

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ ВИЩИЙ
НАВЧАЛЬНИЙ ЗАКЛАД «УНІВЕРСИТЕТ «КРОК»**

Фаховий коледж Університету «КРОК»

ДИПЛОМНА РОБОТА

за темою

«Розробка та створення сайту для перегляду спільного відео»

Студент 4 курсу групи ШЗ-20к/2

Керівник дипломної роботи

Удовенко Артем Юрійович

(прізвище, ім'я та по-батькові студента)

(посада керівника)

Кириченко Віктор Вікторович

(прізвище, ім'я та по-батькові керівника)

До захисту

(резолуція «До захисту»)



(підпис студента)

10.06.2024

(дата)



(підпис викладача)

Київ, 2024 рік

Зміст

1. Вступ.....	3
РОЗДІЛ 1. ТЕОРИТИЧНІ ЗНАННЯ ПО ВЕБ-САЙТАМ, ЇХ ТИПИ ТА ІНСТРУМЕНТИ ЯКІ ВИКОРИСТОВУЮТЬСЯ ДЛЯ СТВОРЕННЯ САЙТУ ...	4
1.1. Основи веб-сайтів: що це таке і як вони працюють.....	4
1.2. Популярні типи веб-сайтів якими користуються всі.....	5
1.3. Інструменти які використовуються для розробки сайтів.....	6
РОЗДІЛ 2. ОСНОВНІ ПРАВИЛА ДО СТВОРЕННЯ ВЕБ-САЙТУ ЯКИХ СЛІД ПРИТРИМУВАТИСЬ.....	16
2.1 Систему керувань веб-контентом. Як вона пов'язана із створенням сайту.....	16
2.2 Основи створення сайту. Які програми для цього використовуються.....	18
2.3 Перелік інструментів та програм які застосовуються для оформлення сайту.....	27
РОЗДІЛ 3. РОЗРОБКА САЙТУ З ЗАСТОСУВАННЯМ ДЕЯКИХ ПРАВИЛ ТА ІНСТРУМЕНТІВ.....	32
3.1 Як використовують система керувань веб-контентом в створенні сайту.....	32
3.2 Порівняння створення сайтів з системою керувань веб-контентом та без неї. Яка була застосована в проекті.....	37
Висновок.....	46
Література.....	47
Додаток	48

Вступ

У світі, де цифрові технології набувають все більшого значення, створення та підтримка веб-присутності стає невід'ємною частиною бізнесу та особистого розвитку. Створення сайту за допомогою систем керування веб-контентом (CMS) стає одним з найефективніших і швидких способів впорядкування вмісту і представлення інформації в Інтернеті.

CMS - це потужні інструменти, які дозволяють навіть тим, хто не має глибоких технічних знань, легко створювати, редагувати та управляти веб-сайтами. Вони надають можливість швидко розгортати сайти, додавати новий контент і змінювати дизайн без необхідності великих витрат часу та коштів.

У цьому доповіді ми розглянемо переваги використання CMS для створення веб-сайтів, розглянемо основні функції таких систем, а також проаналізуємо деякі найпопулярніші платформи для керування веб-контентом.

РОЗДІЛ 1 ТЕОРИТИЧНІ ЗНАННЯ ПО ВЕБ-САЙТАМ, ЇХ ТИПИ ТА ІНСТРУМЕНТИ ЯКІ ВИКОРИСТОВУЮТЬСЯ ДЛЯ СТВОРЕННЯ САЙТУ

1.1 Основи веб-сайтів: що це таке і як вони працюють

Веб-сайт - це електронна платформа, що складається з різних веб-сторінок, що

зазвичай включають текст, графіку, мультимедійні елементи та інші форми контенту. Вони доступні через Інтернет та призначені для перегляду за допомогою веб-браузера. Веб-сайти можуть мати різноманітні цілі та функціональність, такі як надання інформації про певну тему (наприклад, новини, освіта, туризм), взаємодія з користувачами (через блоги, форуми, соціальні мережі), електронна комерція (онлайн-магазини), або надання послуг (банківські, гуртожитки). Структура веб-сайту може включати різноманітні елементи, такі як головна сторінка, меню навігації, сторінки з контентом, форми зв'язку, а також скрипти та програмне забезпечення для забезпечення інтерактивності та функціональності. Розробка веб-сайтів вимагає знань інтернет-технологій, таких як HTML, CSS, JavaScript, а також може включати в себе роботу з веб-дизайном, веб-розробкою, веб-хостингом та іншими аспектами веб-індустрії.[1]

Важливо зазначити, що сайти можуть бути статичними або динамічними. Статичні сайти складаються з фіксованих веб-сторінок, які відображаються однаково для всіх відвідувачів і не мають можливості інтерактивності. Динамічні ж сайти генерують сторінки "на льоту", використовуючи бази даних і скрипти, що дозволяють користувачам взаємодіяти з контентом, залишати коментарі, заповнювати форми, тощо.

1.2 Популярні типи веб-сайтів якими користуються всі

Інформаційні сайти: Це найпоширеніший тип сайтів, що призначені для надання користувачам інформації на певну тему. Це можуть бути новинні портали, сайти компаній з надання послуг, блоги, онлайн-журнали тощо.

Електронна комерція: Онлайн-магазини та інші сайти з електронною комерцією дозволяють користувачам здійснювати покупки через Інтернет. Вони надають можливість переглядати товари, здійснювати покупки та оплачувати їх онлайн.

Освітні та навчальні ресурси: Ці сайти надають користувачам доступ до різноманітного навчального контенту, такого як онлайн-курси, відеоуроки, електронні книги тощо.

Портали та форуми: Вони збирають в собі велику кількість інформації на різні теми, дозволяють користувачам обмінюватися думками, досвідом і враженнями.

Портфоліо та особисті сайти: Ці сайти дозволяють людям представляти свою роботу, досягнення або особисту інформацію. Їх використовують художники, фотографи, письменники та інші творчі особистості.

Інформаційні сайти: Ці сайти призначені для надання користувачам інформації на певну тему. Вони можуть бути новинними порталами, корпоративними веб-сайтами компаній, блогами, відомчими або освітніми ресурсами тощо.

Інформаційні сайти зазвичай містять статичний контент, який змінюється не так часто, і призначені для того, щоб відвідувачі отримували актуальну інформацію.

1.3 Інструменти які використовуються для розробки сайтів

Популярні інструменти:

HTML є основою будь-якого веб-сайту і вважається одним із найпоширеніших інструментів для створення веб-сторінок.

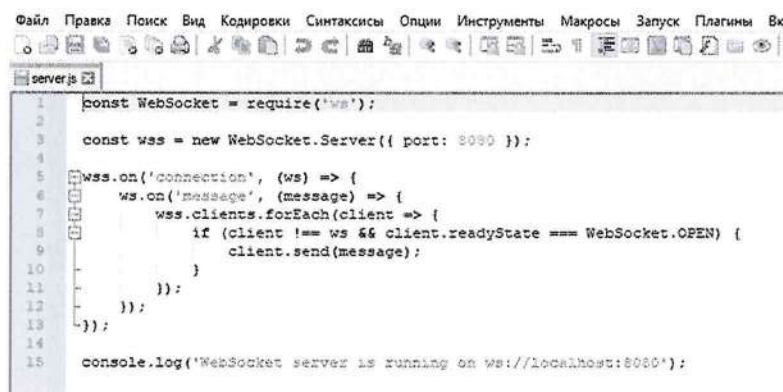
CSS також широко використовується разом з HTML для стилізації веб-сторінок і є стандартом для оформлення.

JavaScript - це одна з найпопулярніших мов програмування, яка використовується для створення інтерактивних веб-сайтів і додатків.

Bootstrap є одним з найпопулярніших фреймворків для розробки веб-сайтів, оскільки він пропонує швидке та просте створення стильних і відзивчивих інтерфейсів.

React.js став вельми популярним завдяки своїй ефективності та зручності у розробці інтерфейсів користувача. Він широко використовується для створення високопродуктивних веб-додатків.

Node.js став дуже популярним серед розробників за останні кілька років завдяки своїй ефективності і можливості використання JavaScript на серверному боці. [2]

A screenshot of a code editor window titled 'server.js'. The code defines a WebSocket server on port 8080. It uses the 'ws' library to create a server and listens for connections. When a connection is established, it listens for messages and broadcasts them to all other clients. A log message is printed to the console when the server starts.

```
1 const WebSocket = require('ws');
2
3 const wss = new WebSocket.Server({ port: 8080 });
4
5 wss.on('connection', (ws) => {
6   ws.on('message', (message) => {
7     wss.clients.forEach(client => {
8       if (client !== ws && client.readyState === WebSocket.OPEN) {
9         client.send(message);
10      }
11    });
12  });
13 });
14
15 console.log('WebSocket server is running on ws://localhost:8080');
```

Рисунок 1.1 - приклад коду який використовувався для написання в цій програмі

Менш популярні інструменти:

Хоча **jQuery** був дуже популярним у минулому, останнім часом його популярність трохи зменшилась через появу сучасних фреймворків, таких як **React** та **Vue.js**, які пропонують більш сучасний підхід до розробки веб-інтерфейсів.

Express.js є дуже корисним інструментом для розробки серверної логіки на Node.js, але в порівнянні з Node.js він може бути менш популярним, оскільки він є лише фреймворком на верхньому рівні, що забезпечує лише додаткові зручності у порівнянні з самим Node.js.

MongoDB - це дуже популярна NoSQL база даних, але в порівнянні з традиційними реляційними базами даних, такими як MySQL або PostgreSQL, її застосування може бути обмеженим у деяких випадках, особливо коли потрібна строга структура даних.

Хоча деякі інструменти можуть бути менш популярними зараз, вони все ще можуть бути корисними в певних сценаріях розробки веб-додатків.

HTML використовується для створення структури веб-сторінок. Він визначає елементи, такі як заголовки, абзаци, списки, таблиці, посилання і форми, за допомогою тегів. HTML відповідає за те, що користувач бачить у своєму браузері.

CSS відповідає за вигляд і оформлення веб-сторінок, які створені за допомогою HTML. Він дозволяє задавати кольори, шрифти, розміри, відступи, вирівнювання і інші аспекти дизайну сторінки. CSS використовується для створення зовнішнього вигляду веб-сайту.

JavaScript є мовою програмування, яка використовується для надання динамічності веб-сторінкам. Він дозволяє створювати інтерактивні елементи,

виконувати обробку подій, взаємодіяти з користувачем і виконувати запити до сервера без перезавантаження сторінки.

Bootstrap - це HTML, CSS і JavaScript-фреймворк для швидкого та простого створення відзивчивих веб-інтерфейсів. Він містить готові компоненти і стилі, які дозволяють розробникам швидко створювати стильні веб-сторінки.

jQuery - це швидкий, легкий і багатофункціональний JavaScript-фреймворк, який спрощує взаємодію з DOM, обробку подій, анімацію і роботу з AJAX. Він робить JavaScript-код коротшим і зручнішим для написання.

React.js є бібліотекою JavaScript для створення інтерфейсів користувача. Вона використовує компонентний підхід до розробки, що дозволяє розбити інтерфейс на незалежні компоненти. React забезпечує ефективне оновлення і відображення змін у стані додатку.

Node.js - це середовище виконання JavaScript, яке дозволяє запускати JavaScript-код на сервері. Він побудований на движку V8 Chrome і дозволяє розробникам створювати швидкі і масштабовані веб-додатки.

Express.js - це мінімалістичний фреймворк для розробки веб-додатків на платформі Node.js. Він дозволяє розробникам швидко створювати маршрути, обробники запитів і реалізовувати інші функції для створення серверної логіки.

MongoDB - це документ-орієнтована NoSQL база даних. Вона використовує JSON-подібні документи для зберігання даних, що робить її особливо підходящою для розробки веб-додатків з гнучкими схемами даних.

Існує багато інструментів для створення бекенду сайту для перегляду фільмів. Вибір конкретного залежить від ваших потреб, рівня технічних знань та здатностей. Ось кілька популярних варіантів:

Django: Django - це високорівневий веб-фреймворк на мові програмування Python. Він забезпечує швидкий старт для створення потужних веб-додатків, включаючи веб-сайти для стрімінгу фільмів.

Ruby on Rails - Це інший високорівневий веб-фреймворк, який дозволяє швидко розробляти веб-додатки. Використання Ruby on Rails може бути особливо корисним, якщо ви вже знайомі з мовою програмування Ruby.

Express.js - Це легкий фреймворк для створення серверних додатків на Node.js. Він є дуже популярним в спільноті JavaScript і дозволяє швидко створювати RESTful API.

Flask - це ще один легкий веб-фреймворк на мові програмування Python. Він дозволяє створювати веб-сайти і веб-додатки, включаючи бекенд для стрімінгу фільмів, з мінімальними зусиллями.

Firebase - це платформа від Google, яка надає широкі можливості для розробки веб-додатків, включаючи аутентифікацію користувачів, зберігання даних та хостинг. Ви можете використовувати Firebase для створення бекенду для веб-сайту зі стрімінгом фільмів.

Ці інструменти мають свої переваги і недоліки, тому вибір залежить від вашого досвіду, потреб проекту та уподобань у програмуванні.

Переглянемо які саме є переваги та недоліки цих програм.

Django:

Плюси:

- Швидкий розвиток завдяки широкому спектру вбудованих функцій і пакетів.
- Велика активна спільнота, що допомагає вирішувати проблеми та надає підтримку.

- Гнучкість і розширюваність за допомогою сторонніх бібліотек і розширень.

Мінуси:

- Вимагає певного рівня знань Python.
- Призначений для складних веб-додатків, тому може бути зайвим для простих проектів.

Ruby on Rails:

Плюси:

- Швидкий розвиток завдяки конвенціям над конфігурацією та великої кількості готових рішень.
- Добра підтримка і документація спільноти.
- Можливість використовувати багато готових розширень.

Мінуси:

- Потребує знань мови програмування Ruby.
- Іноді може виникнути проблема зі сумісністю версій розширень.

Express.js:

Плюси:

- Легко вивчити, особливо для тих, хто вже знайомий з JavaScript.
- Гнучкість у побудові веб-серверів і API.
- Велика кількість розширень та пакетів для різних завдань.
- Мінуси:
- Якщо не врахувати правильну організацію коду, проект може стати заплутаним.

- Відсутність стандартних засобів для багатьох задач, що потребує використання сторонніх бібліотек.

Flask:

- Плюси:
- Простота вивчення і використання, особливо для початківців.
- Гнучкість у розробці веб-сайтів та веб-додатків різного рівня складності.
- Ідеальний для невеликих і середніх проектів.

Мінуси:

- Менш функціональний порівняно з Django або Ruby on Rails.
- Може потребувати встановлення більшої кількості додаткових бібліотек для покриття всіх потреб проекту.[3]

Firestore:

Плюси:

- Простота використання, особливо для розробників, які не мають досвіду з бекендом.
- Швидкий розгортання та масштабування.
- Широкий функціонал, включаючи базу даних, аутентифікацію, зберігання файлів тощо.

Мінуси:

- Обмежена гнучкість порівняно з власним розвитком бекенду.
- Залежність від інфраструктури Google і відповідно висока вартість підписки при великому обсязі даних.

HTML використовується для створення структури веб-сторінок. Ось HTML-код, який було використано для створення сайту

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Сумісний перегляд відео з Ютуба</title>
7    <link rel="stylesheet" href="style.css">
8
9  <style>
10     body {
11       margin: 0;
12       padding: 0;
13       height: 100vh;
14       display: flex;
15       justify-content: center;
16       align-items: center;
17       background: linear-gradient(200deg, #C0C0C0, #000000);
18       font-family: Arial, sans-serif;
19       color: #ffffff;
20       text-align: center;
21     }
22
23     h1 {
24       font-size: 3em;
25       margin: 0;
26     }
27
28     p {
29       font-size: 1.5em;
30       margin-top: 20px;
31     }
32
33     .container {
34       padding: 0px;
35       background: rgba(0, 0, 0, 0.3);
36       border-radius: 20px;
37     }
38   </style>
39 </head>
40 <body>
41   <nav>
42     <ul>
43       <li><a href="#" id="homeLink">Головна</a></li>
44       <li><a href="#" id="aboutLink">Про нас</a></li>
45       <li><a href="#" id="watchLink">Перегляд</a></li>
46     </ul>
47   </nav>
48

```

Рисунок 1.2 – HTML code

CSS використовується для стилізації веб-сторінок, надаючи їм вигляд та оформлення. Ось приклад CSS-коду, який було використано для створення сайту

```
server.js | index.html | style.css
1 body {
2   font-family: Arial, sans-serif;
3   background-color: #708090;
4   margin: 0;
5   padding: 0;
6 }
7
8 nav {
9   position: fixed;
10  top: 0;
11  left: 0;
12  width: 100%;
13  background-color: #333;
14 }
15
16 nav ul {
17  display: flex;
18  list-style-type: none;
19  margin: 0;
20  padding: 0;
21  justify-content: flex-start;
22 }
23
24
25 nav ul li {
26  display: inline;
27 }
28
29 nav ul li a {
30  display: block;
31  color: white;
32  text-align: center;
33  padding: 20px 16px;
34  text-decoration: none;
35 }
36
37 nav ul li a:hover {
38  background-color: #111;
39 }
40
41 .hidden {
42  display: none;
43 }
44
45 #content {
46  padding: 20px;
47  margin-top: 60px; /* Отступ сверху, чтобы контент не перекрывался навигацией */
48 }
49
50 #player {
51  width: 200%;
52  height: 450px;
53  background-color: #000;
54  margin: 20px auto;
55  margin-left: -250px;
56 }
```

Рисунок 1.3 CSS code

Оскільки код на пайтоні мені не знадобився я вирішив додати приклад як можна його використовувати в розробці сайту та як це виглядає.

Python також може використовуватися для веб-розробки, зазвичай у вигляді серверної мови програмування. Ось приклад простого веб-застосунку на Python з використанням фреймворку Flask:

```
from flask import Flask  
  
app = Flask(__name__)  
  
@app.route('/')  
def index():  
    return '<h1>Привіт, Я Python!</h1><p>Це мій перший веб-  
застосунок на Flask.</p>'  
  
if __name__ == '__main__':  
    app.run(debug=True)
```

JavaScript використовується для створення інтерактивності на веб-сторінках. Ось приклад JavaScript-коду, який змінює текст заголовка після завантаження сторінки:

```
document.addEventListener('DOMContentLoaded', function() {  
    var header = document.querySelector('h1');  
    header.textContent = 'Привіт, Я JavaScript!';  
});
```

Відмінності між мовами програмування

HTML: Використовується для створення структури веб-сторінок.

CSS: Використовується для стилізації веб-сторінок.

JavaScript: Використовується для створення інтерактивності на веб-сторінках.

Python: Може використовуватися для створення серверної логіки веб-застосунків та бізнес-логіки.

РОЗДІЛ 2. ОСНОВНІ ПРАВИЛА ДО СТВОРЕННЯ ВЕБ-САЙТУ ЯКИХ СЛІД ПРИТРИМУВАТИСЬ

2.1 Систему керувань веб-контентом. Як вона пов'язана із створенням сайту

Ці інструменти можуть використовуватися окремо або в поєднанні для створення високопродуктивних і функціональних веб-додатків.

Легкість використання і інтуїтивний інтерфейс: Багато CMS, такі як WordPress, Joomla, або Drupal, мають інтуїтивно зрозумілі інтерфейси, які дозволяють навіть не-технічним користувачам легко додавати, редагувати та видаляти контент на сайті. Це означає, що ви не потребуєте глибоких знань веб-програмування для управління сайтом.

Масштабованість: CMS дозволяє ефективно керувати великим обсягом контенту. Ви можете легко створювати нові сторінки, додавати медіафайли, створювати категорії та теги для організації контенту.

Шаблони та дизайн: CMS надають готові шаблони та теми, які дозволяють швидко створити привабливий дизайн для вашого сайту. Крім того, ви можете налаштувати ці шаблони згідно з вашими потребами або навіть створити власні.

Розширення та додатки: Багато CMS мають велику кількість розширень, додатків та модулів, які дозволяють розширити функціональність вашого сайту.

Це може бути все від додаткових інструментів SEO до системи аналізу відвідуваності.

Множинність користувачів та ролей: CMS надають можливість керування різними ролями користувачів і їх правами доступу до сайту. Наприклад, ви можете призначити адміністраторів, редакторів, авторів та інших користувачів з обмеженим доступом до адміністративної частини сайту.

Безпека: Багато CMS надають вбудовані механізми безпеки, такі як автоматичні оновлення, захист від вірусів та зловмисних атак, що дозволяє зберігати ваш сайт в безпеці.

Підтримка та спільнота: Багато CMS мають активні спільноти користувачів та розробників, які надають підтримку, поради та розв'язання проблем.

В цілому, CMS відіграють ключову роль у спрощенні процесу створення та управління веб-сайтами, дозволяючи вам швидко розвивати та підтримувати ваш веб-проект. Вони є потужним інструментом для будь-якого, хто хоче мати присутність в Інтернеті без великих зусиль.

Система керування веб-контентом (CMS) є інструментом, який дозволяє легко створювати, редагувати та управляти веб-сайтами без необхідності великих знань веб-програмування або верстки. CMS іграють важливу роль у процесі створення та управління сайтами, надаючи користувачам зручний інтерфейс для додавання та редагування контенту, керування медіафайлами, налаштування дизайну та багато іншого.

Ось деякі способи, які показують, де і як використовуються CMS у процесі створення сайту:

Створення контенту: CMS дозволяють вам створювати різноманітний контент, такий як статті, блоги, зображення, відео та інше. Ви можете легко додавати новий контент через адміністративний інтерфейс CMS без необхідності робити це через код.

Управління контентом: CMS надають інструменти для організації та управління вашим веб-контентом. Ви можете створювати категорії, теги, меню та інші структурні елементи, щоб легко організувати ваш сайт та полегшити навігацію для користувачів.

Редагування та публікація: CMS дозволяють легко редагувати та оновлювати контент на сайті. Ви можете змінювати текст, зображення, додавати нові сторінки та публікувати їх без необхідності робити це через код.

Управління користувачами: Багато CMS мають системи управління користувачами, що дозволяють створювати різні ролі та рівні доступу для різних користувачів. Наприклад, ви можете мати адміністраторів, редакторів, авторів та інших користувачів з обмеженим доступом до адміністративних функцій.

Налаштування дизайну: Багато CMS мають готові шаблони та теми, які дозволяють швидко створити привабливий дизайн для вашого сайту. Ви можете легко змінювати кольори, шрифти, розміщення елементів та інше за допомогою вбудованих інструментів.

Розширення функціоналу: Багато CMS мають розширення та додатки, які дозволяють розширити функціонал вашого сайту. Наприклад, ви можете встановити плагіни для SEO, аналітики, соціальних мереж та багато іншого.

Отже, системи управління веб-контентом відіграють ключову роль у процесі створення та управління сайтами, надаючи потужні інструменти для організації, редагування та публікації веб-контенту. Вони дозволяють зосередитися на створенні якісного контенту та взаємодії з аудиторією, не витрачаючи багато часу на технічні аспекти розробки веб-сайту.

2.2 Основи створення сайту. Які програми для цього використовуються

Система керування веб-контентом (CMS) - це програмне забезпечення, яке дозволяє легко створювати, редагувати, оновлювати та управляти вмістом веб-сайту без необхідності глибоких знань програмування або HTML. Основні

функції CMS включають управління текстом, зображеннями, відео, аудіо, посиланнями та іншими елементами веб-сайту.

CMS робить процес створення та управління веб-сайтом більш простим та ефективним. Ось як вона пов'язана із створенням сайту:

Спрощення розробки: Завдяки CMS вам не потрібно писати код з нуля. Ви можете використовувати готові теми і шаблони, що значно спростить процес створення дизайну сайту.

Легке редагування вмісту: CMS надає інтерфейс для редагування вмісту сайту через веб-браузер. Це дозволяє вам легко змінювати тексти, додавати нові сторінки, завантажувати зображення тощо без необхідності втручання в код.

Управління користувачами: Багато CMS мають систему управління користувачами, яка дозволяє обмежувати доступ до певних функцій або вмісту в залежності від рівня доступу користувача.

Розширення функціоналу: CMS часто мають велику кількість розширень і плагінів, які дозволяють розширити функціональність сайту, додавши нові функції або інтегруючи сторонні сервіси.

Підтримка і безпека: Багато популярних CMS мають активну спільноту, яка надає підтримку та оновлення для забезпечення безпеки сайту та вирішення можливих проблем.

Отже, CMS виконує роль інструмента, що полегшує процес створення та управління веб-сайтом, забезпечуючи широкий спектр можливостей для редагування вмісту та налаштування функціоналу сайту. Розглянемо цю тему більш детально.

Типи CMS:

- Відкриті (Open Source) CMS: Це CMS, розповсюджені під ліцензією з відкритим вихідним кодом, такі як WordPress, Joomla, Drupal тощо. Вони безкоштовні для використання та мають велику спільноту розробників і користувачів.
- Пропріетарні (Proprietary) CMS: Це комерційні CMS, для використання яких потрібно придбати ліцензію. Приклади включають Adobe Experience Manager, Kentico, Sitecore тощо.
- Хмарні CMS: Це CMS, які працюють на основі хмарних послуг. Приклади: Wix, Squarespace, Shopify тощо.

Функції CMS:

- Редагування вмісту (тексти, зображення, відео).
- Створення нових сторінок та публікацій.
- Управління користувачами і ролями.
- Можливість розгортання додаткових функцій за допомогою плагінів або розширень.
- Підтримка SEO (оптимізація для пошукових систем).
- Аналітика та звіти про відвідуваність сайту.[4]
- Інтеграція з соціальними мережами і зовнішніми сервісами.

Вибір CMS:

- Вибір CMS залежить від потреб вашого проекту, рівня технічних знань, бюджету та інших факторів.
- WordPress часто використовується для блогів та невеликих бізнес-сайтів.

- Joomla та Drupal підходять для складніших веб-проектів з великим обсягом контенту.
- Shopify і Magento - для онлайн-магазинів.
- Wix і Squarespace - для швидкого створення простих сайтів без потреби в технічних знаннях.

Безпека CMS:

- Важливо регулярно оновлювати CMS та всі розширення для запобігання вразливостям.
- Використовуйте надійні паролі та аутентифікацію двофакторного входу.
- Встановлюйте тільки довірені плагіни та теми.

Також щоб більш детально дізнатись про саму суть цих програм ми можемо продивитись [приклад](#) їх застосування:

WordPress:

- Блоги: WordPress почав свою історію як платформа для створення та управління блогами, і він є популярним вибором для блогерів будь-якого рівня.
- Сайти портфоліо: Багато художників, фотографів та дизайнерів використовують WordPress для створення своїх онлайн-портфоліо.

- Малий бізнес: Багато малих компаній використовують WordPress для створення веб-сайтів для своїх бізнесів, таких як ресторани, магазини тощо.

Joomla:

- Корпоративні веб-сайти: Joomla часто використовується для створення веб-сайтів компаній та організацій, оскільки він має ряд функцій для управління багатою інформацією.
- Онлайн-магазини: Joomla може бути використаний для створення електронних комерційних сайтів за допомогою розширень, таких як VirtueMart.
- Соціальні мережі: Деякі проекти соціальних мереж використовують Joomla для створення та управління спільнотами та вмістом.

Drupal:

- Урядові веб-сайти: Drupal часто використовується для створення веб-сайтів урядових установ та організацій, оскільки він має потужні інструменти безпеки та ролевого управління.
- Великі веб-сайти новин: Drupal може бути використаний для створення веб-сайтів новин або медіа, оскільки він має засоби для організації та управління великим обсягом контенту.
- Освітні платформи: Drupal широко використовується в освітній сфері для створення навчальних платформ і порталів.

Wix:

- Онлайн-магазини: Wix може бути використаний для швидкого створення магазинів за допомогою готових шаблонів та інструментів електронної комерції.

- Особисті веб-сайти: Wix дозволяє просто створювати веб-сайти для особистого використання, такі як портфоліо, блоги тощо, без необхідності великих технічних знань.
- Події та інформаційні сайти: Багато людей використовують Wix для створення веб-сайтів для подій, таких як весілля, виставки, конференції тощо.
- Кожен з цих прикладів демонструє різноманітність та потенціал різних CMS в різних сферах інтернет-діяльності.[5]

Переглянемо декілька прикладів використання цих інструментів в самому коді сайту:

WordPress

Приклад шаблону сторінки в WordPress:

php

```
<?php get_header(); ?>
```

```
<div id="content">
```

```
<?php if (have_posts()) : while (have_posts()) : the_post(); ?>
```

```
<div class="post">
```

```
<h2><?php the_title(); ?></h2>
```

```
<div class="entry">
```

```
<?php the_content(); ?>
```

```
</div>
```

```
</div>
```

```
<?php endwhile; endif; ?>
</div>
```

```
<?php get_sidebar(); ?>
```

```
<?php get_footer(); ?>
```

Цей PHP-код використовується в шаблоні Joomla для відображення модулів на певній позиції сторінки.

Joomla

Приклад виклику модуля в Joomla:

php

```
<?php if ($this->countModules('position-1')) : ?>
    <div id="position-1" class="moduletable">
        <jdoc:include type="modules" name="position-1" style="xhtml" />
    </div>
<?php endif; ?>
```

Отже, цей код перевіряє, чи є модулі в позиції position-1, і якщо такі є, створює контейнер для них і відображає ці модулі на сторінці.

Drupal

Приклад створення власного модуля в Drupal:

php

```
function mymodule_menu() {
    $items = array();
```

```

$items['mymodule/page'] = array(
    'title' => 'My Module Page',
    'page callback' => 'mymodule_page_function',
    'access callback' => TRUE,
    'type' => MENU_CALLBACK,
);
return $items;
}


```

```

function mymodule_page_function() {
    return 'Hello, world!';
}


```

Цей PHP-код створює нову сторінку в Drupal за допомогою кастомного модуля.

Wix

Приклад використання JavaScript у Wix для додавання функціональності:

```

javascript
import wixData from 'wix-data';

$w.onReady(function () {
    // Ваш код JavaScript тут


```

```
console.log("Page is ready");  
  
});
```

Це лише декілька прикладів того, як можна використовувати код з різними CMS. Кожен з цих прикладів показує специфічний синтаксис та API для кожної платформи, що використовується для створення веб-сайтів.

Оскільки ми переглянули усі приклади створення сайтів за допомогою CMS, та приклади кодів використання, можемо також переглянути сайти які були створені з використанням різних CMS:

WordPress

The New Yorker (<https://www.newyorker.com/>) - Популярний сайт новин та культурних статей, побудований на WordPress.

TechCrunch (<https://techcrunch.com/>) - Сайт, що займається технологіями та стартапами, також використовує WordPress.

Joomla

Harvard University (<https://www.harvard.edu/>) - Офіційний сайт Гарвардського університету реалізований на Joomla.

Linux.com (<https://www.linux.com/>) - Сайт, присвячений Linux технологіям та спільноті, також побудований на Joomla.

Drupal

WhiteHouse.gov (<https://www.whitehouse.gov/>) - Офіційний сайт Білого дому, який використовує Drupal для управління вмістом та функціональністю.

NASA (<https://www.nasa.gov/>) - Веб-портал Національного аеронавтичного та космічного агентства, побудований на Drupal.

Wix

Blue Moon Brewing Company

(<https://www.bluemoonbrewingcompany.com/av?url=https://www.bluemoonbrewingcompany.com/>) - Веб-сайт пивоварні, побудований з використанням Wix.

Кожен з цих веб-сайтів показує різноманітність та можливості різних CMS, від блогів та новинних сайтів до корпоративних веб-порталів та онлайн-магазинів.[6]

2.3 Перелік інструментів та програм які застосовуються для оформлення сайту

Переглянемо які мають бути кроки для створення свого сайту

1. Визначте свої цілі та потреби

Під час визначення цілей вам слід зрозуміти, що ви хочете досягти зі своїм веб-сайтом: це може бути представлення вашого бізнесу, надання інформації про ваші послуги або товари, продаж товарів онлайн, залучення клієнтів тощо.

Важливо також визначити вашу цільову аудиторію та їхні потреби: хто буде відвідувати ваш сайт і яка інформація є для них важливою?

Важливо чітко визначити цілі та потреби вашого проекту, щоб мати чітке розуміння того, що ви хочете досягти. Однак може виникнути складність у визначенні цілей, якщо у вас недостатньо досвіду або якщо цілі вашого проекту не є чіткими.

2. Виберіть платформу для створення сайту

Різні платформи мають свої особливості та переваги. Наприклад, WordPress є дуже популярним для блогів та малих бізнесів, Joomla використовується для створення складніших веб-сайтів, а Shopify - для електронної комерції. Оберіть ту, яка найбільше відповідає вашим потребам та рівню технічних знань.

З великою кількістю доступних платформ може бути важко обрати ту, яка найкраще відповідає вашим потребам. Це особливо важко для початківців, які можуть не мати достатнього розуміння різниці між різними платформами.

3. Оберіть домен та хостинг

Доменна назва повинна бути легко запам'ятовуваною та відображати суть вашого бізнесу або проекту. Виберіть надійного хостинг-провайдера зі зручними умовами та хорошою підтримкою клієнтів.

При виборі домену і хостингу можуть виникнути труднощі у знаходженні відповідних варіантів за доступною ціною, а також у розумінні технічних аспектів таких як DNS, розширення домену, пропагація DNS тощо.

4. Створіть дизайн сайту

Дизайн вашого сайту повинен бути привабливим, легким для користувача та відповідати його цілям та аудиторії. Важливо врахувати мобільну адаптивність та доступність для людей з обмеженими можливостями.

Хоча існують багато шаблонів та інструментів для створення дизайну, може виявитися складним створити дизайн, який відповідає вашим потребам та відображає вашу унікальну ідентичність.

5. Розробіть вміст сайту

Напишіть чіткий та зрозумілий текст для вашого сайту, використовуючи ключові слова для SEO. Додайте якісні зображення та відео, які доповнюватимуть ваш вміст та робитимуть його більш цікавим для користувачів.

Написання якісного вмісту може бути складним, особливо якщо ви не маєте досвіду в копірайтингу або SEO-оптимізації. Також може бути складно знайти або створити графічний контент для вашого сайту.

6. Розробіть та опублікуйте сайт

Використовуйте вибрану вами платформу для створення сайту та налаштуйте його згідно зі своїми потребами. Перед публікацією перевірте, чи працює сайт належним чином на різних пристроях та веб-браузерах.

Під час розробки сайту можуть виникати труднощі з кодом, функціональністю та сумісністю з різними пристроями та браузерами. Також може бути важко знайти і виправити помилки під час тестування сайту перед публікацією.

7. Продвигайте ваш сайт

Після публікації сайту розробіть стратегію його продвиження, включаючи SEO-оптимізацію, рекламні кампанії, активність в соціальних медіа тощо. Залучайте аудиторію та розширюйте свою онлайн-присутність, щоб забезпечити успіх вашого сайту.

Основна складність полягає в тому, щоб привернути увагу до вашого сайту в конкурентному інтернет-середовищі. Для цього потрібно мати стратегію SEO, рекламні зусилля та активність в соціальних медіа, а це може вимагати значних зусиль та часу.[7]

Програми які зазвичай використовуються для створення сайту:

Текстові редактори

Visual Studio Code - Це безкоштовний та відкритий текстовий редактор, розроблений компанією Microsoft. Він має багато корисних функцій, таких як автоматичне завершення коду, інтеграція з системами керування версіями,

вбудована підтримка для JavaScript, TypeScript, HTML, CSS та інших мов програмування, а також велику кількість розширень, які дозволяють розширити його функціональність. Використовується для редагування та написання коду на різних мовах програмування, таких як HTML, CSS, JavaScript, Python, PHP тощо. За допомогою нього можна створювати веб-сайти, веб-додатки, мобільні додатки, ігри та інші програмні продукти.

Sublime Text - Це швидкий та легкий текстовий редактор, який має багато корисних функцій, таких як підтримка широкого спектру мов програмування, швидке перемикання між файлами, можливість використання макросів та плагінів для автоматизації рутинних завдань, а також можливість налаштування вигляду та поведінки редактора. Використовується для редагування коду на різних мовах програмування. За допомогою нього можна створювати веб-сайти, веб-додатки, мобільні додатки, настільні додатки та інші програмні рішення.

Інтегровані середовища розробки (IDE)

Visual Studio - Це комплексне інтегроване середовище розробки (IDE) для роботи з різними мовами програмування, такими як C#, Visual Basic, F#, C++, Python, JavaScript, TypeScript та іншими. Він має широкий функціонал, такий як інструменти для розробки програмного забезпечення для Windows, веб-додатків, мобільних додатків та ігор. Використовується для розробки програмного забезпечення на різних мовах програмування, таких як C#, Visual Basic, F#, C++, JavaScript тощо. За допомогою нього можна створювати веб-сайти, веб-додатки, мобільні додатки, настільні додатки та інші програмні продукти.

IntelliJ IDEA - Це інтегроване середовище розробки (IDE) для роботи з Java, Kotlin, Groovy, Scala, JavaScript, TypeScript, SQL та іншими мовами програмування. Воно відоме своєю потужною системою автодоповнення, вбудованими інструментами для аналізу коду, відлагодження та тестування, а

також великою кількістю плагінів для розширення його функціональності. Використовується для розробки програмного забезпечення на мовах програмування Java, Kotlin, Groovy, Scala, JavaScript, TypeScript та інших. За допомогою нього можна створювати веб-сайти, веб-додатки, мобільні додатки, серверні додатки та інші програмні продукти.

Графічні редактори

Adobe Photoshop - Це програмне забезпечення для роботи з растровою графікою, що використовується для редагування та створення зображень, ілюстрацій, логотипів, веб-дизайну, макетів та інших графічних елементів. Воно має широкий набір інструментів та функцій для роботи з графічними об'єктами, фільтрами, шарами, текстом та іншими елементами. Використовується для створення та редагування растрової графіки для веб-сайтів, веб-додатків, ігор, мультимедійних презентацій тощо.

Adobe Illustrator - Це векторний графічний редактор, який використовується для створення ілюстрацій, логотипів, макетів, іконок та інших векторних графічних елементів. Він має потужні інструменти для створення та редагування векторних об'єктів, маніпулювання кривими та формами, застосування ефектів та стилів, а також можливості для роботи з текстом та іншими елементами. Використовується для створення та редагування векторної графіки для веб-сайтів, логотипів, ілюстрацій, іконок, макетів тощо.

Це лише огляд кожної програми та інструменту, і кожен з них має багато інших функцій та можливостей, які можуть бути корисними для різних типів проектів.

РОЗДІЛ 3. РОЗРОБКА САЙТУ З ЗАСТОСУВАННЯМ ДЕЯКИХ ПРАВИЛ ТА ІНСТРУМЕНТІВ

3.1 Як використовують система керувань веб-контентом в створені сайту

Для оформлення веб-сайту можна використовувати різні інструменти та програми залежно від потреб і навичок користувача.

Ось деякі з них:

- HTML та CSS: Це базові мови для розмітки та стилізації веб-сторінок. Вони використовуються для визначення структури та вигляду вашого сайту.

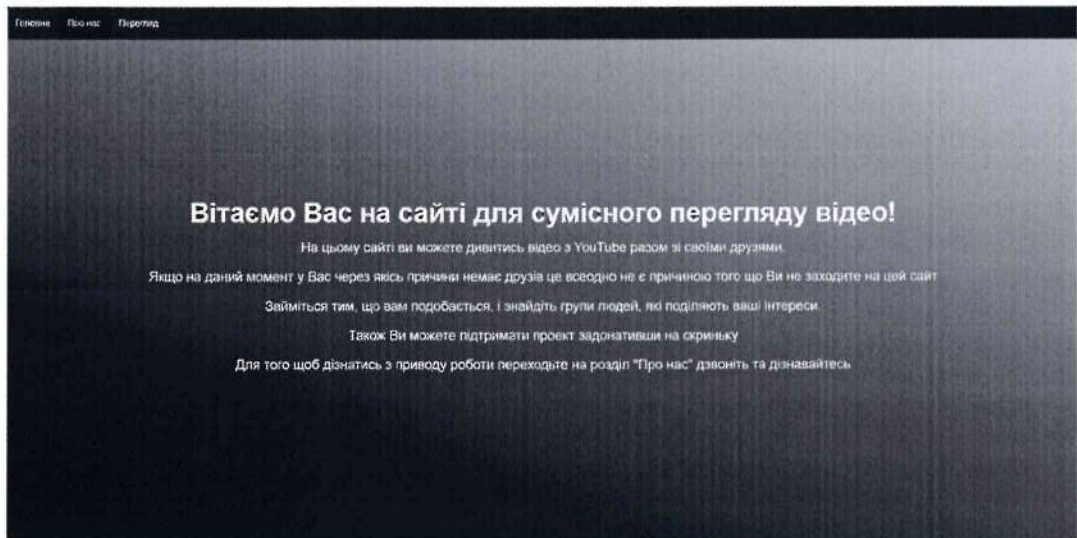


Рисунок 3.1 - стилізація сайту за допомогою CSS та HTML

- JavaScript: Ця мова програмування використовується для створення динамічного веб-змісту, анімацій та взаємодії користувача з веб-сторінкою.
- Фреймворки: Наприклад, Bootstrap, Foundation, або Materialize CSS дозволяють швидко розробляти стилізовані та адаптивні веб-сайти за допомогою готових компонентів та шаблонів.

- CMS (Системи управління контентом): Наприклад, WordPress, Joomla, або Drupal. Вони дозволяють створювати та керувати веб-сайтами без необхідності знання програмування.

Графічні редактори:

- Такі як Adobe Photoshop, Sketch або GIMP використовуються для створення графічних елементів, логотипів, фонів тощо.

Редактори коду:

- Наприклад, Visual Studio Code, Sublime Text або Atom допомагають вам писати код HTML, CSS та JavaScript зручніше та ефективніше.
- Шаблони: Використання готових веб-шаблонів може значно полегшити процес створення веб-сайту. Вони дозволяють швидко створити професійний вигляд без великих зусиль.
- Розробницькі інструменти браузера: Як Chrome DevTools або Firefox Developer Edition дозволяють вам відлагоджувати та вдосконалювати ваш сайт прямо в браузері.

Приклади веб-сайтів, які були створені за допомогою різних програм та інструментів:

- Facebook: Використовується комбінація PHP, HTML, CSS та JavaScript для створення динамічної соціальної мережі.
- Twitter: Аналогічно Facebook, Twitter використовує PHP, HTML, CSS та JavaScript для створення своєї соціальної платформи.
- Instagram: Цей веб-сайт також використовує PHP, HTML, CSS та JavaScript разом із фреймворком Django для побудови своєї соціальної мережі.

- WordPress.com: WordPress є однією з найпопулярніших CMS і використовується для створення блогів, новинних сайтів та навіть електронних магазинів.
- The New York Times: Цей сайт використовує фреймворк Django для публікації новин та інформаційних статей.[8]
- GitHub Pages: GitHub Pages надає можливість хостити веб-сайти безкоштовно з використанням HTML, CSS, JavaScript та Jekyll.
- Bootstrap Expo: Це сайт, який демонструє різноманітні веб-сайти, створені за допомогою Bootstrap, що дозволяє швидко розробляти адаптивні та стильні сторінки.
- CodePen: CodePen - це веб-сервіс, який дозволяє вам створювати та демонструвати веб-сайти, використовуючи HTML, CSS та JavaScript прямо в браузері.

Також розглянемо приклади кодів які можуть використовуватися для оформлення сайту:

- HTML

```

49 <div id="content">
50 <!-- Главная страница -->
51 <div id="homePage">
52 <h1>Вітаємо Вас на сайті для сумісного перегляду відео!</h1>
53 <p>На цьому сайті ви можете дивитись відео з YouTube разом зі своїми
54 друзями.</p>
55 <p>Якщо на даний момент у Вас через якісь причини немає друзів це всеодно не є причиною того що Ви не заходите на цей сайт</p>
56 <p>Займіться тим, що вам подобається, і знайдіть групи людей, які поділяють ваші інтереси. </p>
57 <p>Також Ви можете підтримати проєкт задонативши на скриньку</p>
58 <p>Для того щоб дізнатись з приводу роботи переходьте на розділ "Про нас" дзвоніть та дізнавайтесь</p>
59
60
61
62 </div>
63
64 <!-- Про нас -->
65 <div id="aboutPage" class="hidden">
66 <h1>Про нас</h1>
67 <p>Сайт знаходиться в бета-тесті.</p>
68 <p>Контакти: nema@nic.gmail.com</p>
69 </div>
70
71 <!-- Просмотр видео -->
72 <div id="watchPage" class="hidden">
73 <h1>Перегляд відео</h1>
74 <div id="player"></div>
75 <input type="text" id="videoUrl" placeholder="Вставьте URL видео">
76 <button id="loadVideo">Завантажити відео</button>
77 </div>
78 </div>
79
80 <script src="https://www.youtube.com/iframe_api"></script>
81 <script src="script.js"></script>
82 </body>
83
84 </head>
85 <body>
86 <script src="https://www.youtube.com/iframe_api"></script>
87 <script src="script.js"></script>
88 </body>
89 </html>

```

Рисунок 3.2 - HTML code який використовувався для оформлення візуалу сайту

- CSS

```

49
50 #player {
51     width: 200%;
52     height: 450px;
53     background-color: #000;
54     margin: 20px auto;
55     margin-left: -250px;
56 }
57
58 input, button {
59     margin-top: 20px;
60     padding: 10px;
61     font-size: 16px;
62 }
63 #mainImage {
64     max-width: 100%;
65     height: auto;
66     margin-top: 20px;
67 }
68
69 #player {
70     width: 100%;
71     max-width: 800px;
72     height: 450px;
73     background-color: #000;
74     margin: 20px auto;
75 }
76
77 input, button {
78     margin-top: 20px;
79     padding: 10px;
80     font-size: 16px;
81     width: 100%;
82     max-width: 300px;
83     box-sizing: border-box; /* Учитывает padding и border в общей ширине элемента */
84 }
85
86 @media (max-width: 768px) {
87     #player {
88         height: 300px;
89     }
90 }
91
92 @media (max-width: 480px) {
93     nav ul {
94         flex-direction: column;
95         align-items: flex-start;
96     }
97
98     nav ul li a {
99         padding: 10px 14px;
100    }
101
102    #player {
103        height: 200px;
104    }
105

```

Рисунок 3.3 - CSS code code який використовувався для оформлення візуалу сайту

- JavaScript

```

1  const WebSocket = require('ws');
2
3  const wss = new WebSocket.Server({ port: 8080 });
4
5  wss.on('connection', (ws) => {
6    ws.on('message', (message) => {
7      wss.clients.forEach(client => {
8        if (client !== ws && client.readyState === WebSocket.OPEN) {
9          client.send(message);
10         }
11       });
12     });
13   });
14
15   console.log('WebSocket server is running on ws://localhost:8080');

```

Рисунок 3.4 JavaScript code який використовувався для створення динамічної сторінки

3.2 Порівняння створення сайтів з системою керувань веб-контентом та без неї

Гнучкість та простота використання: CMS надають простий інтерфейс для створення та редагування вмісту без необхідності знання HTML, CSS або інших мов програмування. Це робить їх доступними для широкого кола користувачів, від початківців до досвідчених розробників.

Масштабованість: CMS можуть бути використані для будь-якого типу веб-сайту, від простого блогу до складних корпоративних порталів. Вони зазвичай мають гнучкість та можливості розширення, що дозволяють адаптувати їх до різних потреб.

Модульність та плагіни: Багато CMS пропонують широкий вибір модулів, плагінів та розширень, які дозволяють розширити функціональність вашого

сайту. Це може включати різноманітність функцій, від аналітики та SEO до соціальних мереж та електронної комерції.[9]

Спільнота та підтримка: Багато популярних CMS мають велику спільноту користувачів та розробників, які надають підтримку, допомогу та розвиток. Це означає, що ви можете легко знайти відповіді на свої запитання, допомогу з проблемами та навіть внести свій внесок у вдосконалення платформи.

Безпека: Важливою перевагою використання CMS є те, що багато з них мають вбудовані заходи безпеки, які допомагають захистити ваш сайт від зловмисників. Більшість CMS регулярно оновлюються для виправлення виявлених уразливостей.

SEO-оптимізація: Багато CMS мають вбудовані інструменти для оптимізації пошукових систем (SEO), що дозволяє вам оптимізувати ваш вміст для кращого рейтингу в пошукових системах і залучення більше відвідувачів.

Загалом, CMS є потужними інструментами, які дозволяють навіть неспеціалістам легко створювати та управляти веб-сайтами, забезпечуючи широкий спектр можливостей та функцій.

Якщо розглядати цей пункт детальніше то ми маємо таку картину

Гнучкість та простота використання:

- **Простий інтерфейс:** CMS зазвичай мають інтуїтивно зрозумілий інтерфейс, який дозволяє користувачам легко додавати, редагувати та видаляти вміст на сайті без складнощів.
- **Безпека відносно користувача:** Деякі CMS надають можливість обмежувати доступ до різних частин сайту для різних користувачів, забезпечуючи безпеку відносно ролей.

Масштабованість:

- **Адаптивність до різних потреб:** CMS можуть бути використані для будь-якого типу веб-сайту, від особистих блогів до великих корпоративних порталів.
- **Можливість розширення:** Вони зазвичай мають можливості для розширення функціоналу через додаткові модулі, плагіни та теми.

Модульність та плагіни:

- **Розширення функціоналу:** Плагіни та модулі дозволяють додавати нові функції та можливості до сайту без необхідності писати власний код.
- **Гнучкість налаштувань:** Багато плагінів мають різні параметри, які можна налаштувати відповідно до вашого завдання.

Спільнота та підтримка:

- **Форуми та ресурси спільноти:** Багато CMS мають активні форуми та ресурси, де користувачі можуть обговорювати питання та ділитися досвідом.
- **Оновлення та підтримка:** Популярні CMS регулярно оновлюються, щоб виправляти помилки безпеки та додавати нові функції.

Безпека:

- **Вбудовані заходи безпеки:** Багато CMS мають вбудовані механізми безпеки, які допомагають захистити ваш сайт від зловмисників.
- **Оновлення безпеки:** Важливо регулярно оновлювати свою CMS, щоб мати останні версії з виправленими уразливостями.

SEO-оптимізація:

- **Дружні URL-адреси:** Багато CMS автоматично створюють дружні для SEO URL-адреси, що полегшує індексацію сайту пошуковими системами.
- **Метатеги та ключові слова:** Багато CMS мають інструменти для додавання метатегів та ключових слів, що допомагають покращити SEO-показники вашого сайту.

Ці переваги роблять CMS популярними серед веб-розробників та користувачів, які шукають простий та ефективний спосіб створення та управління веб-сайтами.

Системи управління веб-контентом (CMS) широко використовуються на різних типах веб-сайтів, від особистих блогів до великих корпоративних порталів. Ось кілька прикладів веб-сайтів, які використовують різні CMS:

WordPress:

- Блоги та особисті сайти.
- Новинні ресурси.
- Магазини електронної комерції.
- Корпоративні веб-сайти.

Приклади: The New York Times, BBC America, TechCrunch, Sony Music.

Joomla:

- Корпоративні веб-сайти.
- Онлайн-журнали та видання.
- Освітні портали.

Приклади: Harvard University, Linux.com, IKEA.

Drupal:

- Корпоративні портали.
- Урядові веб-сайти.
- Організаційні ресурси.

Приклади: The Economist, Tesla Motors, NASA.[10]

Magento:

- Онлайн-магазини.
- Електронна комерція.
- Приклади: Coca-Cola, Nike, Ford.

Shopify:

- Онлайн-магазини.
- Малі бізнеси.

Приклади: Kylie Cosmetics, Gymshark, Allbirds.

Wix:

- Особисті сайти.
- Малі бізнеси.

Приклади: Airbnb, Spotify, SquareSpace.

Ghost:

- Блоги та видавництва.
- Особисті сайти.

Приклади: DigitalOcean, Tinder Engineering, Mozilla Hacks.

3.3 Порівняння створення сайтів з системою керувань веб-контентом та без неї

Порівняння створення сайтів з використанням системи управління веб-контентом (CMS) та без неї може допомогти краще зрозуміти переваги кожного підходу. Ось деякі ключові аспекти порівняння:

Складність використання:

- **CMS:** Використання CMS зазвичай спрощує процес створення веб-сайту для користувачів без глибоких технічних знань. Вони надають інтуїтивно зрозумілі інтерфейси для додавання та редагування вмісту.
- **Без CMS:** Створення сайту без використання CMS може вимагати глибоких знань HTML, CSS, JavaScript та інших мов програмування. Це може бути складним для початківців.

Час створення:

- **CMS:** Завдяки готовим шаблонам та інструментам CMS, створення веб-сайту може бути швидше порівняно з ручною розробкою.
- **Без CMS:** Побудова веб-сайту з нуля може зайняти більше часу через необхідність написання всього коду вручну.

Гнучкість та розширюваність:

- **CMS:** CMS зазвичай мають широкий вибір модулів, плагінів та тем, що дозволяє розширити функціональність сайту без необхідності писати власний код.
- **Без CMS:** При ручній розробці ви маєте повний контроль над кодом і можливість налаштувати його відповідно до своїх потреб.

Безпека:

- **CMS:** Багато CMS мають вбудовані заходи безпеки, такі як оновлення безпеки та механізми аутентифікації.
- **Без CMS:** При ручній розробці безпеку слід ретельно враховувати під час написання коду, оскільки помилки можуть призвести до уразливостей.

Вартість:

- **CMS:** Використання безкоштовних CMS, таких як WordPress або Joomla, може значно зменшити вартість розробки сайту. Проте, додаткові платні модулі та плагіни можуть потребувати додаткових витрат.
- **Без CMS:** Ручна розробка може бути дорожчою через потребу наймати досвідчених розробників для написання коду та розробки функціоналу з нуля.[11]

Обираючи між використанням CMS та без неї, важливо враховувати власні потреби, рівень технічних знань та бюджет проекту.

Також розглянемо плюси та мінуси використання інструментів.

Плюси використання CMS:

- **Простота використання:** CMS надають інтуїтивно зрозумілі інтерфейси, що дозволяють навіть неспеціалістам легко керувати веб-сайтом.
- **Швидкість розробки:** Завдяки готовим шаблонам, плагінам та інструментам, створення веб-сайтів з використанням CMS може бути значно швидше, ніж ручна розробка.

- **Масштабованість:** CMS зазвичай мають гнучку архітектуру, яка дозволяє легко розширювати функціональність сайту за допомогою плагінів та модулів.
- **Активна спільнота:** Багато CMS мають велику та активну спільноту користувачів і розробників, яка надає підтримку, розвиток та вирішення проблем.
- **Безпека:** Багато CMS надають вбудовані засоби безпеки та регулярно оновлюються, щоб виправляти виявлені уразливості.

Мінуси використання CMS:

- **Обмежена гнучкість:** Деякі CMS можуть мати обмежену гнучкість у порівнянні з ручною розробкою, особливо для складних функціональних вимог.
- **Залежність від плагінів:** Використання багатьох плагінів для розширення функціональності може призвести до проблем з сумісністю та збільшенням часу завантаження сторінок.
- **Обмежене налаштування:** Налаштування деяких аспектів веб-сайту може бути обмеженим у порівнянні з ручною розробкою.
- **Вартість додаткових функцій:** Додаткові плагіни та теми можуть бути платними, що збільшує вартість розробки веб-сайту.

Плюси ручної розробки без використання CMS:

- **Повний контроль:** Ручна розробка дає повний контроль над кожним аспектом веб-сайту і може бути оптимізована для конкретних потреб.
- **Більша гнучкість:** Ручна розробка дозволяє створити унікальний дизайн та функціонал, який може бути складніше досягнути з використанням CMS.

Мінуси ручної розробки без використання CMS:

- **Складніше використання:** Ручна розробка може бути складнішою для навчання та вимагати глибоких знань веб-розробки.
- **Часові затрати:** Ручна розробка може зайняти більше часу, оскільки кожен елемент потрібно розробляти з нуля.

Обидва підходи мають свої переваги та недоліки, і обрання відповідного методу залежить від конкретних потреб, ресурсів та вмінь розробника.[12]

Висновок

Отже, на підставі порівняння плюсів та мінусів використання систем управління веб-контентом (CMS) та ручної розробки веб-сайтів без них можна зробити наступні висновки:

CMS:

- Відмінно підходять для швидкої розробки веб-сайтів без глибоких технічних знань.
- Надають простий інтерфейс для додавання та редагування вмісту.
- Мають вбудовані засоби безпеки та широкий вибір розширень для розширення функціональності.

Ручна розробка без CMS:

- Надає повний контроль над кожним аспектом веб-сайту і більшу гнучкість у реалізації унікального дизайну та функціоналу.
- Вимагає глибоких знань веб-розробки та більше часу для розробки.

Загалом, вибір між використанням CMS та ручною розробкою без них залежить від конкретних потреб, ресурсів та вмінь розробника. CMS відмінно підходять для швидкої розробки сайтів з мінімальними зусиллями, тоді як ручна розробка надає повний контроль та гнучкість, але вимагає більше часу та експертних знань.

Література

1. Site | Definition of Site by Oxford Dictionary on Lexico.com also meaning of Site. Lexico Dictionaries | English (англ.). Архів оригіналу за 11 листопада 2020. Прочитовано 22 серпня 2020.
2. Rachel Bistricher Jan 25 Get started by: Creating a website → | Getting a domain
3. digitl hub “I will design a professional wordpress website or web design”
4. Ilya S. WHAT TYPES OF SITES ARE THERE?
5. Oleg F. “WE CREATE PRESENTABLE WEBSITES WITH HIGH EFFICIENCY”
6. Managing Enterprise Content: A Unified Content Strategy. Ann Rockley, Pamela Kostur, Steve Manning. New Riders, 2003.
7. Informatics: understudy. for 8th grade closing total among. of education / [O. O. Bondarenko, V. V. Lastovetskyi, O. P. Pylypchuk, E. A. Shestopalov]. — Kharkiv: Ranok Publishing House, 2021 “Means of automated creation and publication of web resources”
8. Olga V. blog “Web design tools”
9. _Andrii K. “What is a content management system (CMS)?” 09.02.2024
10. “MOST POPULAR CMS FOR WEBSITE DEVELOPMENT: COMPARISON AND REVIEW” 06.09.2023 cms web development
11. Oleksiy S. “WordPress vs. Joomla: Which CMS to Choose?”
12. Volodymyr K. “Advantages and disadvantages of free CMS”

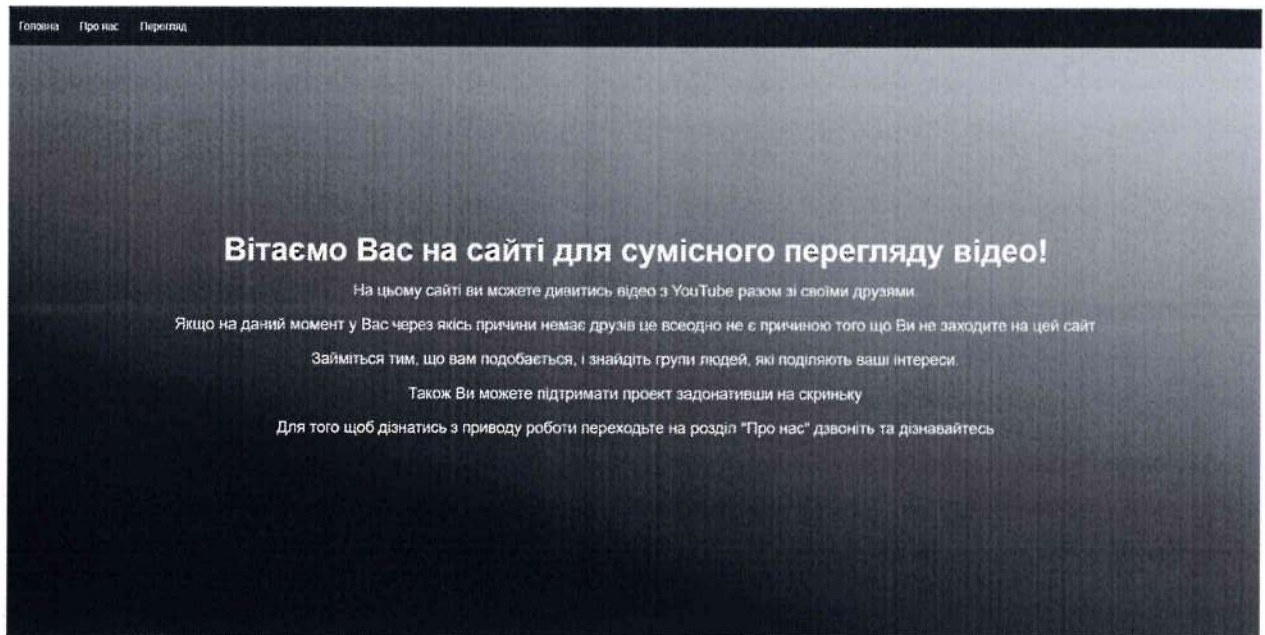


Рисунок 3.4 – вкладка «Головна»



Рисунок 3.5 - Вкладка «Про нас»

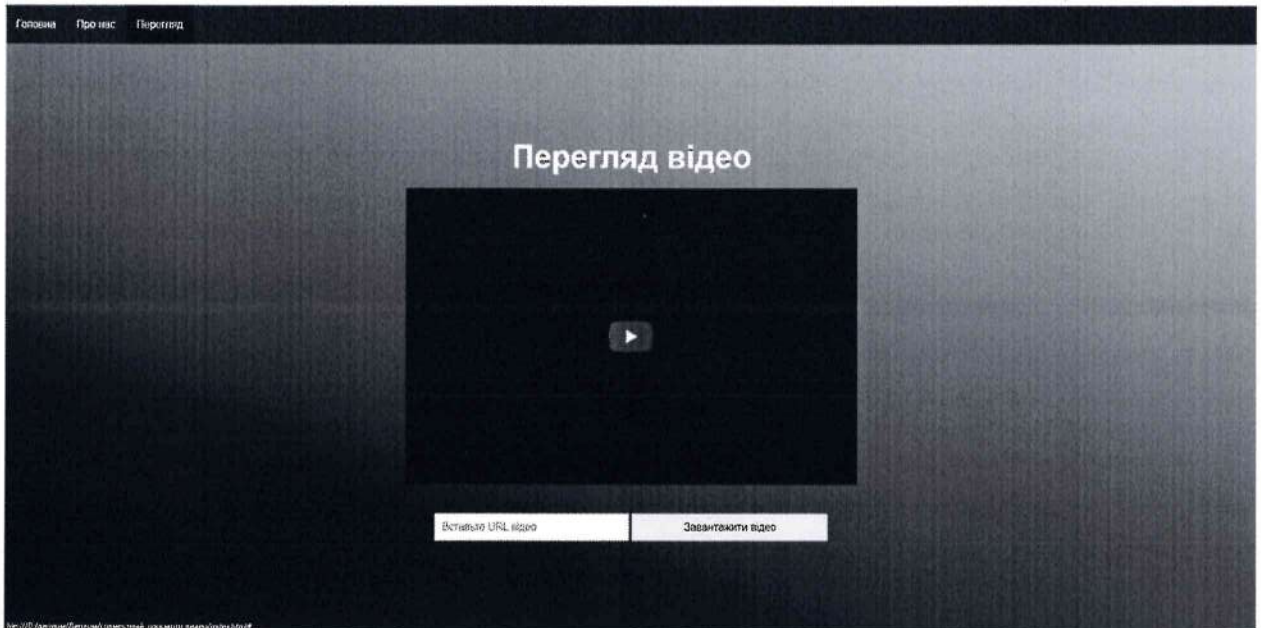


Рисунок 3.6 - вкладка перегляд

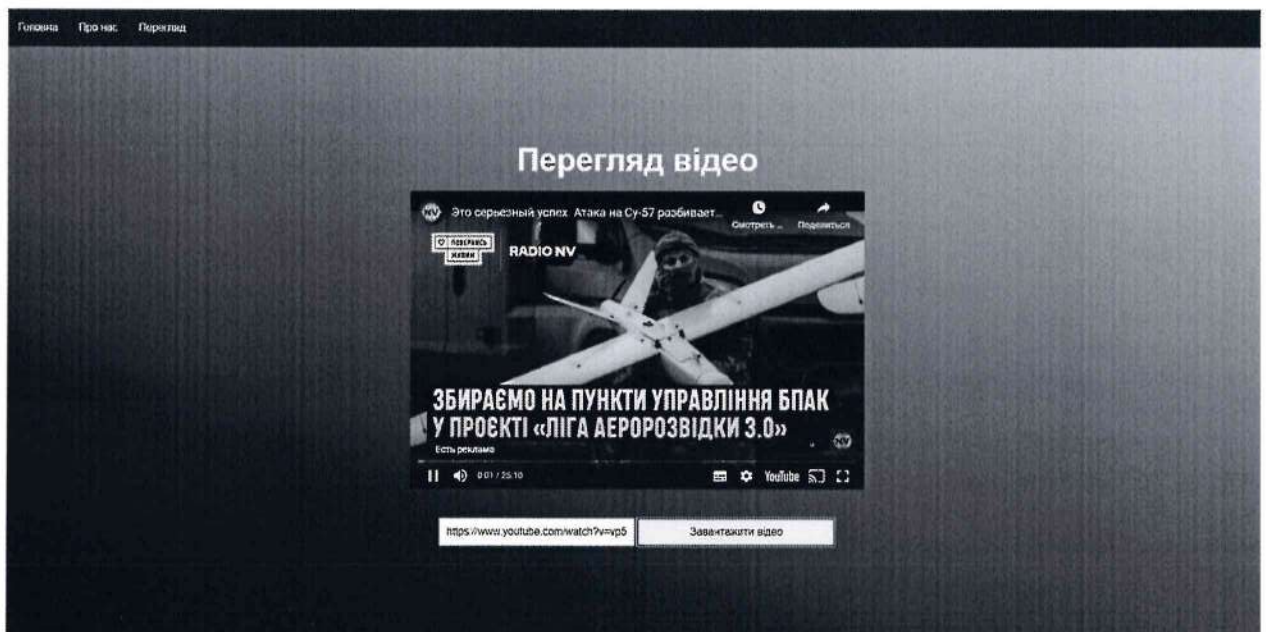


Рисунок 3.7 - сам перегляд

JavaScript код для сайту

```
const WebSocket = require('ws');

const wss = new WebSocket.Server({ port: 8080 });

wss.on('connection', (ws) => {
  ws.on('message', (message) => {
    wss.clients.forEach(client => {
      if (client !== ws && client.readyState === WebSocket.OPEN) {
        client.send(message);
      }
    });
  });
});

console.log('WebSocket server is running on ws://localhost:8080');
```

```
1 body {
2     font-family: Arial, sans-serif;
3     background-color: #708090;
4     margin: 0;
5     padding: 0;
6 }
7
8 nav {
9     position: fixed;
10    top: 0;
11    left: 0;
12    width: 100%;
13    background-color: #333;
14 }
15
16 nav ul {
17    display: flex;
18    list-style-type: none;
19    margin: 0;
20    padding: 0;
21    justify-content: flex-start;
22 }
23
24 nav ul li {
25    display: inline;
26 }
27
28 nav ul li a {
29    display: block;
30    color: white;
31    text-align: center;
32    padding: 20px 16px;
33    text-decoration: none;
34 }
35
36 nav ul li a:hover {
37    background-color: #111;
38 }
39
40 .hidden {
41    display: none;
42 }
43
44 #content {
45    padding: 20px;
46    margin-top: 60px; /* Отступ сверху, чтобы контент не перекрывался навигацией */
47 }
48
49 #player {
50    width: 200%;
51    height: 450px;
52    background-color: #000;
53    margin: 20px auto;
54    margin-left: -250px;
55 }
56
57 input, button {
58    margin-top: 20px;
59    padding: 10px;
60    font-size: 16px;
61 }
62
63 #mainImage {
64    max-width: 100%;
65    height: auto;
```

Рисунок 3.8 CSS код сайта