РЕФЕРАТ

Пояснювальна записка до курсової роботи

Метою роботи є створення пізнавального сайту для пізнавання загальної інформації та історії про фотоіндустрію

Об’єкт дослідження – історія фотографії та деяких відомих фотографів.

Практичним результатом даної роботі є створений сайт для тих хто має бажання дізнатися зось нове про фотоіндустрію.

ЗМІСТ

ВСТУП

Розділ 1 Методи розробки сайтів.

1.1 Ручна розробка за допомогою HTML.

1.2 Розробка сайту за допомогою програмних засобів.

Розділ 2 Основні функції систем управління Web-контентом.

2.1 Розробка контенту.

2.2 Управління сайтом.

2.3 Доставка контенту.

Розділ 3 Поняття і сутність системи управління контентом.

3.1 Основна ідея систем управління контентом.

Розділ 4 Класифікація систем управління контентом.

4.1 Проста система управління контентом.

4.2 Шаблонна система управління контентом.

4.3 Професійна система управління контентом.

4.4. Універсальна система управління контентом.

Розділ 5 Використання фреймворків.

5.1 Типи фреймворків.

Розділ 6 Створення сайтів на SaaS-платформах у Cloud.

6.1 Проекти на SaaS- платформах.

6.2 Кроки створення сайтів нa SaaS-платформах.

6.3 Мінуси SaaS-платформ для створення сайту.

6.4 Приклади SaaS-платформ.

ВИСНОВКИ

ВСТУП

**Фотографія – галузь науки, техніки і мистецтва, що використовує і вивчає методи отримання на світлочутливих матеріалах зображень (фотознімків) об’єктів або способи реєстрації оптичного та інших випромінювань.**

В основі фотографії лежить використання спеціальних матеріалів, в світлочутливому шарі яких в результаті дії випромінювання та подальшої хіміко-фотографічної обробки відбуваються фотохімічні реакції. Зазвичай фотографічні матеріали використовуються в поєднанні з тим або іншим оптичним пристроєм:фотографічним апаратом, фотографічним збільшувачем, копіювальним верстатом і т.д., що створює на світлочутливому шарі оптичне зображення.  
  
Фотографічний метод виник на початку XIX століття. Офіційною датою винаходу фотографії є 7 січня 1839, коли спосіб практичного отримання зображення на солях срібла, відкритий французом Луї Жаком Дагерром, був оприлюднений на засіданні Французької Академії наук.  
  
Термін «фотографія» був запропонований 14 березня 1839 англійським астрономом Д. Гершелем. Ця назва згодом стало загальновизнаним. У перекладі з грецької мови фотографія (photos – світло, grapho – пишу) означає «світлопис», хоча ця назва не було повним її визначенням. Знадобилося 160 років напруженоїроботи численних колективів дослідників, щоб фотографія стала досконалим технічним методом реєстрації інформації, якою вона є сьогодні.

[Альберт Магнус](https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D1%8C%D0%B1%D0%B5%D1%80%D1%82_%D0%92%D0%B5%D0%BB%D0%B8%D0%BA%D0%B8%D0%B9) виявив [нітрат срібла](https://uk.wikipedia.org/wiki/%D0%9D%D1%96%D1%82%D1%80%D0%B0%D1%82_%D1%81%D1%80%D1%96%D0%B1%D0%BB%D0%B0(I)) який пізніше буде використовуватися як світлочутливий матеріал в фотоемульсії на плівці.

[Георг Фабріціус](https://uk.wikipedia.org/w/index.php?title=%D0%93%D0%B5%D0%BE%D1%80%D0%B3_%D0%A4%D0%B0%D0%B1%D1%80%D1%96%D1%86%D1%96%D1%83%D1%81&action=edit&redlink=1)(1516–71) виявив [хлорид срібла](https://uk.wikipedia.org/wiki/%D0%A5%D0%BB%D0%BE%D1%80%D0%B8%D0%B4_%D1%81%D1%80%D1%96%D0%B1%D0%BB%D0%B0) що став основою виробництва [фотопаперу](https://uk.wikipedia.org/wiki/%D0%A4%D0%BE%D1%82%D0%BE%D0%BF%D0%B0%D0%BF%D1%96%D1%80).

1727 року лікар з Галлі [Йоган Шульц](https://uk.wikipedia.org/w/index.php?title=%D0%99%D0%BE%D0%B3%D0%B0%D0%BD_%D0%A8%D1%83%D0%BB%D1%8C%D1%86&action=edit&redlink=1) проводив досліди з розчином [азотнокислого срібла](https://uk.wikipedia.org/wiki/%D0%9D%D1%96%D1%82%D1%80%D0%B0%D1%82_%D1%81%D1%80%D1%96%D0%B1%D0%BB%D0%B0(I)) й [крейдою](https://uk.wikipedia.org/wiki/%D0%9A%D1%80%D0%B5%D0%B9%D0%B4%D0%B0). Коли посудину з сумішшю виставляли на сонячне світло, її поверхня одразу чорніла. При струшуванні розчин знову ставав білим. За допомогою шматочків паперу Шульц одержував на поверхні рідини силуети, за допомогою збовтування знищував їх й одержував нові візерунки. Але ці оригінальні досліди здавалися йому тільки забавою. Минуло ще сто років, перш ніж помічену ним властивість здогадалися використати при виготовленні фотографічних пластин.

Одного разу [Йоганн Християн Фабрицій](https://uk.wikipedia.org/wiki/%D0%99%D0%BE%D0%B3%D0%B0%D0%BD%D0%BD_%D0%A5%D1%80%D0%B8%D1%81%D1%82%D0%B8%D1%8F%D0%BD_%D0%A4%D0%B0%D0%B1%D1%80%D0%B8%D1%86%D1%96%D0%B9) змішав [кухонну сіль](https://uk.wikipedia.org/wiki/%D0%9A%D1%83%D1%85%D0%BE%D0%BD%D0%BD%D0%B0_%D1%81%D1%96%D0%BB%D1%8C) з розчином [азотнокислого](https://uk.wikipedia.org/wiki/%D0%90%D0%B7%D0%BE%D1%82%D0%BD%D0%B0_%D0%BA%D0%B8%D1%81%D0%BB%D0%BE%D1%82%D0%B0) [срібла](https://uk.wikipedia.org/wiki/%D0%A1%D1%80%D1%96%D0%B1%D0%BB%D0%BE) й одержав молочно-білий осад, що чорнів від сонячного світла. Фабриціус досліджував це явище, й у своїй книжці про метали, виданій 1556 року, повідомив, що за допомогою лінзи одержав зображення на поверхні осаду, відомого тепер під назвою [хлористого срібла](https://uk.wikipedia.org/wiki/%D0%A5%D0%BB%D0%BE%D1%80%D0%B8%D1%81%D1%82%D0%B5_%D1%81%D1%80%D1%96%D0%B1%D0%BB%D0%BE), і що зображення це ставало чорним або сірим залежно від тривалості освітлення його сонячними променями. Це був перший дослід в історії фотографії.

Наступна сторінка в історії фотографії пов'язана з ім'ям [Томаса Веджвуда](https://uk.wikipedia.org/w/index.php?title=%D0%A2%D0%BE%D0%BC%D0%B0%D1%81_%D0%92%D0%B5%D0%B4%D0%B6%D0%B2%D1%83%D0%B4&action=edit&redlink=1). Він клав на папір, зволожений розчином азотнокислого срібла, листя рослин. Після витримки на сонці вкрита листям частина паперу залишалася світлою, освітлена ж частина чорніла. Результатом цього досліду був білий силует на чорному тлі. Однак ці зображення можна було розглядати тільки у світлі свічки, тому що під сонячним промінням вони псувалися. Веджвуд спробував просочити розчином шкіру й виявив, що зображення на ній з'являються швидше. 1802 року Веджвуд опублікував результати своїх дослідів. Поступово він навчився одержувати контурні зображення на папері, шкірі й склі: протягом трьох хвилин — при експозиції їх на сонці, і протягом декількох годин — при витримці їх у тіні. Але ці знімки не витримували сонячного світла (вони не були «зафіксовані»). Лише 1819 року [Джон Гершель](https://uk.wikipedia.org/wiki/%D0%94%D0%B6%D0%BE%D0%BD_%D0%93%D0%B5%D1%80%D1%88%D0%B5%D0%BB%D1%8C) знайшов речовину, що закріплювала зображення. 1813 року за досліди з фотографічними пластинками взявся французький художник [Нісефор Ньєпс](https://uk.wikipedia.org/wiki/%D0%9D%D1%96%D1%81%D0%B5%D1%84%D0%BE%D1%80_%D0%9D%D1%8C%D1%94%D0%BF%D1%81) ([фр.](https://uk.wikipedia.org/wiki/%D0%A4%D1%80%D0%B0%D0%BD%D1%86%D1%83%D0%B7%D1%8C%D0%BA%D0%B0_%D0%BC%D0%BE%D0%B2%D0%B0) *Nicephore Niepce*), якому належить основна заслуга у винаході фотоапарата. Близько 1816 року він дійшов до думки одержувати зображення предметів за допомогою так званої [камери-обскури](https://uk.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BC%D0%B5%D1%80%D0%B0-%D0%BE%D0%B1%D1%81%D0%BA%D1%83%D1%80%D0%B0). Ця камера була відома ще з давнини. У найпростішому вигляді вона являє собою світлонепроникний ящик з невеликим отвором. Якщо стінка, протилежна отвору, буде з матового скла, то на ній виходить перевернуте зображення предметів, що перебувають перед камерою. Чим менший отвір, тим чіткіші контури зображення й тим воно слабше. На жаль, перше зображення, отримане Ньєпсом 1822 року, не збереглося. Тому першою в історії фотографією вважають «вид із вікна», отриманий ним [1826](https://uk.wikipedia.org/wiki/1826) року за допомогою [камери-обскури](https://uk.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BC%D0%B5%D1%80%D0%B0-%D0%BE%D0%B1%D1%81%D0%BA%D1%83%D1%80%D0%B0) на циновій пластинці, вкритій тонким шаром [асфальту](https://uk.wikipedia.org/wiki/%D0%90%D1%81%D1%84%D0%B0%D0%BB%D1%8C%D1%82). [Експозиція](https://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%BE%D0%B7%D0%B8%D1%86%D1%96%D1%8F_(%D1%84%D0%BE%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D1%96%D1%8F)) тривала вісім годин на яскравому сонячному світлі.

[1839](https://uk.wikipedia.org/wiki/1839) року француз [Луї Жак Дагерр](https://uk.wikipedia.org/wiki/%D0%9B%D1%83%D1%97_%D0%96%D0%B0%D0%BA_%D0%94%D0%B0%D0%B3%D0%B5%D1%80%D1%80) опублікував спосіб одержання зображення на [мідній](https://uk.wikipedia.org/wiki/%D0%9C%D1%96%D0%B4%D1%8C) пластині, вкритій [сріблом](https://uk.wikipedia.org/wiki/%D0%A1%D1%80%D1%96%D0%B1%D0%BB%D0%BE). Після 30-хвилинного [експонування](https://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%BE%D0%BD%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) Дагерр переніс пластину до темної кімнати й якийсь час тримав її над парами нагрітої [ртуті](https://uk.wikipedia.org/wiki/%D0%A0%D1%82%D1%83%D1%82%D1%8C). Як [закріплювач](https://uk.wikipedia.org/wiki/%D0%97%D0%B0%D0%BA%D1%80%D1%96%D0%BF%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F_%D1%84%D0%BE%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D1%96%D1%87%D0%BD%D0%B5) зображення Дагерр використав [кухонну сіль](https://uk.wikipedia.org/wiki/%D0%9A%D1%83%D1%85%D0%BE%D0%BD%D0%BD%D0%B0_%D1%81%D1%96%D0%BB%D1%8C). Знімок вийшов досить високої якості — добре видно деталі як у світлих місцях, так і в тінях. Свій спосіб одержання фотографічного зображення Дагерр назвав [дагеротипія](https://uk.wikipedia.org/wiki/%D0%94%D0%B0%D0%B3%D0%B5%D1%80%D0%BE%D1%82%D0%B8%D0%BF%D1%96%D1%8F).

Практично тоді ж англієць [Вільям Генрі Фокс Телбот](https://uk.wikipedia.org/wiki/%D0%92%D1%96%D0%BB%D1%8C%D1%8F%D0%BC_%D0%93%D0%B5%D0%BD%D1%80%D1%96_%D0%A4%D0%BE%D0%BA%D1%81_%D0%A2%D0%B5%D0%BB%D0%B1%D0%BE%D1%82) винайшов інший спосіб одержання фотографічного зображення, який він назвав [калотипією](https://uk.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BB%D0%BE%D1%82%D0%B8%D0%BF%D1%96%D1%8F). Як носій зображення Телбот використав [папір](https://uk.wikipedia.org/wiki/%D0%9F%D0%B0%D0%BF%D1%96%D1%80), просочений [хлористим сріблом](https://uk.wikipedia.org/wiki/%D0%A5%D0%BB%D0%BE%D1%80%D0%B8%D1%81%D1%82%D0%B5_%D1%81%D1%80%D1%96%D0%B1%D0%BB%D0%BE). Телбот одержав перший у світі [негатив](https://uk.wikipedia.org/wiki/%D0%9D%D0%B5%D0%B3%D0%B0%D1%82%D0%B8%D0%B2). [Експозиція](https://uk.wikipedia.org/wiki/%D0%95%D0%BA%D1%81%D0%BF%D0%BE%D0%B7%D0%B8%D1%86%D1%96%D1%8F_(%D1%84%D0%BE%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D1%96%D1%8F)) тривала близько години, на знімку — ґратчасте вікно будинку Телбота.

Назву «фотографія» було обрано як наймилозвучнішу з декількох варіантів у [Французькій академії](https://uk.wikipedia.org/wiki/%D0%A4%D1%80%D0%B0%D0%BD%D1%86%D1%83%D0%B7%D1%8C%D0%BA%D0%B0_%D0%B0%D0%BA%D0%B0%D0%B4%D0%B5%D0%BC%D1%96%D1%8F) [1839](https://uk.wikipedia.org/wiki/1839) року.

Першою жінкою-фотографом була ботанік [Анна Аткінс](https://uk.wikipedia.org/wiki/%D0%90%D0%BD%D0%BD%D0%B0_%D0%90%D1%82%D0%BA%D1%96%D0%BD%D1%81).

“Фотограф ізолює і зберігає момент часу: важливий або незначний, залежить від розуміння теми фотографом і його майстерності”, – сказав Едвард Вестон. Сьогодні важко уявити наше життя без фотографії. Адже насамперед це мистецтво закарбовувати певну мить та бачити прекрасне у буденному. Фотографія — це один із найважливіших винаходів людства.

Куди б ми не пішли, скрізь нас оточують фото. Вони можуть бути у вигляді білбордів, оголошень, телебачення, реклам магазинів, фото в інтернеті тощо. Люди завжди хотіли зупинити певну мить, щоб потім неодноразово переглядати її. Тому доволі часто зустріч з друзями, прогулянка на вулиці, гарний пейзаж, свята, відпочинок із сім’єю супроводжується фотографіями.

Перша вдало зафіксована фотографія була зроблена ще у 1826 році Нісефором Ньєпсом. Згодом фотоопарат почав стрімко розвиватись. Таким чином, кінець ХІХ – початок ХХ століття можна назвати періодом злету фотографії. Збільшується кількість фотографів та фотомайстерень. Однак, ця професія тривалий час не була визнаною у суспільстві та не була високооплачуваною.

Важливим етапом в історії розвитку фотографії стало винайдення фотоплівки. Згодом відбувся перехід від плівки до цифрової камери, що суттєво змінив наше уявлення про значення фото. Роль фотографії змістилася від збереження важливих подій до спілкування з друзями, створення власного іміджу та ведення соціальних мереж.

Звісно цифрова фотографія має ширше застосування у сферах життя людини, ніж плівкова. Ми живемо у нову епоху фотографії – цифрову. Вона доволі зручна і проста, оскільки здійснюється в одну мить. Тепер можна навіть автоматизувати всі необхідні для вдалого знімка налаштування. Крім того, цифрова фотографія зробила камери більш компактними і зручними. В результаті сьогодні ми маємо можливість без проблем зробити фото в будь-який час і в будь-якому місці.

РОЗДІЛ 1

МЕТОДИ РОЗРОБКИ САЙТІВ

**1. Ручна за допомогою HTML**

**2. За допомогою програмних засобів розробки сайтів**

**3. За допомогою інструментальних систем таких як CMS**

**4. З використанням популярних на цей час фреймворків**

**5. На SaaS-плат­формах у CLOUD**

### 1.1 Ручна розробка за допомогою HTML.

При появі стандарту HTML, цей метод був найпоширенішим. Основною програмою для розробки був Notepad. Але у цього методу є істотні недоліки. Цей спосіб досить трудомісткий. І до того ж зробити нормальний Web-сайт без CSS, JavaScript та інших мов програмування досить важко. Цей підхід Ви вивчали раніше. Хотілось би звернути Вашу увагу на 2 наступних питання HTML5 та Canvas:

***Що таке HTML5?***

HTML5 – це нова специфікація мови розмітки, що використовується в створенні веб-сторінок. На відміну від попередніх версій це не просто специфікація мови для гіпертекстової розмітки, а набір різнопланових модулів – від HTML-елементів до відео-, аудіо-, векторної графіки SVG, растрової JavaScript-графіки Canvas, локальних баз даних і навіть різних API браузера.  
Весь цей список модулів дозволяє HTML5 успішно конкурувати з технологіями Flash і Silverlight. Причому успішна конкуренція з Flash можлива ще й тому, що HTML5 потенційно набагато менше навантажує процесор комп'ютера, ніж Flash, не вимагає установки плагінів і оновлень, а значить, менш вразливий для хакерських атак.  
Фактично саме поява в HTML5 нових тегів <video> та <audio> робить його потенційним конкурентом існуючих технологій від Adobe і Microsoft.

***Нова розмітка***

HTML 5 вводить кілька нових елементів і атрибутів. Деякі з них технічно є еквівалентами <div> і <span>, але мають своє семантичне значення, наприклад <nav> (навігаційна панель) і <footer>. Ці теги будуть полегшувати роботу пошукачам, а також обробку сайту з Корман персонального комп'ютера або читають програм. Інші елементи надають нову функціональність, такі як <audio> і <video>.   
Деякі застарілі елементи HTML 4, такі як <font> і <center>, були видалені з HTML 5.

### *Приклад веб-сторінки HTML 5*  <! DOCTYPE html>   <Html>    <Head>     <Title> Example HTML 5 document </ title>    </ Head>    <Body>     <Header> … </ header>     <Nav> … </ nav>     <Section>        <Article>           …        </ Article>     </ Section>     <Aside> … </ aside>     <Footer> … </ footer>   </ Body>   </ Html>

***Нові API***

Крім визначення розмітки, в HTML 5 визначені application programming interface, API. Існуючі інтерфейси DOM (Document Object Model - «об'єктна модель документа») розширені, також були додані нові API:  
• Малювання 2D-картинок в реальному часі;  
• Контроль над відтворенням медіафайлів, який може використовуватися, наприклад, для синхронізації субтитрів з відео;  
• Зберігання даних у браузері;  
• Редагування;  
• Drag-and-drop;  
• Робота з мережею.

***Відмінності від HTML 4***

Основні відмінності HTML 5 від HTML 4 складають:  
• Нові правила лексичного розбору;  
• Нові елементи – header, footer, section, article, video, audio, progress, nav, meter, time, aside, canvas;  
• Нові типи input-елементів;  
• Нові атрибути;  
• Глобальні атрибути – id, tabindex, repeat;  
• Застарілі елементи прибрані – center, font, strike.

***Обробка помилок***  
HTML 5-сумісні браузери дуже гнучкі при обробці помилок, на відміну від XHTML. HTML 5 розроблений так, що не підтримують його браузери можуть спокійно ігнорувати елементи HTML 5. На відміну від четвертої, п'ята версія чітко прописує правила лексичного розбору, щоб різні браузери відображали один і той же результат в разі некоректного синтаксису.

**Canvas** (англ. Canvas - «полотно») – елемент HTML5, призначений для створення растрового двомірного зображення за допомогою скриптів, зазвичай на мові JavaScript. Початоквідліку блоку знаходиться зліва зверху. Від нього і будується кожен елемент блоку. Використовується, як правило, для відтворення графіків для статей і ігрового поля в деякихбраузерних іграх. Але також може використовуватися для вбудовування відео в сторінку і створення повноцінного плеєра. Використовується в WebGL для апаратного прискорення 3Dграфіки.

### 1.2 Розробка сайту за допомогою програмних засобів.

Існує багато готових рішень, для більш швидкої і зручної розробки сайтів. Вони надають можливість генерувати html код, розробляти сайт у візуальному режимі і мають багато інших можливостей.

**Виділимо декілька інструментальних систем для розробки HTML:**

• ***програми, що мають у своєму складі візуальні редактори (design-based editor)*** – засоби, які автоматично формують необхідний HTML-код, дозволяючи розробляти Web-сторінки в режимі WYSIWYG;  
• ***програми-редактори (code-based editors)***, які надають редактор і допоміжні засоби для автоматизації написання коду.

**Розглянемо найбільш популярні design-based редактори:**

•***Adobe DreamWeaver*** – один з кращих візуальних редакторів, що генерують HTML код. Він дозволяє працювати в декількох режимах одночасно, з HTML кодом або у візуальному режимі. Але основним недоліком є те, що програма генерує занадто «важкий» код, додаючи багато зайвого. Але, якщо знайомі з HTML, тоді текст HTML можна відредагувати. Ця програмна система випускалася до 2005 року компанією Macromedia, після чого була придбана фірмою Abode.  
• ***Microsoft FrontPage*** – це простий в засвоєнні і зручний Web-редактор для проектування, підготовки і публікації Web-сайтів. Завдяки інтеграції з сімейством продуктів MS Office, звичного інтерфейсу і великої кількості шаблонів програма дозволяє швидко засвоїти роботу навіть початківцям, які знайомі з основами роботи в MS Word. При цьому FrontPage не можна назвати рішенням для «чайників»: програма надає широкі функціональні можливості та різноманітні засоби оптимізації при колективній розробці. Вона дозволяє швидко створювати динамічні комплексні Web-вузли практично будь-якої складності.

Розглянемо популярні code-based редактори:  
•***Adobe HomeSites*** – це потужний пакет, до складу якого входить багато корисних функцій і підпрограм. Об'ємний дистрибутив редактора включає в себе, крім самого редактора, редактор TopStyle для редагування таблиць CSS, перевірку орфографії та багато іншого.  
•***HotDog –*** цілком професійний редактор. Вбудована підтримка дуже широкого набору інструментів, що використовуються в Web-дизайні: HTML, CSS, JavaSript, VBScript, ASP, а також DOM – об'єктну модель документа, що використовується при програмуванні на VBScript і JavaScript. При цьому перевірка синтаксису цих інструментів може налаштовуватися в досить широких межах. Наприклад, HTML можна перевіряти на відповідність версії 3.2, 4, або на «перегляді» тільки в Internet Explorer та інше.  
• ***AceHTML*** – основні функціональні можливості – подібно HomeSite і FirstPage. З цінних якостей AceHTML треба відзначити вмілу роботу з кодуваннями російської мови. Другий плюс – дуже непогана вбудована утиліта для перегляду графічних файлів у комп'ютері. У просторому вікні відображаються ескізи всіх картинок в директорії, а також їх параметри і розмір у пікселях.

РОЗДІЛ 2

ОСНОВНІ ФУНКЦІЇ УПРАВЛІННЯ

WEB-КОНТЕНТОМ

Виробники систем управління Web-контентом одностайні лише в самому загальному визначенні того, що повинні надавати такі системи. Їх основні параметри можна згрупувати в три категорії:

2.1 Розробка контенту.

Розробка контенту є одним з ключових компонентів всієї системи. Саме тут починається життєвий цикл будь-якого матеріалу що публікується на сайті. На цьому етапі відбувається створення, редагування та затвердження контенту, а роль системи полягає в автоматизації цих процесів. Завдання підтримки спільної роботи авторів, редакторів, програмістів та менеджерів повністю покладається на систему. Це завдання здійснюється завдяки розподілу контенту і дизайну. Всі компоненти сайту, включаючи шаблони і наповнення, зберігаються у визначених місцях сховища даних. Система ж автоматично звертається в потрібні місця сховища, дозволяючи безлічі користувачів, які навіть не є технічними фахівцями, працювати над підготовкою контенту до публікації, включаючи перевірку його достовірності.

2.2 Управління сайтом.

На цьому рівні відбувається розробка самого сайту, попередній перегляд і публікація підготовленого контенту. Тут розробляється зовнішній вигляд, готуються шаблони, розподіляються ролі користувачів і класифікація необхідної бізнес-інформації (наприклад, товари, ціни). Важливими компонентами цього рівня є служби що підтримують своєчасність надходження необхідного контенту.

2.3 Доставка контенту

Коли сайт повністю підготовлений до публікації, необхідні ресурси для динамічного формування Web-сторінок в залежності від виду конкретних користувачів. У зв'язку з цим, одним з важливих компонентів даного етапу є персоналізація або розподіл профілів, щоб кожен користувач отримував тільки ту інформацію, яка відповідає його ролі.

Треба відзначити, охоч і не існує абсолютно однакових систем управління Web-контентом, експерти сходяться в одному. При розвитку Web-технологій системи повинні будуть концентруватися більше на управлінні контентом, ніж на Web-паблішинг.  
Дуже швидко CMS перетворилися із просто систем управління контентом в повнофункціональні системи управління сайтом. При цьому популярні і широко розповсюдженні скрипти, які окремо використовувалися на сайтах, були об’єднані загальною програмою під єдиним інтерфейсом. В багатьох випадках інтегровані зручні автоматичні інсталятори і візуальні HTML редактори. В результаті отримали на 100% готовий до застосування повнофункційний сайт — портал. Інсталювати такий портал на сервер, встановити (вибрати) необхідні функціональні модулі (наприклад, модуль публікації статей з уже вбудованою можливістю додавання коментаріїв відвідувачами), наповнювати інформаційним контентом свій сайт уже може будь-яка людина без спеціальних знань. Часто програмні можливості CMS дозволяють (в повністю автоматичному режимі) змінювати розташування виведення інформації на Web-сторінці, вибирати схеми кольорів, дизайн із шаблонів, що є в наявності.  
В наш час за допомогою сучасних CMS, будь-яка людина, яка має бажання відкрити своє представництво в Інтернет, може без спеціальних знань, «титанічної» праці, особливих матеріальних витрат організувати свій якісний і багатофункційний сайт. Організації також отримали більш дешеву і мобільну можливість вести свій Інтернет-бізнес. Це ж стосується і рекламних агентств, значна кількість яких в наш час є розробниками своїх власних CMS.  
В більшості випадків CMS будуються із програмного ядра — «движка» та модулів, що підключаються.   
«Движок» CMS — це програма, яка підпорядкована певному алгоритму введення і виведення інформації. Цей алгоритм у кожної CMS різний, свій, в відповідності з ідеєю і ціллю розробника. Це дає можливість користувачу вибирати CMS, що найбільш підходить. В зв’язку з цим, зовнішній вигляд інформації що виводиться (наприклад, новин), в кожній CMS різний, індивідуальний і змінити його досить складно (або неможливо, без переробки, зрозуміло, самого «движка»), тому якщо зовнішній вигляд контента початково не влаштовує, краще вибрати іншу CMS, ніж займатися його «підгонкою».

Універсальність і, багато в чому популярність CMS, залежить від кількості наявних модулів, можливості и простоти створення нових. Наявність, кількість, оновлення модулів  є одним із критеріїв оцінки CMS. Модульна структура дозволяє розробникам охопити велике коло користувачів з різними вимогами до функційності порталу, програмістам — створювати свої модулі під конкретні задачі чи замовлення, а користувачам — отримати зручність і можливість мобільно реалізовувати різні свої ідеї.  
Багато які CMS, крім функційних модулів, мають шаблону структуру. За допомогою шаблонів можна швидко змінити зовнішній вигляд всього порталу або окремих модулів. Крім того, наявність шаблоної структури припускає і дозволяє корекцію і створення своїх, індивідуальних шаблонів.  
CMS з одного боку дає широкі можливості відвідувачам-користувачам, а з іншого – адміністраторам, які керують ресурсами,  адмініструють інформацію, керують загальним виглядом сайту, спілкуються з  відвідувачами і клієнтами. Більшість CMS забезпечені системою авторизації, і адміністратор може дозволяти, забороняти, обмежувати доступ до інформації, або частини інформації на своєму ресурсі окремим відвідувачам.

Можна впевнено сказати, що CMS необхідні всім, хто так чи інакше планує свою присутність в Інтернеті. А так як ринок CMS постійно змінюється і кількісно і якісно, виникає необхідність оцінки і вибору оптимальної для конкретного проекта CMS.  
Багато які CMS, крім функційних модулів, мають шаблону структуру. За допомогою шаблонів можна швидко змінити зовнішній вигляд всього порталу або окремих модулів. Крім того, наявність шаблоної структури припускає і дозволяє корекцію і створення своїх, індивідуальних шаблонів.  
CMS з одного боку дає широкі можливості відвідувачам-користувачам, а з іншого – адміністраторам, які керують ресурсами,  адмініструють інформацію, керують загальним виглядом сайту, спілкуються з  відвідувачами і клієнтами. Більшість CMS забезпечені системою авторизації, і адміністратор може дозволяти, забороняти, обмежувати доступ до інформації, або частини інформації на своєму ресурсі окремим відвідувачам.  
Можна впевнено сказати, що CMS необхідні всім, хто так чи інакше планує свою присутність в Інтернеті. А так як ринок CMS постійно змінюється і кількісно і якісно, виникає необхідність оцінки і вибору оптимальної для конкретного проекта CMS.

### РОЗДІЛ 3

### ПОНЯТТЯ І СУТНІСТЬ СИСТЕМИ УПРАВЛЯННЯ КОНТЕНТОМ

### 3.1 Основна ідея систем управління контентом

***Сьогодні виділяють два види сайтів:*** ***статичні і динамічні***. Статичні сайти створюються відносно просто і дешево, але трудомісткі і дорогі в супроводі. Замовні динамічні сайти складні і дорогі в розробці, але при вдалій постановці завдання можуть бути достатньо дешевими у супроводі.

***Системи управління контентом, або движки,***пропонують компроміс між цими двома варіантами. За рахунок обмежень, накладених на логічну структуру контенту, зовнішній вигляд (дизайн) і функціональні можливості створюваних динамічних сайтів, такі системи дозволяють радикально знизити трудомісткість розробки та підтримки.

***Скрипти CMS*** розташовуються на самому сайті і щоб скористатися ними потрібно просто зайти на спеціальну сторінку сайту, захищену паролем. Відповідно все, що потрібно для оновлення сайту з встановленою системою, - це наявність підключеного до Інтернету комп'ютера і браузера. Оновлення сайту за допомогою движка відбувається швидше і простіше, ніж редагування коду, і може здійснюватися особами, абсолютно не знайомими з HTML і іншими мовами розмітки. Багато сучасних CMS настільки автоматизовані, що при роботі з системою не потрібен фахівець.

***Контент – це будь-яка інформація, графічна чи текстова, розміщена на сервері.***

**Система управління контентом**:

1. ***це система управління інформацією на сайті;***  
   ***2) це система, за допомогою якої власник може керувати змістом свого ресурсу, не вдаючись при цьому до послуг професійних html-верстальників і програмістів;***  
   ***3) це спеціалізоване програмне забезпечення для автоматизованої розробки і підтримки динамічних сайтів.***

Такі системи дозволяють розробляти і підтримувати сайти, не вимагаючи при цьому знання HTML.

**Основна ідея систем управління контентом – поділ дизайну сайту і його змісту.** При створенні сайту за допомогою такої системи розробляються кілька шаблонів сторінок, в яких згодом розміщується інформація, тобто система містить шаблони сторінок і сховище (базу даних) інформації, з якої формується інформаційне наповнення сайту.

Незважаючи на те, що всі системи управління контентом спроектовані по-різному, вони призначені для однієї мети – оновлення змісту сайту, і важливу роль тут відіграє модуль документообігу, що входить в систему управління сайтом.   
У центрі системи знаходиться база даних, в якій зберігається весь контент, шаблони сторінок і інформація про управління користувачами/продуктами.

РОЗДІЛ 4

КЛАСИФІКАЦІЯ СИСТЕМ УПРАВЛІННЯ КОНТЕНТОМ

В основу класифікації можуть бути покладені наступні критерії:   
***користувацький сервіс*** – наявність тих чи інших функцій і модулів, зрозумілість і доступність користувачеві;   
***технологічність*** – використання технологій, що дозволяють підвищити пропускну здатність і надійність системи;   
***сумісність (апаратна і програмна)*** – можливість функціонування системи на різних платформах, сумісність з СУБД, можливість підключення додаткових модулів;   
***масштабованість*** – можливість розвитку та нарощування системи.

На основі вищевикладених критеріїв виділяють: ***просту, шаблонну, професійну та універсальну системи управління контентом.***

### 4.1 Проста система управління контентом

Інтерфейс користувача контент-системи збирається з програмних модулів, набір яких визначається в індивідуальному порядку для кожного окремого проекту. Модулі один раз налаштовуються розробником, чим жорстко закріплюється структура проекту. Від користувача системи потрібне знання розмітки HTML. Система функціонує на основі динамічного формування сторінок (щоразу при запиті користувача) і обмеженій пропускній спроможності (в залежності від апаратного забезпечення) – до 3 000-5 000 відвідувачів на добу.  
Проста система сумісна з певними платформами і типами СУБД. Спроба впровадження додаткових модулів, залежно від технологій що використовуються, може привести до повної переробки проекту. Масштабованість досягається шляхом перекладу сайту на нову, більш пристосовану до поточних завдань, версію системи управління контентом.

4.2 Шаблонна система управління контентом

Інтерфейс користувача системи представлений єдиним модулем або набором модулів з жорстко закріпленою структурою сайту. Шаблонна система функціонує на основі динамічного формування сторінок або використання кешування даних. Залежно від методів формування сторінок – від 5 000 до 50 000 відвідувачів на добу. Система сумісна з певними платформами і типами СУБД. Спроба впровадження додаткових модулів, також, як і в попередньому випадку, обмежена. Масштабованість відбувається, як і в простій системі управління контентом.

### 4.3 Професійна система управління контентом

Система представлена інтуїтивно зрозумілим для користувача інтерфейсом і розширеними можливостями редагування. На основі таких систем управління контентом створюються самі різноманітні сайти. Надається можливість підключення додаткових модулів, як від розробника, так і прикладного ПЗ. Формовані динамічні сторінки кешуються, що веде до безмежної пропускної здатності, яка залежить тільки від апаратного забезпечення. Система сумісна з різними програмно-апаратними платформами. Рівень пропонованої в системі масштабованості дозволяє підключати додаткові модулі без порушення структури та ідеології керування веб-ресурсом.

4.4універсальна система управління контентом

Користувацький сервіс представлений передовими засобами управління контентом, система надає засоби для розробки нових сервісів і можливостей. За технічними характеристиками дана система повністю пристосована до внутрішньокорпоративного використання в зв'язці з ERP-пакетами, що забезпечено наявністю сертифікованої системи безпеки – розмежування прав доступу до контент-системи на внутрішньокорпоративному рівні. У процесі функціонування системи проводиться кешування формованих динамічних сторінок і при цьому пропускна здатність – не обмежена.  
Система сумісна з різними програмно-апаратними платформами. Можливості масштабування дозволяють підключати додаткові модулі, що розширюють функціональність ресурсу, у тому числі готові модулі інтеграції з внутрішньокорпоративними системами. Можливе розширення за рахунок кластеризації.

### РОЗДІЛ 5

### ВИКОРИСТАННЯ ФРЕЙМВОРКІВ

Фреймворк  це програмний продукт, який є основою для створення сайтів, але він не має готових рішень для побудови сайтів, не має рішень для виконання певних функцій. Це більш низький рівень ніж CMS. Розробники на фреймворках створюють і інтерфейсну частину, і базу даних, і алгортми та програмні рішення проблемно орієнтованої частини і скоріше не сайту , а Веб додатку. Створюючи також його адміністративний інтерфейс.

**Фреймворк** (в інформаційних системах)- це структура програмної системи, що полегшує розробку і об'єднання різних компонентів великого програмного проекту. На відміну від бібліотек, які об'єднують набір підпрограм близької функціональності, фреймворк містить в собі велику кількість різних за призначенням бібліотек. Вживається також слово каркас, а деякі автори використовують його в якості основного, в тому числі не базуючись взагалі на англомовному аналогу. Можна також говорити про каркасний підход як про підхід до побудови програм, де будь-яка конфігурація програми будується з двох частин: перша, постійна частина - каркас, незалежний  від конфігурації до конфігурації і несе в собі гнізда, в яких розміщується друга, змінна частина - змінні модулі (або точки розширення).

Для того щоб відповісти на питання що повинен мати фреймворк , щоб розглядатися як WEB-технологія розглянемо, які види фреймворків є:-         Фреймворки програмної системи; Фреймворки додатків;Фреймворки  концептуальної моделі;

5.1 Типи фреймворків

**Фреймворк програмної системи** - це каркас системи або підсистеми. Він може включати допоміжні програми, мови сценаріїв, все, що полегшує розробку і об'єднання різних компонентів. Від бібліотеки він відрізняється виконанням коду, який написаний для нього, але не виконується сам. До цього виду фреймворків відносяться і фреймворки для WEB.

**Фреймворк додатку** має стандартну структуру. З ростом необхідності в графічних інтерфейсах користувача з'явилася і необхідність у фреймворках додатків. З їх допомогою простіше створювати засоби для створення графічних інтерфейсів автоматично. Для створення фреймворку додатків використовують об'єктно-орієнтоване програмування. Перший такий Фреймворк написала компанія Apple для Macintosh. Спочатку він був створений за допомогою Паскаль, потім же перероблений в С ++.

**Фреймворк концептуальної моделі** - це абстрактне поняття даної структури для визначення способів вирішення конкретної проблеми.

**WEB фреймворки** - це каркас, призначений для створення динамічних веб-сайтів, мережевих додатків, сервісів або ресурсів. Він спрощує розробку і позбавляє від необхідності написання рутинного коду. Багато фреймворків спрощують доступ до баз даних, розробку інтерфейсу, а також зменшують дублювання коду .

Є п'ять типів веб-фреймворків: Request-based, Component-based, Hybrid, Meta and RIA-based.

**Request-based:**фреймворки, які безпосередньо обробляють вхідні запити. Збереження стану відбувається за рахунок серверних сесій. Приклади: Django, Ruby на Rails, Struts, Grails.

**Component-based:**фреймворки, які абстрагують обробку запитів всередині стандартних компонентів і самостійно стежать за станом. Своєю поведінкою дані каркаси нагадують стандартні програмні графічні інтерфейси.. Приклади: JSF, Tapestry, Wicket.

**Hybrid-based:**фреймворки, які комбінують Request-based  та Component-based фреймворки, беручи під свій контроль всі дані і логічний потік в заснованої на запиті моделі. Розробники мають повний контроль над URL, формами, параметрами, cookies і pathinfos. Однак замість того, щоб відобразити дії і контролери безпосередньо до запиту, гібридні фреймворки забезпечують об'єктну модель компонентів, яка поводиться тотожно в багатьох різних ситуаціях, таких як окремі сторінки, перервані запити, подібні порталу фрагменти сторінок і інтегровані віджети. Компоненти можуть розподілятися окремо і ефективно інтегруватися в інші проекти. Приклади: RIFE.

**Meta -based:** у фреймворків  є ряд базових інтерфейсів для загального обслуговування і основу яка легко розширюється, для інтегрування компонентів і служб. Приклад: Keel.

**RIA-based**: фреймворки для розробки Rich Internet Applications (RIA). Служать для розробки повноцінних додатків, що запускаються всередині браузера. Приклад: Flex.

Найбільш поширеними є Request-based і Component-based веб-фреймворки. Більшість  WEB-фреймворків побудовані на архітектірі MVC.Model View Controller (MVC, «модель- представлення-контролер», «модель-вид- контролер») – схема використання декількох шаблонів проектування, за допомогою яких модель додатки, інтерфейс і взаємодія з користувачем розділені на три окремі компоненти таким чином, щоб модифікація одного з компонентів чинила мінімальний вплив на інші . Дана схема проектування часто використовується для побудови архітектурного каркаса, коли переходять від теорії до реалізації в конкретній предметної області.

Архітектурний шаблон Модель-Вид-Контролер (MVC) поділяє програму на три частини. У тріаді до обов'язків компоненту Модель (Model) входить зберігання даних і забезпечення інтерфейсу до них. Вигляд (View) відповідальний за представлення цих даних користувачеві. Контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача, і повідомляє про зміни компоненту Модель. Така внутрішня структура в цілому поділяє систему на самостійні частини і розподіляє відповідальність між різними компонентами.

MVC поділяє цю частину системи на три самостійні частини: введення даних, компонент обробки даних і виведення інформації. Модель, як вже було відмічено, інкапсулює ядро даних і основний функціонал з їх обробки. Також компонент Модель не залежить від процесу введення або виведення даних. Компонент виводу Вигляд може мати декілька взаємопов'язаних областей, наприклад, різні таблиці і поля форм, в яких відображається інформація. У функції Контролера входить моніторинг за подіями, що виникають в результаті дій користувача (зміна положення курсора миші, натиснення кнопки або введення даних в текстове поле).

Зареєстровані події транслюються в різні запити, що спрямовуються компонентам Моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через Контролер внесе зміни до Моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися

### РОЗДІЛ 6

### СТВОРЕННЯ САЙТІВ НА

### SaaS-платформах У Cloud

### SaaS-платформи для створення сайтів - це можливість запустити досить простий веб-проект дуже швидко і досить дешево (ще і на умовах оренди і без необхідності хостінгу). Щось подібне до Гугл сайтів, але зі значно більшими можливостями.  Рішення підходить для простих сайтів, тимчасових проектів і для перевірки бізнес-ідей. SaaS-платформи, як і CMS, бувають специфічними (наприклад, тільки для інтернет-магазинів) і універсальними (для всіх типових видів сайтів). Цей метод зараз активно розвивається та використовується.

### 6.1 проекти на SaaS-платформах

### SaaS-рішення мають свої переваги і недоліки, тому не кожен проект може бути реалізований подібним способом. На SaaS має сенс створювати прості сайти, які не особливо вимогливі до дизайну і ні відразу, ні в перспективі не зажадають яких-небудь доопрацювань логіки роботи та навігації. SaaS відмінно підходить для запуску проектів, мета яких полягає в перевірці на практиці бізнес-гіпотез. Можна і для створення простих проектів, вимога до яких одне - «потрібен сайт», а також для створення «заглушок» при розробці серйозних проектів. Хоча простий сайт доцільно створювати, наприклад, за допомогою засобів Гугл сайт.

6.2 Кроки створення сайтів на SaaS-платформах

Крок 1  Зареєструватися;

Крок 2 Вибрати тип сайту (наприклад, сайт-візитка, інформативний сайт або інтернет-магазин)

Крок 3 Вибрати потрібні модулі (новини, каталог, форум та інше);

Крок 4 Вибрати шаблон дизайну зі переліку шаблонів;

Крок 5 Ввести необхідну інформацію в систему управління контентом;

Крок 6 Ввести контент

SaaS-рішення входить все необхідне для повноцінної роботи проекту - не треба вибирати, встановлювати і налаштовувати CMS, не треба замовляти хостинг і налаштовувати сервер, а в подальшому не треба займатися технічним супроводом проекту. Все це робить за вас SaaS-рішення

6.3 Мінуси SaaS–платформ для створення сайту

Фактично це шаблонний дизайн - оформлення сайтів на SaaS-платформах проводиться за готовими шаблонами (часто не дуже високої якості), які можна тільки «розфарбувати» і на деяких платформах можна окремі блоки місцями поміняти. Для сайтів, до яких пред'являються вимоги до оформлення, такі рішення не підходять.

Рамки функціональних можливостей - якщо платформа «не має якихось можливостей», то це ніяк не виправити. Програмний продукт типовий і його налаштування під індивідуальні побажання обмежена. Якщо проект відразу або в перспективі повинен вирішувати специфічні завдання і гнучко налаштовуватися, то SaaS-платформа для його розробки не підходить.



### 6.4 Приклади SaaS-платформ

UMI, WIX, InSales, Shopify, Setup, uCoz - деякі з цих платформ специфічні (тільки для простих сайтів або тільки для інтернет-магазинів), а деякі - досить універсальні. Якщо ваш проект не має ніяких істотних вимог до дизайну і до функціональності, то має сенс звернути увагу на ці SaaS-рішення. В іншому випадку, вибирайте іншу платформу для створення сайтів

ВИСНОВОК

В данній курсовій роботі розроблений сайт на тему «Фотоіндустрія»

При вивченні данного матеріалу була дана змога дізнатися, що таке фотографія її історію,

виникнення та поширення як вид мистецтва, для чого її використовували, та як змінилося ставлення до фотографії у сучасном світі.

Також описано як можна створити свій власний сайт, також є можливість вибрати платформу для створення сайту або писати код власноруч.

Данна інформація може бути використана для саморозвитку так як є не важкою для засвоєння.