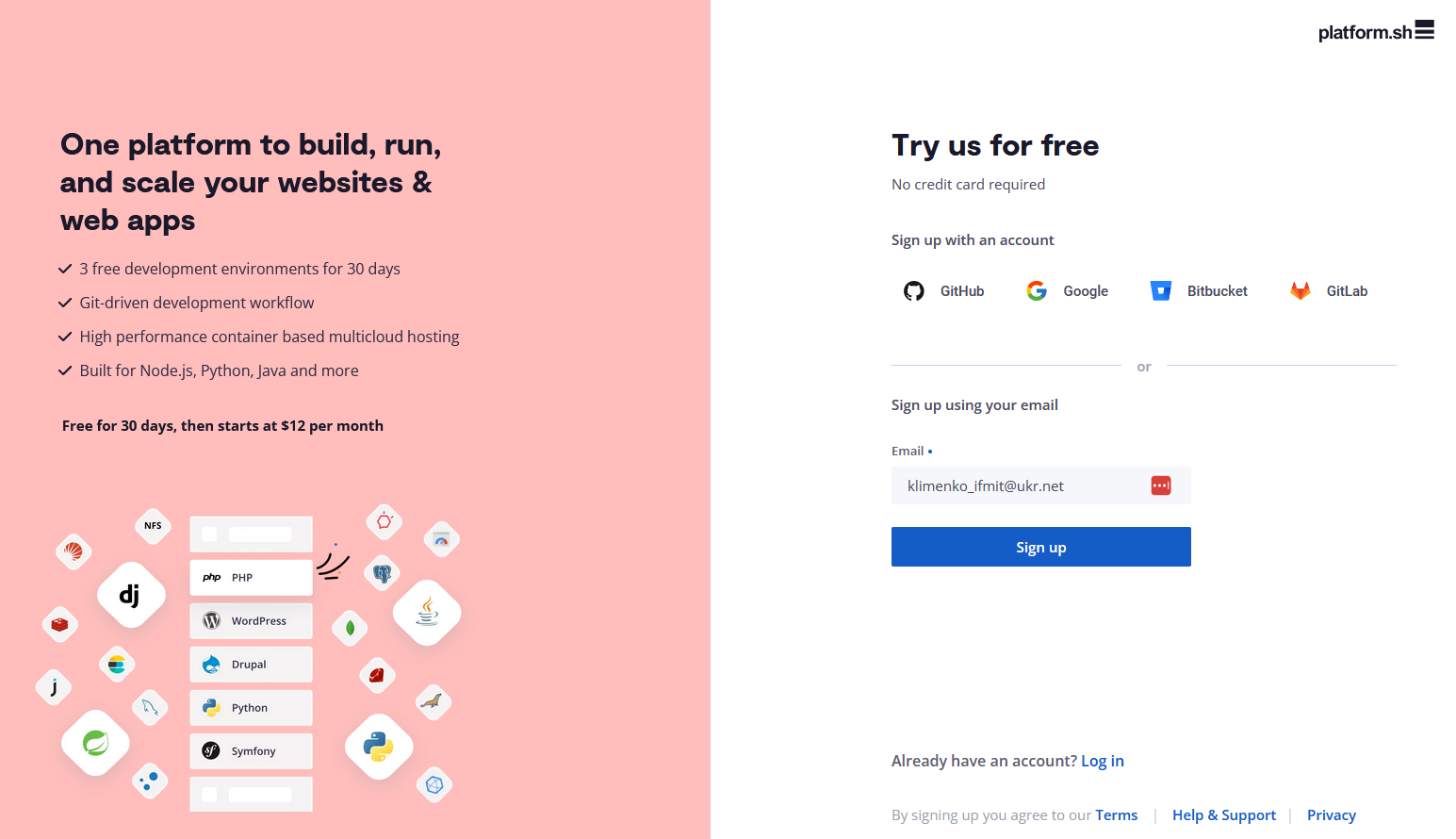
# Додатки

## Додаток А Процес реєстрації на platform.sh

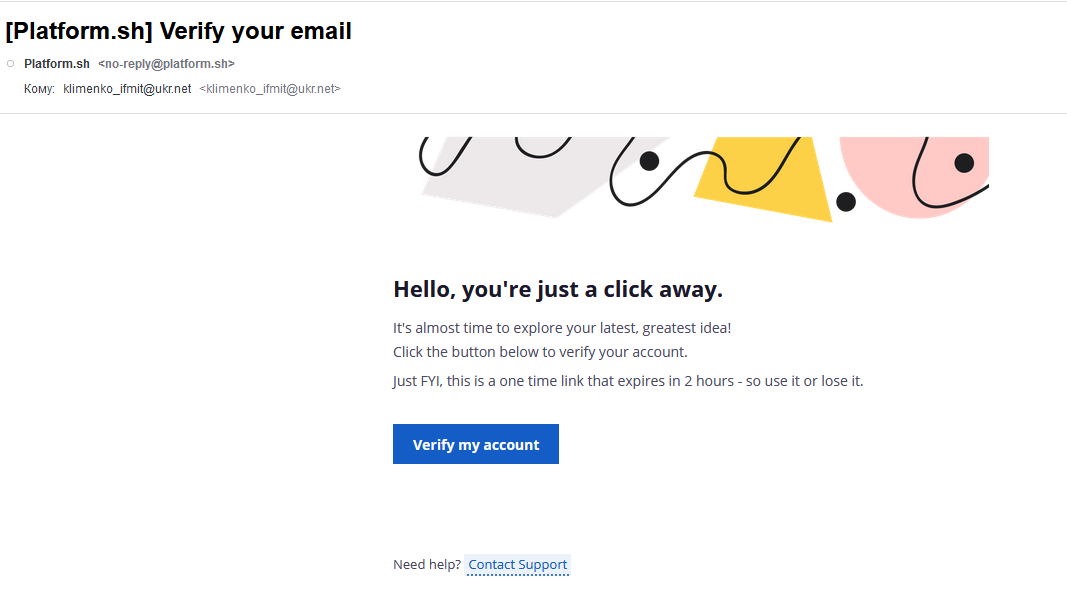
Крок 1 початок реєстрації



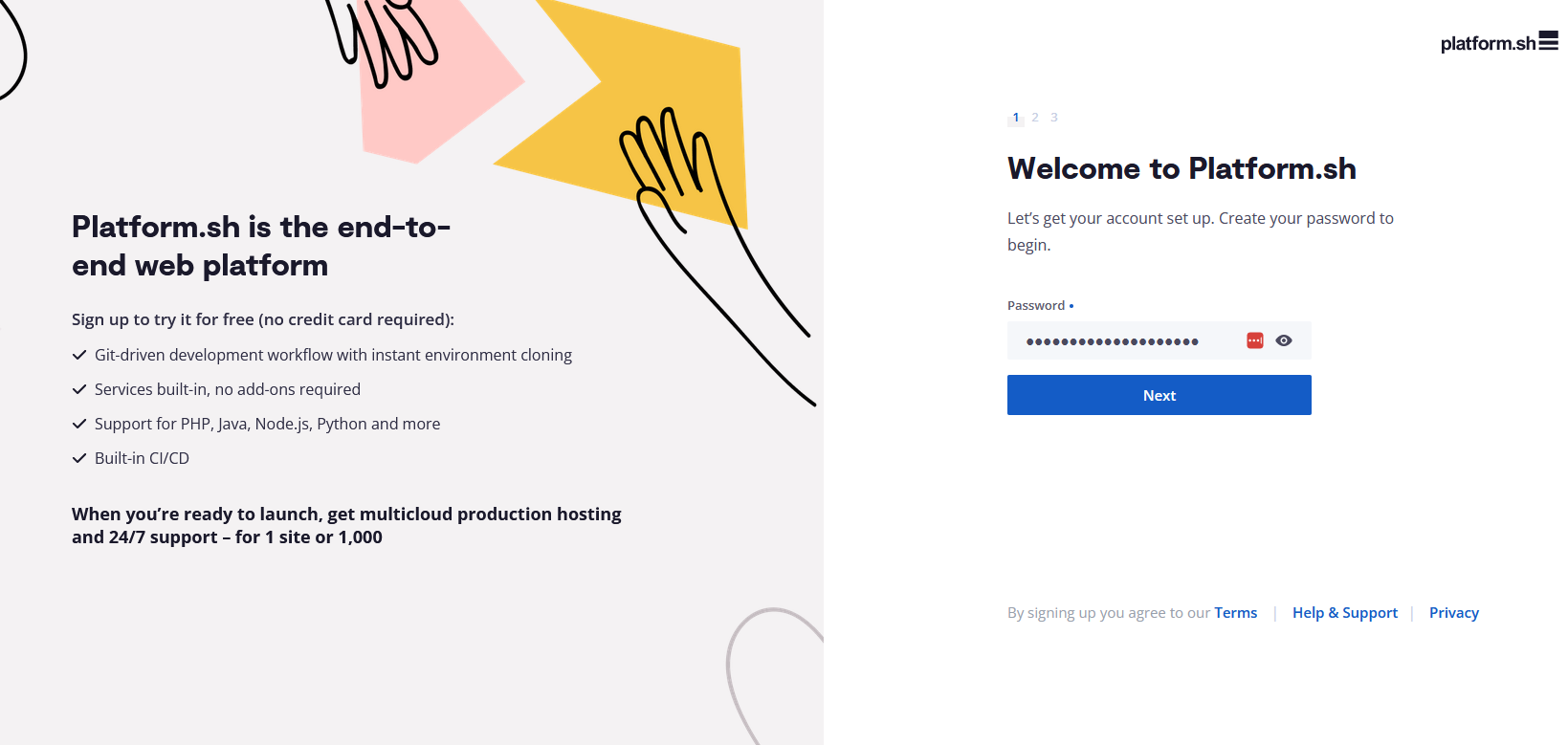
Крок 2 Вказівка електроної пошти



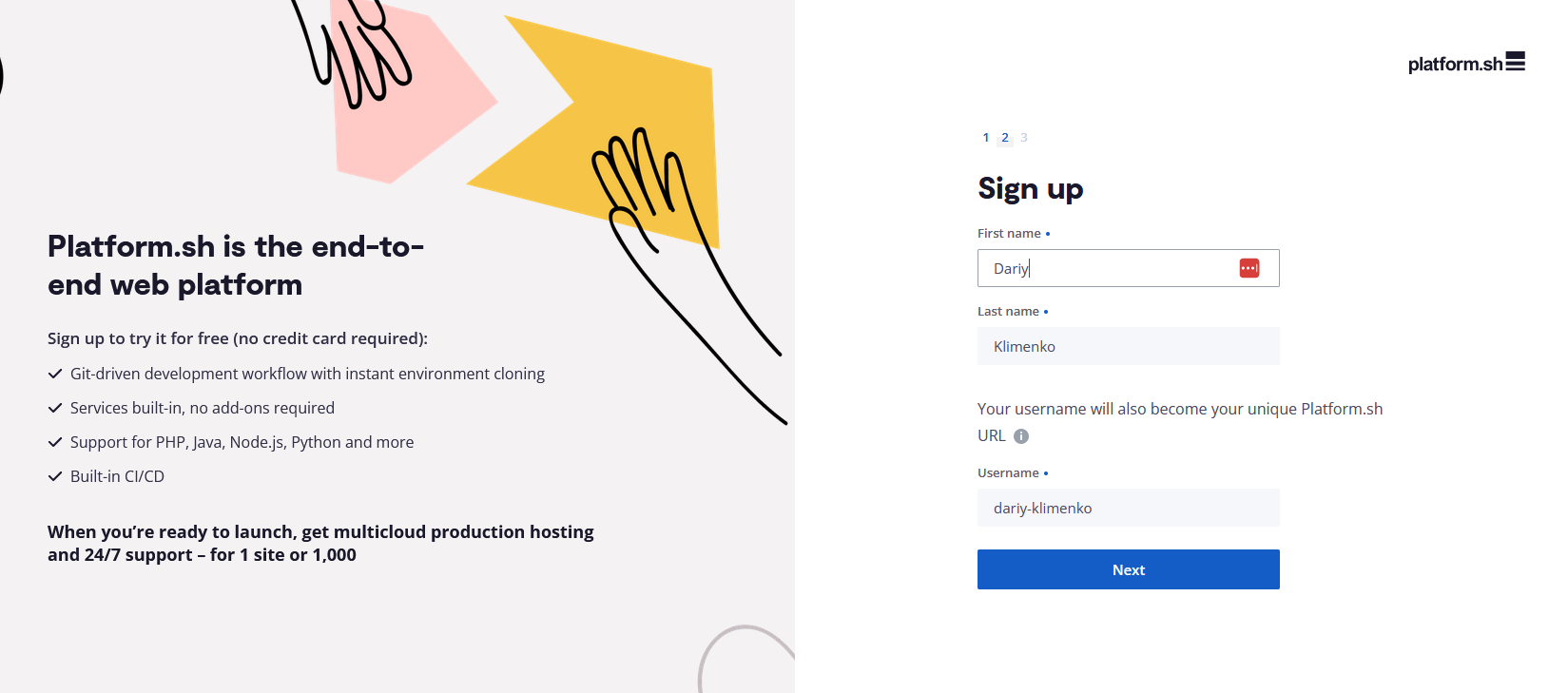
Крок 3 Підтвердження електронної пошти



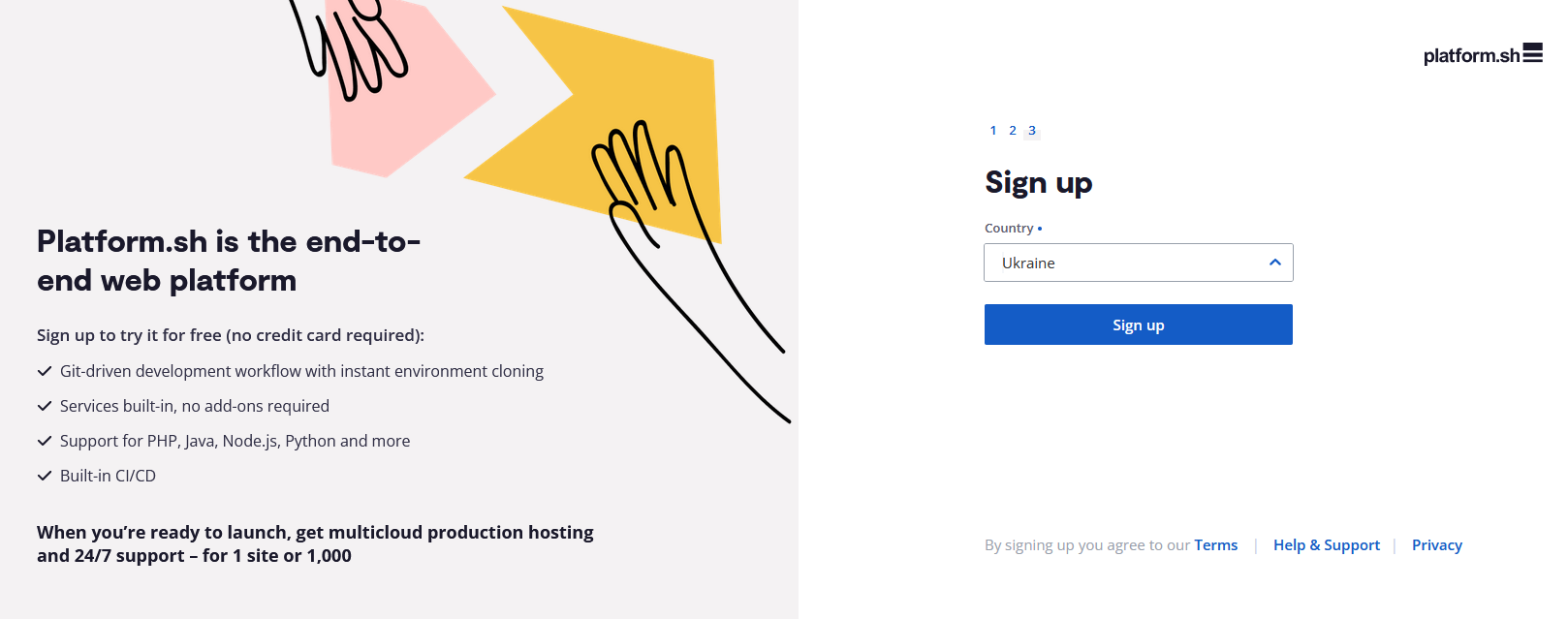
Крок 4 встановлення паролю



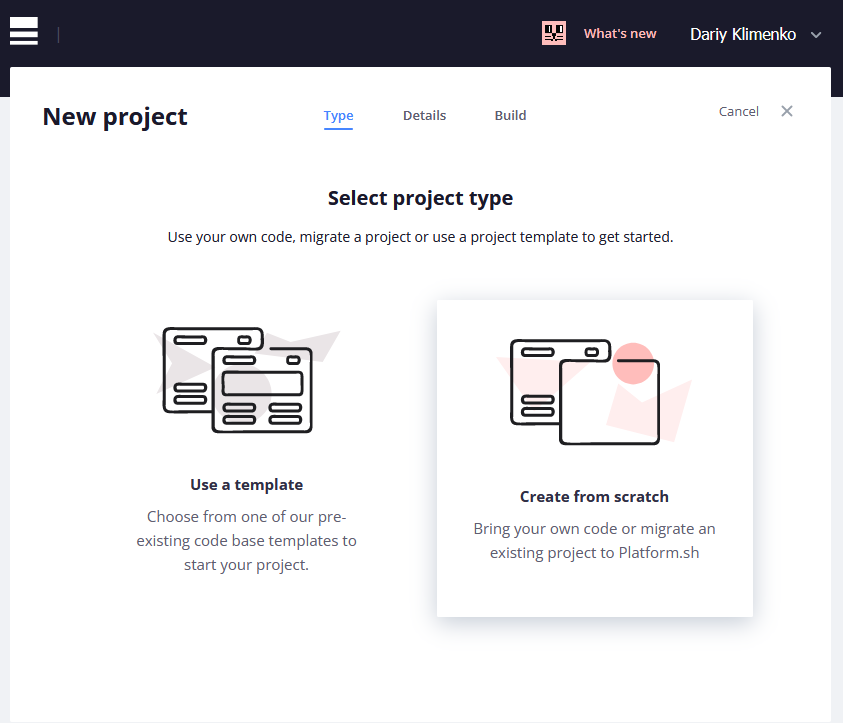
Крок 5 Вказівка особистих даних



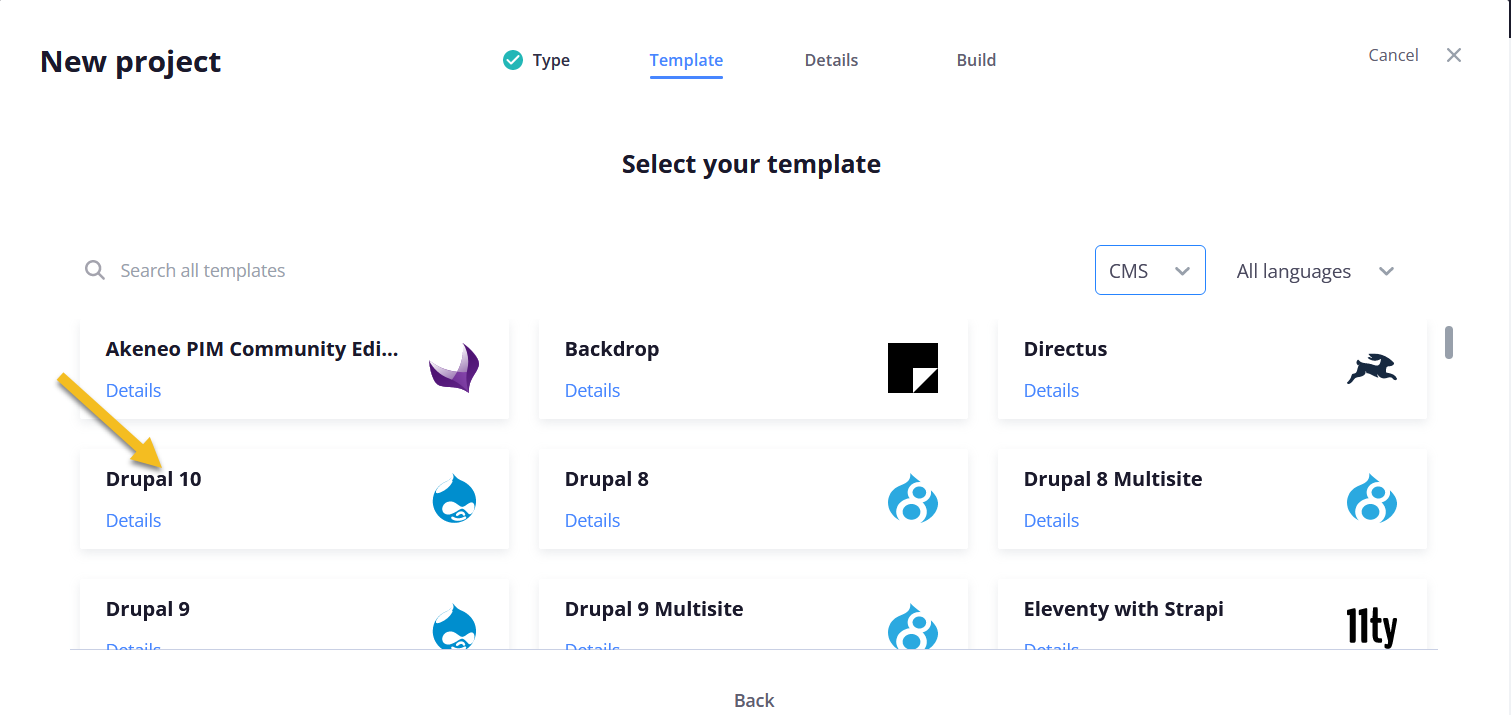
Крок 6 Вказівка країни



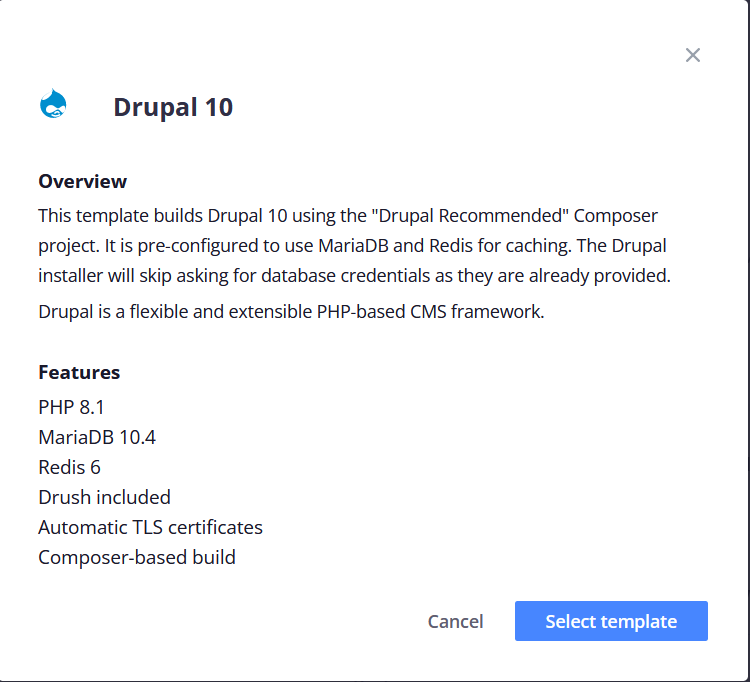
Крок 7 Вибір типу проєкту



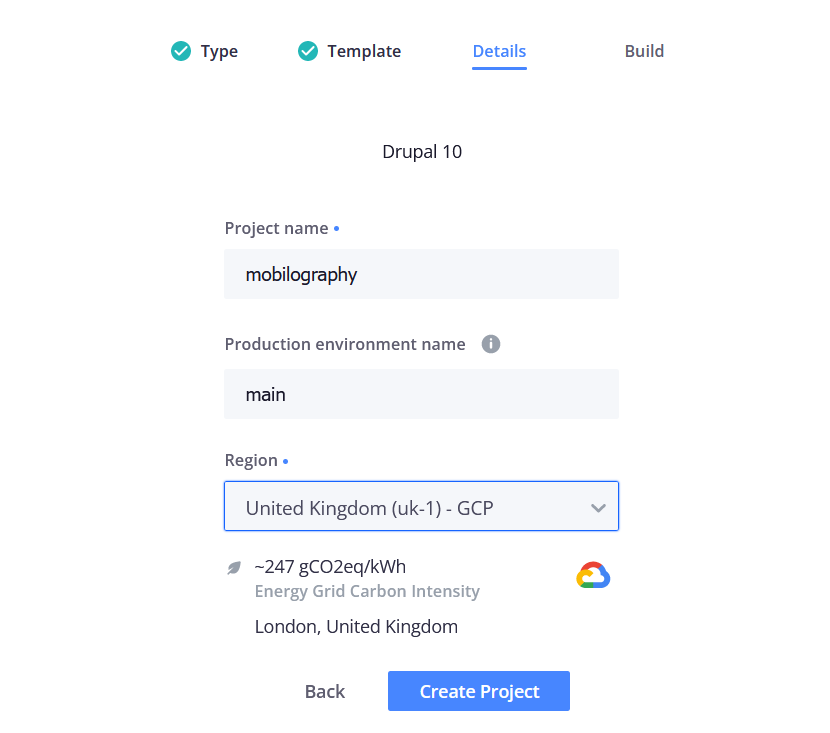
Крок 8 Вибір проєкту Drupal 10



Крок 11 Вибір налаштування



Крок 12 Вибір хмарного провайдера



## Догадок Б Основні команди composer для праці із Drupal

**Створення сайту**

composer create-project drupal/recommended-project my\_site\_name\_dir

Це створить проєкт у 'my\_site\_name\_dir' і автоматично виконає установку композитора для завантаження останньої стабільної версії Drupal і всіх її залежностей.

**Оновлення залежностей.**

composer update

Використовуючи цей рекомендований репозиторій як початкову точку, ми уникаємо залежності проблеми, використовуючи лише залежності, які вже були протестовані з вашим версія Drupal.

Наданий файл composer.json містить найкращий практичний спосіб структурування проєкт drupal. Він автоматично забезпечить отримання модулів, тем і бібліотек завантажено в потрібну папку. Створений файл composer.lock відстежує кожну точну версію, яку встановили.

web — це веб-корінь проєкту це папка, на яку буде вказано ваше доменне ім’я;

vendor — папка з усіма залежностями.

**Встановлення залежностей (модулів, тем)**

composer require drupal/redirect

ця команда завантажить модуль перенаправлення до modules/contrib, а посилання на нього буде додано до нашого файлу composer.json і composer.lock. Якщо модуль має визначені залежності, їх буде завантажено та додано до карти постачальника.

**Видалення залежностей.**

composer remove drupal/redirect

Це видаляє модуль. Завжди переконайтеся, що ви спочатку вимкнули модуль у drupal.

## Додаток В Зміст info.yml

name: MobileGraphy

type: theme

description: Base theme inspirited by Material Design concept.

core\_version\_requirement: ^9 || ^10

base theme: stable9

screenshot: screenshot.svg

libraries:

  - core/normalize

  - material\_base/base

  - material\_base/grid

  - material\_base/fonts

  - material\_base/icons-font

regions:

  navbar: 'Navbar'

  header: 'Header'

  actions: 'Actions'

  main\_top: 'Main top'

  content\_above: 'Content above'

  breadcrumb: 'Breadcrumb'

  messages: 'Messages'

  highlighted: Highlighted

  help: 'Help'

  content: 'Content'

  content\_secondary: 'Content secondary'

  sidebar\_first: 'Sidebar first'

  sidebar\_second: 'Sidebar second'

  content\_below: 'Content below'

  main\_bottom: 'Main bottom'

  footer: 'Footer'

  drawer: 'Drawer'

  overlay: 'Overlay'

regions\_hidden:

  page\_bottom: 'Page bottom'

  page\_top: 'Page top'