

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «РОЗРОБКА ТЕМИ ДЛЯ СИСТЕМИ
КЕРУВАННЯ САЙТОМ DRUPAL 3
ВИКОРИСТАННЯМ TAILWIND CSS»

Виконала: студентка 4 курсу, групи 6.1220-з
спеціальності 122 комп'ютерні науки
(шифр і назва спеціальності)

освітньої програми комп'ютерні науки
(назва освітньої програми)

Дребезова Ангеліна Юріївна
(ініціали та прізвище)

Керівник доцент кафедри комп'ютерних наук, к.т.н.,
Добровольський Г.А.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент професор кафедри програмної інженерії,
доцент, к.ф.-м.н. Кудін О.В.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти бакалавр

Спеціальність 122 комп'ютерні науки

(шифр і назва)

Освітня програма комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук, к.ф.-м.н., доцент

_____ Шило Г.М.

(підпис)

“ 22 ” грудня 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ

Дребезовій Ангеліні Юрїївні

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка теми для системи керування сайтом Drupal з використанням Tailwind CSS

керівник роботи Добровольський Г.А., доцент кафедри комп'ютерних наук, к.т.н.
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 21 » грудня 2023 року № 2181-с

2. Строк подання студентом роботи 15.05.2024 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. Основні теоретичні відомості.

3. Розробка та реалізація теми.

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

Презентація за темою роботи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 25.12.2023 р.**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	09.01.2024	
2.	Збір вихідних даних.	31.01.2024	
3.	Обробка методичних та теоретичних джерел.	26.02.2024	
4.	Розробка першого та другого розділу.	01.01.2024	
5.	Розробка третього розділу.	01.05.2024	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	15.05.2024	
7.	Захист кваліфікаційної роботи.	30.05.2024	

Студент _____
(підпис)А. Ю. Дребезова
(ініціали та прізвище)Керівник роботи _____
(підпис)Г.А. Добровольський
(ініціали та прізвище)**Нормоконтроль пройдено**Нормоконтролер _____
(підпис)О. Г. Спиця
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка теми для системи керування сайтом Drupal з використанням Tailwind CSS»: 38 с., 19 рис., 4 табл., 6 джерел, 2 додатки.

CSS, DRUPAL, TAILWIND, VITE, КОЛІРНА СХЕМА, ТЕМА, ФРЕЙМВОРК.

Об'єкт дослідження – процес створення та впровадження теми для CMS Drupal з використанням CSS-фреймворку Tailwind.

Мета роботи: оволодіння сучасними методами дослідження та розв'язання комплексної задачі розробки теми для системи керування контентом з застосуванням сучасної технології.

В результаті роботи розроблено тему, яка адаптується під час роботи з різними пристроями, що спрощує процес створення сайтів; показано легкість кастомізації: Tailwind CSS дозволяє швидко та легко змінювати стилі, що допомагає користувачам Drupal налаштувати дизайн відповідно до своїх потреб.

SUMMARY

Bachelor's qualifying paper «Creation of Drupal CMS Theme Using Tailwind CSS»: 38 pages, 19 figures, 4 tables, 6 references, 2 supplements.

CSS, DRUPAL, TAILWIND, VITE, COLOR SCHEME, THEME, FRAMEWORK.

The object of the study is : the process of creating and implementing a theme for Drupal CMS using the Tailwind CSS framework.

The aim of the study is to master modern research methods and solve the complex task of developing a theme for a content management system using modern technology.

As a result of the work, a theme has been developed that adapts to different devices, which simplifies the process of creating websites; it shows the ease of customization: Tailwind CSS allows you to change styles quickly and easily, which helps Drupal users customize the design to suit their needs.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат	4
Summary	5
Вступ.....	7
1 Дослідження предметної галузі	8
1.1 Система керування контентом drupal	8
1.2 Важливість дизайну веб-сайтів.....	10
1.3 Огляд tailwind css. Основні концепції та переваги	11
1.4 Висновки першого розділу.....	14
2 Проєктування дизайн системи	15
2.1 Використання tailwind css для створення дизайну	15
2.1.1 Палітра кольорів	15
2.1.2 Конфігурація адаптивності	18
2.1.3 Проєктування компонентів	21
3 Розробка теми	24
3.1 Необхідні інструменти	24
3.1.1 Tailwind css інструменти	24
3.1.2 Drupal середовище	25
3.1.3 Редактор коду	26
3.1.4 Система контролю версій	26
3.1.5 Засоби контейнеризації	28
3.2 Налаштування теми	29
3.3 Реалізація компонентів	31

Висновки	33
Перелік посилань.....	34
Додаток А.....	35
Додаток Б	37

ВСТУП

В сучасному світі враховуючи всі тренди на поширення цифрових технологій кількість роботи яка виконується з дому, використання зручних та сучасних інструментів стає надмірно важливим. І не секрет, що за останній час кількість нових технологій значно зросла.

Однією з ніш в якій з'явилися нові підходи до використання технологій стала веб-розробка, а саме браузерна частина (клієнтська частина), і як приклад того, що було створено це Tailwind CSS – сучасний фреймворк з відкритим кодом, який був створений на фоні широкого розвитку інтернет технологій та боротьби за швидкість завантаження сторінок (зменшення розміру коду який завантажується на сторону клієнта, тим самим зменшуючи використання інтернет трафіку) [1].

Але в той же час набули другого дихання проекти створені роки тому, і велика кількість проектів були оновлені за минулу пару років. Одним з таких проектів є Drupal – популярна система курування контентом веб-сайтів та застосунків з відкритим кодом та модульною архітектурою. Широкого застосування набув в галузях освіти та корпоративному середовищі [2].

Кількість програмного забезпечення, яке оновлюється та те, яке треба підтримувати, наштовхнула нас на мета та предметну галузь дослідження.

Мета дослідження – поєднати стару, але популярну технологію корпоративного рівня, з новою але не менш популярною технологією останніх років. В якості поєднання було обрано розробку теми засобами Tailwind CSS для системи керування вмістом Drupal.

В результаті роботи хотілось би отримати готову до використання тему для Drupal, що дозволить користувачам створювати власні веб-сайти та застосунки з її використанням. А також супутню документацію, яка буде містити схему компонентів та інструкцію з застосування.

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Система керування контентом Drupal

Drupal – некомерційна система, яка може слугувати як окремою системою керування контентом (content management system) (далі CMS), так і фреймворком (каркасом) для створення набору представлень інформації в мережі інтернет. З восьмої версії системи може використовуватись для управління знаннями завдяки новим модулям організації меню та класифікацій [2].

Зазвичай система керування контентом написана на скриптовій мові, і ці скрипти запускаються на машині де є база даних та веб-сервер. Контент та налаштування для системи зберігаються в базі даних, і для кожного запиту який йде на сервер, скрипти збирають інформацію з бази даних і додаткові файли, такі як Javascript або Typescript скрипти, CSS файли стилів, файли зображень, інших типів медіа та документів, і так далі, які є складовими частинами сторінки, або ресурсу який запитує користувач.

У випадку Drupal – написаний на скриптовій мові PHP з частинами Javascript та CSS. Саме ядро Drupal, основна частина яка надає скрипти для запуску основного функціоналу системи керування інформацією, повністю написана на мові PHP. А ось інші складові системи (модулі) можуть бути написані з використанням іншої мови зі світу веб-розробки.

Модуль це набір файлів, які розширяють можливості сайту та додають функціональність. Кожен модуль призначено для економії часу на розробку та тестування працездатності системи, а також надає користувачам можливість створювати більш унікальний контент.

Ядро Drupal надає такі модулі за замовчуванням:

- керування обліковими записами користувачів;

- керування основним змістом (модуль ядра Node) та полями (модуль Field);
- керування меню;
- керування списками та таблицями.

Завдяки своїй модульній архітектурі система Drupal дозволяє додавати, публікувати, редагувати, та видаляти контент за сайтах за допомогою браузера на телефоні, планшеті, ноутбучі чи комп'ютері через простий інтерфейс користувача без необхідності роботи з кодом у більшій частині сценаріїв використання [3].

Інтерфейс сторінок в розроблених сайтах складається з елементів контенту, які входять до складу модулю Node. Такими блоками є:

- головне меню;
- другорядне меню;
- заголовок сторінки;
- основний текст сторінки;
- підвал сайту;
- блок зв'язаного контенту;
- і т.д.

Кожен елемент має власні налаштування, що дозволяють обрати на якому місці він буде відображатись, а також зробити декілька копій одного блоку з різними конфігураціями[4].

Ще одною частиною системи Drupal є тема оформлення.

Тема оформлення – це набір файлів, які визначають зовнішній вигляд сайту, або застосунку. Ядро Drupal і модулі визначають контент вашого сайту, а тема оформлення визначає як HTML та CSS прикрасять контент.

Ядро Drupal надає декілька базових тем разом зі встановлювальними файлами системи. Теми були розроблені спільнотою протягом багатьох років і будуть гарним вибором для створення першого сайту. Але через підтримку спільнотою та розповсюдженість системи Drupal ввести постійні зміни в стандартні теми, а особливо з використанням сучасних технологій, є складним

процесом з тривалим періодом застосування змін. Тому багато розробників розроблюють тему для системи під власні потреби та необхідний дизайн сторінок клієнтської частини. Проста користувачька тема може бути простим CSS файлом створеним та підключеним згідно до документації, а інша тема може використовувати JS фреймворки та збиральники фронтенд частини, все залежить від потреб замовника та розробника.

1.2 Важливість дизайну вебсайтів

Дизайн вебсайту – це процес створення вигляду та функціоналу вебсторінок з метою забезпечення ефективної взаємодії між користувачем та вебсайтом. Він охоплює такі аспекти, як вигляд, організація контенту, взаємодія з користувачем та адаптивність для різних пристроїв

Роль дизайну вебсайту у онлайн-середовищі надзвичайно важлива і багатогранна та складається з наступних факторів.

Перше враження користувача. Дизайн вебсайту – є першим контактом користувача з сайтом чи застосунком і від нього залежить враження та бажання користувача продовжувати роботу з інформаційному просторі, що надає інтернет ресурс.

Психологічний вплив. Кольори, шрифти, композиція та інші дизайнерські елементи можуть впливати на емоції, настрої та поведінку користувачів. Вірно підібраний дизайн може збільшити впізнаваність сайту чи застосунку.

Навігація та відповідність контенту. Чітка та інтуїтивно зрозуміла навігація дозволяє користувачам швидко знаходити потрібну інформацію. Дизайн також допомагає візуально відокремлювати різні типи контенту, безпосередньо впливають на зручність користування та здатність відвідувачів знаходити потрібну інформацію. Ось деякі ключові аспекти:

- чітка структура меню: меню вебсайту повинне бути чіткою та логічною. Кожен пункт меню повинен відповідати певній категорії або розділу вебсайту, щоб користувачі могли легко зорієнтуватися та знайти потрібну інформацію;
- використання ієрархії: ієрархічна організація сторінок дозволяє групувати схожий контент разом, що полегшує навігацію. Важливі розділи або сторінки повинні бути доступні за декілька кліків від домашньої сторінки;
- пошукова функція: важливою частиною навігації є можливість швидко знаходити інформацію за допомогою пошукової функції. Пошукове поле повинно бути легко знайдене та доступне на всіх сторінках вебсайту.

Адаптивність до різних пристроїв. У сучасному світі більшість веб-трафіку генерується з мобільних пристроїв. Тому важливо, щоб дизайн вебсайту був адаптивним та оптимізованим для різних типів пристроїв, забезпечуючи приємний досвід користувачам навіть на невеликих екранах.

Усі ці аспекти демонструють, що дизайн вебсайту відіграє ключову роль у взаємодії з користувачами та досягненні поставлених цілей, таких як залучення відвідувачів, збільшення конверсії та підвищення задоволеності від користування.

1.3 Огляд Tailwind CSS. Основні концепції та переваги

Tailwind CSS – CSS фреймворк з відкритим кодом, був створений Адамом Уетеном і наразі підтримується Tailwind Labs.

Головна особливість tailwind – концепція utility-first. В Tailwind CSS не існує класів які визначають стилі для окремих елементів (кнопка, панель, меню, текстове поле і т.д.), а навпаки кожен клас відповідає одній конкретній CSS властивості(колір, шрифт, розмір, положення, ефекти і т.д.). Замість написання власного CSS розробники можуть використовувати класи для застосування стилів безпосередньо в HTML коді (див. табл. 1.1.), таким чином збільшуючи швидкість розробки та покращуючи продуктивність [3].

Таблиця 1.1 – Інтерпретація HTML коду з класами Tailwind в CSS

Код	<code><div class="m-4 p-4 bg-yellow-200 font-bold rounded-lg"> <p>Please be careful when feeding the birds.</p> </div></code>	
Класи	Tailwind	CSS інтерпретація
	m-4	margin: 1rem;
	p-4	padding: 1rem;
	bg-yellow-200	background-color: rgba(254, 240, 138, 1);
	font-bold	font-weight: 700;
	rounded-lg	border-radius: 0.5rem;

Як можна побачити Tailwind CSS пропонує атомарний підхід до стилів, де кожен клас представляє собою окремий атомарний стиль (наприклад, "bg-blue-500" для задання колірнього фону). Це дозволяє створювати швидкі та гнучкі стилі, комбінуючи класи за потребою.

Також Tailwind CSS легко налаштовується за допомогою файлу конфігурації, що дозволяє налаштовувати колірну палітру, розміри, відступи та інші значення стилів. Всі можливості конфігурації детально описані в документації фреймворку (див. рис 1.1 - 1.2).

```
module.exports = {
  content: ["**/*.twig"],
  theme: {
    colors: {
      primary: "#3b82f6",
      secondary: "#64748b",
      success: "#22c55e",
      info: "#3b82f6",
      warning: "#f59e0b",
      danger: "#ef4444",
      light: "#f1f3f5",
    },
  },
  extend: {},
},
variants: {
  extend: {},
},
plugins: [],
};
```

Рисунок 1.1 – Приклад файлу конфігурації з реального проекту

```

/** @type {import('tailwindcss').Config} */
module.exports = {
  content: ['./src/**/*.{html,js}'],
  theme: {
    colors: {
      'blue': '#1fb6ff',
      'purple': '#7e5bef',
      'pink': '#ff49db',
      'orange': '#ff7849',
      'green': '#13ce66',
      'yellow': '#ffc82c',
      'gray-dark': '#273444',
      'gray': '#8492a6',
      'gray-light': '#d3dce6',
    },
  },
  fontFamily: {
    sans: ['Graphik', 'sans-serif'],
    serif: ['Merriweather', 'serif'],
  },
  extend: {
    spacing: {
      '8xl': '96rem',
      '9xl': '128rem',
    },
    borderRadius: {
      '4xl': '2rem',
    }
  }
}

```

Рисунок 1.2 --- Приклад файлу конфігурації з реального проєкту

Ще цікавою особливістю Tailwind є – JIT (Just in time) компіляція. Це миттєва компіляція в процесі розробки. Так як стилі генеруються по мірі їх запити в проєкті, то немає потреби згодом позбавлятися стилів, що не використовуються на етапі продакшену, а це означає, що розробник отримує абсолютно однаковий CSS і при розробці, так і після «збирання» застосунку чи сайту.

Як можна було зрозуміти з тексту вище Tailwind CSS має наступні переваги:

- Швидкість розробки. Завдяки використанню utility-first підходу, розробники можуть швидко створювати та модифікувати стилі без необхідності власного CSS;

- Гнучкість. Tailwind CSS дозволяє створювати будь-який дизайн, навіть найскладніший, за допомогою комбінації класів, що робить його дуже гнучким та універсальним;
- Оптимізація розміру файлів. Використання лише необхідних класів у HTML-кодi дозволяє уникнути зайвих стилів та оптимізує розмір CSS-файлів;
- Спільнота та документація. Tailwind CSS має активну спільноту розробників, яка надає допомогу та підтримку. Крім того, велика та зрозуміла документація дозволяє швидко ознайомитися з фреймворком та його можливостями.

1.4 Висновки першого розділу

Проаналізувавши технології обрані для дослідження можна виділити наступні твердження:

- Drupal – є потужним інструментом для створення різноманітних веб-проектів, який надає гнучкість, масштабованість та безпеку, сприяючи ефективному управлінню контентом та розвитку веб-присутності;[5]
- Tailwind CSS – це потужний та ефективний інструмент для швидкого та гнучкого створення стилів у веб-розробці. Його основні концепції та переваги роблять його відмінним вибором для будь-якого проекту, який вимагає швидкого розвитку та відмінного дизайну.

2 ПРОЄКТУВАННЯ ДИЗАЙН СИСТЕМИ

Дизайн система – це набір затверджених стандартів, компонентів, правил та принципів, які використовуються для створення та підтримки дизайну. Вона дозволяє забезпечити однорідність та консистентність візуального та функціонального досвіду для користувачів.

2.1 Використання Tailwind CSS для створення дизайну

Як було досліджено раніше Tailwind CSS має гнучкі особливості для кастомізації та конфігурації. Саме цей механізм будемо використовувати для налаштування теми.

Для початку визначимо властивості які ми можемо задати для теми за замовчуванням:

- шрифт;
- стиль шрифту;
- розміри тексту;
- колірна схема;
- розміри екранів для адаптивності.

Кожен з розділів конфігурації є необов'язковим, але для створення гарного кастомізованого зовнішнього вигляду ми розберемо кожен з пунктів переліку [6].

2.1.1 Палітра кольорів

Для налаштування кольорів в темі Tailwind CSS ми можемо обрати одну з включених у фреймворк палітр (див. рис. 2.1). Або ми можемо оголосити власні кольори в файлі конфігурації `tailwind.config.js` розширивши розділ `theme.colors` або `theme.extend.colors` (див.рис. 2.2).



Рисунок 2.1 – Частина стандартних колірних палітр

```
tailwind.config.js

module.exports = {
  theme: {
    extend: {
      + colors: {
      +   'regal-blue': '#243c5a',
      + },
    }
  }
}
```

Рисунок 2.2 – Приклад розширення колірної схеми з документації

Ще одним інструментом який надає Tailwind CSS для конфігурації кольорів є градації. Градації кольорів – дають можливість використовувати

різні відтінки одного кольору для додання візуальної глибини та акцентування елементів на вашому сайті. Типово, кожен колір має градації від 50 до 900, де 50 є найсвітлішим, а 900 – найтемнішим (див. рис. 2.3).[6]

```

module.exports = {
  theme: {
    extend: {
      colors: {
        primary: {
          50: '#f3f4f6', // Світлий відтінок для фону
          100: '#e1e1e1', // Трохи темніший для фону
          200: '#cfcfd2', // Нормальний
          300: '#b1b1b5', // Звичайний
          400: '#8e8e93', // Затемнений для тексту
          500: '#6e6e73', // Основний колір
          600: '#4a4a4f', // Затемнений для тексту
          700: '#2c2c2e', // Ще темніший для акцентів
          800: '#1a1a1a', // Дуже темний для акцентів
          900: '#0d0d0d' // Найтемніший для акцентів або фону
        }
      }
    }
  }
}

```

Рисунок 2.3 –Градації кольорів

Градації ми можемо використати для таких цілей:

- для візуальної ієрархії та створення глибини контенту;
- для кастомізації бренду, створюючи унікальний та однорідний вигляд сайту або застосунку.

Для генерації колірної палітри використаємо один з онлайн сервісів. Щоб успішно скористатись таким сервісом нам слід обрати основний колір. В нашому випадку це колір – midnight-blue (#003366), основний колір Запорізького національного університету. В результаті роботи інструменту генерації палітри ми отримали колірні коди 11 градацій (див. табл. 2.1).

Таблиця 2.1 – Згенерована колірна палітра

Значення градації	Колірний код	Клас для кольору тексту	Клас для кольору фону
50	#eef7ff	text-midnight-blue-50	bg-midnight-blue-50
100	#dceeff	text-midnight-blue-100	bg-midnight-blue-100
200	#b2dfff	text-midnight-blue-200	bg-midnight-blue-200
300	#6dc6ff	text-midnight-blue-300	bg-midnight-blue-300
400	#20a8ff	text-midnight-blue-400	bg-midnight-blue-400
500	#008dff	text-midnight-blue-500	bg-midnight-blue-500
600	#006edf	text-midnight-blue-600	bg-midnight-blue-600
700	#0057b4	text-midnight-blue-700	bg-midnight-blue-700
800	#004a94	text-midnight-blue-800	bg-midnight-blue-800
900	#003366	text-midnight-blue-900	bg-midnight-blue-900
950	#002651	text-midnight-blue-950	bg-midnight-blue-950

2.1.2 Конфігурація адаптивності

Адаптивність реалізується через медіа-запити, які вже будовані в стандартні атомарні класи фреймворку. Це означає, що ви зможете легко застосовувати різні стилі (марджини, падінги, шрифт і т.д.) до елементів залежно від розміру екрану.

Tailwind CSS використовує мобільну стратегію «first», яка полягає в першочерговому створенні дизайну для мобільних пристроїв. Тому розробники які працюють з фреймворком повинні орієнтуватись на мобільні пристрої, а потім на більші розміри екрану. І для реалізації адаптивності в Tailwind вже реалізовані стандартні класи адаптивності (див. табл. 2.2).

Таблиця 2.2 – Стандартні значення адаптивності

Мінімальна ширина екрану	Утилітарний клас	Тип пристрою
640px	sm	Смартфони
768px	md	Таблети
1024px	lg	Менші десктопи
1280px	xl	Десктопи
1536px	2xl	Великі десктопи

Але такий підхід до стандартних класів виключає з переліку підтримуваних пристроїв великі монітори та розкладні смартфони через їх специфічні розміри екрану. Тому додаємо класи та точки зупинки для підтримки широких моніторів(див. табл. 2.3).

Таблиця 2.3 – Додані значення адаптивності

Мінімальна ширина екрану	Утилітарний клас	Тип пристрою
1920px	3xl	Для широких десктоп моніторів
2560px	4xl	Для моніторів 2К
3840px	5xl	Для моніторів 4К

І зможемо використовувати їх таким чином (див рис.2.4)

```
<div class="text-base 3xl:text-lg 4xl:text-xl 5xl:text-2xl">
  <p>Цей текст збільшиться відповідно до розміру екрану.</p>
</div>
```

Рисунок 2.4 –Приклад використання нових класів адаптивності

Зі складними смартфонами ситуація трошки складніша. Наприклад, Galaxy Fold має основний екран з роздільною здатністю приблизно 1536x2152

пікселів у розгорнутому стані. Для налаштування таких медіа запитів слід використати режим роботи зі стилями CSS на пряму (див. рис. 2.5).

```
module.exports = {
  theme: {
    extend: {
      screens: {
        'fold': {'raw': '(width: 280px) and (height: 653px)'}, // Samsung Galaxy Z Flip
        'fold-lg': {'raw': '(width: 585px) and (height: 773px)'}, // Samsung Galaxy Fold
      }
    }
  }
}
```

Рисунок 2.5 – Адаптивні класи для складних смартфонів

Використовувати їх ми можемо наступним чином (див. рис. 2.6).

```
<div class="bg-blue-500 fold:bg-green-500 fold-lg:bg-red-500">
  <p class="p-4 fold:p-2 fold-lg:p-8">Змінний відступ залежно від розміру екрану.</p>
  <h1 class="text-base fold:text-sm fold-lg:text-lg">Змінний розмір шрифту для розкладних смартфонів.</h1>
</div>
```

Рисунок 2.6 – Приклад використання класів адаптивності для складаних пристроїв

Розкладні пристрої часто змінюють орієнтацію використання зі складеного до розгорнутого стану, тому деякі стилі потрібно буде адаптувати залежно від орієнтації екрану (портретна чи ландшафтна) (див. рис. 2.7).

```
@media (orientation: portrait) {
  .portrait\:text-lg { font-size: 1.125rem; }
}

@media (orientation: landscape) {
  .landscape\:text-lg { font-size: 1.125rem; }
}
```

Рисунок 2.7 – Налаштування відповідно до орієнтації

Врахуємо також декілька з кращих практик адаптивного дизайну. Одна з них мінімалістичний підхід – завдяки якому ми не будемо перевантажувати майбутню тему занадто багатьма медіа-запитами, а будемо використовувати їх там, де це дійсно необхідно. При обранні оптимального розміру шрифту будемо використовувати інструменти тестування сайту на різних пристроях та орієнтаціях для забезпечення доступності та читабельності інформації [6].

Застосування цих принципів та налаштувань дозволить вам створити більш гнучкий, адаптивний веб-сайт, який ефективно працюватиме на різноманітних пристроях, покращуючи загальний користувацький досвід (див. рис. 2.8).

```
module.exports = {
  theme: {
    extend: {
      screens: {
        xs: '480px', // Додаємо дуже маленький розмір для особливо маленьких пристроїв
        sm: '640px',
        md: '768px',
        lg: '1024px',
        xl: '1280px',
        '2xl': '1536px',
        '3xl': '1920px', // Для широких десктоп моніторів
        '4xl': '2560px', // Для моніторів 2K
        '5xl': '3840px',
        'fold': {'raw': '(width: 280px) and (height: 653px)'}, // Samsung Galaxy Z Flip
        'fold-lg': {'raw': '(width: 585px) and (height: 773px)'}, // Samsung Galaxy Fold
      },
    },
  }
}
```

Рисунок 2.8 – Фінальна конфігурація адаптивності

2.1.3 Проєктування компонентів

Для проєктування теми в Drupal важливо зрозуміти, як працює його шаблонізатор. Drupal використовує систему шаблонізації Twig, яка є гнучкою і потужною. Ось основні аспекти, які слід врахувати:

- twig templates: Twig є головним механізмом шаблонізації в Drupal 8 і новіших версіях. Це система шаблонів, що дозволяє вам створювати HTML-структури за допомогою змінних, фільтрів та операторів управління. Кожен шаблон Twig в Drupal має розширення файлу `.html.twig`;

- шаблонна ієрархія: Drupal має детально визначену ієрархію шаблонів, де базові шаблони можуть бути перевизначені спеціалізованими версіями для конкретних типів вмісту, полів, блоків або виглядів. Це дозволяє вам легко керувати відображенням вмісту без зміни функціональності базових компонентів;

- перемінні шаблону: Передача даних у шаблони Twig відбувається через змінні. Ці змінні можуть містити весь необхідний контент, поля, налаштування тощо, які потім можуть бути використані у шаблонах для відображення потрібної інформації;

- функції та фільтри Twig: Twig надає широкий спектр функцій і фільтрів для модифікації виводу, таких як форматування дат, локалізація тексту, маніпуляції з рядками тощо. Це суттєво спрощує процес візуалізації даних в шаблоні;

- тема й підтеми: Ви можете створити власну тему з нуля або розробити підтему на базі існуючої. Підтеми наслідують всі стилі, шаблони і JavaScript від батьківської теми, але дозволяють вносити свої власні зміни та доповнення;

- робота з Assets: Включення CSS і JavaScript в тему Drupal керується системою бібліотек, яка дозволяє вам визначити, коли і які стилі та скрипти повинні бути додані до сторінок.[5]

Враховуючи особливості шаблонізатору можемо виділити наступні компоненти для реалізації у темі:

- сторінка;

1. макет сторінки (layout);

- меню;
- дворівневе меню;

- стандартна головна секція (hero section);
- секція контенту;
- секція заклику до дії;
- секція команди;
- підвал сайту;
- соціальна секція;
- секція пов'язаних розділів;

2. адміністративний шаблон

- бічна навігація;
- форма та її поля;
- модальне вікно;
- елементи керування таблицею;
- таблиця;
- пагінація;
- панель інтерактивного звіту;
- панель керування.

Результатом проектування схеми компонентів теми для системи керування контентом Drupal став перелік що повністю охоплює всі компоненти, що можуть бути використані при розробці та імплементації власних сторінок, як публічних, так і закритих від загалу.

3 РОЗРОБКА ТЕМИ

3.1 Необхідні інструменти

Для ефективної роботи з Tailwind CSS та Drupal і розробки нової теми необхідно мати набір ключових технологій та інструментів.

3.1.1 Tailwind CSS інструменти

Перш за все нам треба середовище в якому буде виконуватись збірка стилів нашої теми. Ми будемо використовувати інструмент який став вже стандартом для застосунків та інструментів які працюють на мові Javascript – Node.js.

Node.js – це серверна платформа, яка дозволяє виконувати JavaScript код на сервері, а не в браузері. Це основа для багатьох інструментів розробки веб-додатків, включаючи систему збірки проектів на базі Tailwind CSS.

Пакетним менеджером який буде встановлювати залежності для нашої теми буде npm.

npm (Node Package Manager) – це менеджер пакетів, який дозволяє вам легко встановлювати, оновлювати та управляти залежностями (бібліотеками, фреймворками тощо) у вашому проекті. Він використовується для встановлення Tailwind CSS та інших потрібних пакетів.

Для встановлення даного набору інструментів заходимо на офіційний сайт Node та просто завантажуюмо інсталятор для нашої операційної системи.

Після того як середовище встановлено обираємо інструмент який буде виконувати завдання з автоматизації задач, таких як мініфікація CSS і JavaScript та hot reloading.

Ми обрали Vite. Vite – це інструмент, який використовує сучасні можливості браузерів для імпорту модулів, що дозволяє відмовитися від

традиційних збірок і використовувати код належним чином практично без затримок.

3.1.2 Drupal середовище

Система Drupal працює на сервері, тому для локальної розробки нам слід налаштувати серверне середовище на наших машинах. Є декілька варіантів встановлення:

- LAMP (Linux, Apache, MySQL, PHP);
- WAMP (Windows, Apache, MySQL, PHP);
- MAMP (Mac, Apache, MySQL, PHP);
- LEMP (Linux, Nginx, MySQL, PHP).

Ми оберемо 2 варіанти для адаптивності та тестування яке буде покривати більшу частину випадків. Наше середовище буде включати:

- Linux, Windows, як операційну систему;
- Apache як вебсервер;
- MySQL як систему управління базами даних;
- PHP – мова програмування, на якій написаний Drupal.

Drupal – має специфічні вимоги до версій PHP. Тому використовуємо більш нову версію 8.0. Після інсталяції PHP інсталуємо composer з офіційного сайту (див. рис. 3.1).[4]

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === 'dac665fdc30fdd8ec78b38b9800061b4150413ff2e3b6f88543c
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

Рисунок 3.1 – Сценарій для інсталяції composer

Одразу після composer встановлюємо командний рядок який надасть нам скриптовий інтерфейс для керування Drupal – Drush.

Через Drush ми можемо виконувати завдання адміністрування, оновлення бази даних, імпорт та експорт конфігурацій і т.д.

3.1.3 Редактор коду

Редактором коду на який пав наш вибір став PHPStorm.

PHPStorm – це потужний інтегрований середовище розробки (IDE) від JetBrains, спеціально призначений для PHP, який також пропонує великі можливості для розробки на Drupal. PHPStorm підтримує PHP, HTML, CSS, JavaScript і має розширені функції для роботи з базами даних і версійним контролем.

Особливості PHPStorm для роботи з Drupal:

- Підтримка Drupal: PHPStorm має вбудовану підтримку Drupal, що полегшує інтеграцію та розробку;
- Інтеграція з Composer і Drush: Легке використання цих інструментів прямо з IDE;
- Відладка: Інтеграція з Xdebug і Zend Debugger для зручної відладки PHP коду;
- Автодоповнення коду: Вдосконалена підтримка PHP, включаючи Drupal API, що значно спрощує написання коду;
- Рефакторинг: Могутні інструменти для рефакторингу PHP коду, що допомагають підтримувати кодову базу організованою та чистою;
- Інтеграція з системами контролю версій: Підтримка Git, SVN, Mercurial і інших.

3.1.4 Система контролю версій

Системою контролю версій було обрано Bitbucket. Bitbucket є популярним веб-сервісом для управління репозиторіями, що підтримує

системи контролю версій Git та Mercurial. Він надає командам інструменти для співпраці над кодом в рамках приватних або публічних проєктів.

Основні особливості Bitbucket:

– Приватні репозиторії: На відміну від деяких інших платформ, Bitbucket надає необмежену кількість приватних репозиторіїв безкоштовно, що робить його ідеальним вибором для стартапів та малих команд.

– Інтеграція з Jira та Trello: Bitbucket глибоко інтегрований з іншими продуктами Atlassian, такими як Jira (для відстеження завдань та управління проєктами) та Trello (для управління завданнями в форматі канбан-дошок).

– Pull Requests та Code Reviews: Bitbucket сприяє культурі code review через систему pull requests, що дозволяє розробникам обговорювати зміни перед їх злиттям у головний код.

– CI/CD через Bitbucket Pipelines: Bitbucket Pipelines є вбудованим інструментом для неперервної інтеграції та неперервної доставки (CI/CD), що дозволяє автоматизувати тести, збірки та розгортання з допомогою конфігурації у форматі YAML.

– Безпека: Bitbucket забезпечує ряд функцій безпеки, таких як двофакторна аутентифікація, обмеження доступу на основі IP-адрес та шифрування на стороні сервера.

На даному етапі було створено 2 репозиторія. Перший містить основні налаштування системи керування контентом Drupal. А другий містить безпосередню вихідні файли нашої теми (див. рис.3.2).



Рисунок 3.2 – Створені репозиторії

3.1.5 Засоби контейнеризації

В якості засобу для роботи з контейнером став вже стандартний для галузі інструмент – Docker (див. рис. 3.3).

```
# Вибір базового образу
FROM php:7.4-apache

# Встановлення необхідних пакетів
RUN apt-get update && apt-get install -y \
    libpng-dev \
    libjpeg-dev \
    libfreetype6-dev \
    libwebp-dev \
    libmcrypt-dev \
    libzip-dev \
    zip \
    git \
    curl \
    && docker-php-ext-configure gd --with-freetype --with-jpeg --with-webp \
    && docker-php-ext-install gd pdo pdo_mysql zip

# Включення mod_rewrite для Apache
RUN a2enmod rewrite

# Копіювання існуючого коду Drupal в контейнер
COPY . /var/www/html/

# Встановлення Composer
COPY --from=composer:latest /usr/bin/composer /usr/bin/composer

# Встановлення прав доступу
RUN chown -R www-data:www-data /var/www/html

# Відкриття порту 80
EXPOSE 80
```

Рисунок 3.3 – Налаштування контейнеру

Для оркестрації іншими контейнерами які необхідні для виконання завдання та досягнення мети кваліфікаційної роботи було обрано Docker-Compose (див. рис. 3.4).

```
version: '3.8'
services:
  web:
    build: .
    ports:
      - "8888:88"
    volumes:
      - ./var/www/html/
    depends_on:
      - db
    environment:
      VIRTUAL_HOST: drupal.local
      VIRTUAL_PORT: 88

  db:
    image: mysql:5.7
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: drupal
      MYSQL_USER: drupal
      MYSQL_PASSWORD: drupal
    volumes:
      - db_data:/var/lib/mysql

  vite:
    image: node:14
    volumes:
      - ./app
    working_dir: /app/web/themes/custom/mytheme
    command: sh -c "npm install && npm run dev"
    ports:
      - "3000:3000"

volumes:
  db_data:
```

Рисунок 3.4 – Файл конфігурації Docker Compose

3.2 Налаштування теми

Першим етапом у створенні теми є файл *.info.yml який визначає властивості теми (див. рис. 3.5).

```

name: My Tailwind Theme
type: theme
description: 'A custom Drupal theme using Tailwind CSS and Vite.'
core_version_requirement: ^8 || ^9
base theme: stable
package: Custom

# Визначення регіонів
regions:
  header: Header
  content: Content
  sidebar: Sidebar
  footer: Footer

# Підключення бібліотек
libraries:
  - mytheme/global-styling

# Файли JavaScript та CSS будуть додані через Vite

```

Рисунок 3.5 – tailwind.info.yaml

Далі опишемо файл який надасть нам можливість інтеграції Javascript та CSS в нашу тему. Такий файл має закінчуватись ключовими словами .libraries.yaml та знаходитись в тій же папці що й попередній файл (див. рис. 3.6).

```

global-styling:
  version: 1.x
  js:
    assets/build/main.js: { type: external, minified: true }
  css:
    theme:
      assets/build/main.css: { type: external, minified: true }

```

Рисунок 3.6 – tailwind.libraries.yaml

Також налаштуємо конфігурацію Vite, щоб в результаті збирання наших стилів та файлів логіки вони потрапляли в директорію вказану у налаштуваннях та підключеннях теми (див. рис. 3.7).

```
import { defineConfig } from 'vite';

export default defineConfig({
  base: '.', // Важливо для коректної інтеграції з Drupal
  build: {
    outDir: 'assets/build',
    rollupOptions: {
      input: '/path/to/your/entry/file.js'
    }
  }
});
```

Рисунок 3.7 – vite.config.js

3.3 Реалізація компонентів

У файлі tailwind.libraries.yaml ми визначили регіони. В них ми можемо розміщувати компоненти які визначили на етапі проектування. Наприклад у регіоні header ми можемо розмістити компонент меню (див. рис. 3.8 та Додаток А).

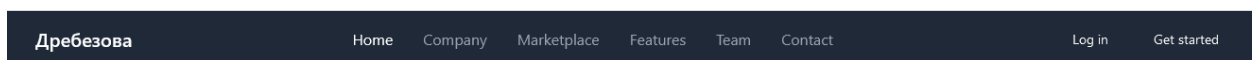


Рисунок 3.8 – Меню на сторінці

Наступні компоненти відповідно будуть відображатись в регіонах виділених для них (див. Додаток Б).

Також була розроблена адміністративна частина теми для Drupal (див. рис. 3.9).



Рисунок 3.9 – Адміністративна частина теми з динамічними полями

Загалом в результаті розробки було створено 20 компонентів для реалізації різноманітних інтерфейсів.

ВИСНОВКИ

У сучасному світі, де цифрові технології розвиваються надзвичайно швидко, а обсяг домашньої роботи збільшується, важливість використання новітніх і зручних інструментів стає все більш актуальною. Особливий прогрес спостерігається в області веб-розробки, зокрема в клієнтській частині, де Tailwind CSS виявився визначним нововведенням. Цей фреймворк відрізняється своєю оптимальною швидкістю завантаження та гнучкістю, що значно підвищує ефективність розробки. Водночас, технології, які були популярні раніше, наприклад Drupal, також оновлюються та залишаються в тренді в корпоративному секторі та освіті. Це дослідження має за мету інтеграцію перевіреної часом технології з новітніми тенденціями для створення нової, гнучкої теми для Drupal, розробленої на базі Tailwind CSS. Завдяки цьому проекту, користувачі отримають змогу легше створювати та адаптувати вебсайти, що відповідають сучасним стандартам швидкості та адаптивності. Основні досягнення проекту включають:

- Швидкість завантаження: Оптимізація Tailwind CSS забезпечує високу швидкість завантаження, поліпшуючи користувацький досвід та позитивно впливаючи на SEO;

- Адаптивність: Розроблена тема адаптується під різні пристрої, що спрощує процес створення сайтів, які виглядають добре на будь-якому обладнанні;

- Легкість кастомізації: Tailwind CSS дозволяє швидко та легко змінювати стилі, що допомагає користувачам Drupal налаштувати дизайн відповідно до своїх потреб.

Цей проект відображає ефективне поєднання довіреності та інновацій, відкриваючи нові перспективи для розробників та кінцевих користувачів, і розширює можливості веброзробки.

ПЕРЕЛІК ПОСИЛАНЬ

1. Geetha, S., Shasvat, Shasvat., & Shah, Harsh. Web development techniques: Comparisons between traditional coding and CMS platforms. ScienceGate, 2021, pp. 212-216. URL: <http://www.sciencegate.app/web-development-techniques>.
2. Britto Pichiliani, Talita Cristina Pagani, & Pizzolato, Ednaldo Brigante. Cognitive disabilities and web accessibility: A survey in the Brazilian web development community. Journal on Interactive Systems, 2021, Vol. 12 (1), pp. 308-327. URL: <http://www.jis.com/cognitive-disabilities-web-accessibility>.
3. Bright Side of Life. "Drupal 9 Development Using Tailwind CSS". Available online: www.bright-side-of-life.com/drupal-9-development-using-tailwind-css (accessed on 01 February 2024).
4. Drupal Sun. "How to Use Tailwind CSS in Drupal for Atomic Web Design". Available online: www.drupalsun.com/how-to-use-tailwind-css-in-drupal (accessed on 27.02.2024).
5. Sipos Daniel. Drupal 10 Module Development: Develop and deliver engaging and intuitive enterprise-level apps, Packt Publishing, 2023. — 580 p.
6. Bhat Kartik. Ultimate Tailwind CSS Handbook: Build sleek and modern websites with immersive UIs using Tailwind CSS, Orange Education Pvt Ltd., 2023. — 293 p.


```

01-1-1zM3 15a1 1 0 011-1h12a1 1 0 110 2H4a1 1 0 01-1-1z" clip-
rule="evenodd"></path></svg>
  </button>
</div>
<div class="hidden w-full items-center justify-between lg:order-1 lg:flex lg:w-
auto" id="mobile-menu-2">
  <ul class="mt-4 flex flex-col font-medium lg:mt-0 lg:flex-row lg:space-x-8">
    {% for item in menu_items %}
      <li>
        <a href="{{ item.url }}" class="{% if item.active %}bg-primary-700
lg:text-primary-700{% else %}lg:hover:text-primary-700{% endif %} block
rounded py-2 pl-3 pr-4 text-gray-700 hover:bg-gray-50 lg:border-0 lg:p-0
lg:hover:bg-transparent dark:text-gray-400 dark:hover:bg-gray-700
dark:hover:text-white lg:dark:hover:bg-transparent lg:dark:hover:text-white">{{
item.name }}</a>
      </li>
    {% endfor %}
  </ul>
</div>
</nav>
</header>

```

ДОДАТОК Б

Компонент коментарів

