Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

ОРГАНІЗАЦІЯ ПРОЦЕСУ БЕЗПЕРЕРВНОЇ РОБОТИ З ІНФОРМАЦІЙНИМ КОНТЕНТОМ ВЕБ-СИСТЕМ

Текстова частина до курсової роботи

за спеціальністю «Інженерія програмного забезпечення» 121

Керівник курсової роботи

Асистент Корнійчук М.А.

_____

(підпис)

“___” _____ 2021 р.

Виконав студент Кучерявий В. Ю.

“___” _____ 2021 р.

Київ 2021

ЗМІСТ

АНОТАЦІЯ **3**

ВСТУП **4**

РОЗДІЛ 1. Безперервна розробка Контенту(Continious Content Development) **6**

РОЗДІЛ 2. Планування **8**

2.1. “Content-first” підхід **8**

2.2. RACI матриця **9**

РОЗДІЛ 3. Збір(контент-стратегія) **12**

3.1. Аналіз цільової аудиторії **12**

3.2. Складання ключових слів **15**

3.3. Планування публікацій **15**

3.4. Референси **15**

РОЗДІЛ 4. обробка **17**

РОЗДІЛ 5. Тестування **20**

5.1. Ручне тестування **20**

5.1. Автоматичне тестування **21**

РОЗДІЛ 6. Реліз **30**

6.1. Стратегія розгалуження на базі основної гілки **30**

6.2. Модель GitFlow **31**

6.3. Реліз в моделі GitFlow **31**

6.4. Реліз в моделі GitHub Flow **32**

РОЗДІЛ 7. Моніторинг **34**

ВИСНОВКИ **36**

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ **37**

ДОДАТКИ **38**

ДОДАТОК 1. Основна сторінка сайту факультету інформатики **38**

АНОТАЦІЯ

В процесі виконання роботи відбулося більш детальне ознайомлення із методами роботи з контентом, способи його покращення і була розроблена методологія циклічної роботи з ним: Continuous Content Development. Методологія була закладена у створення нового сайту факультету інформатики. З сайтом можна ознайомитися за посиланням: https://mkornijchuk.wixsite.com/my-site-2.

Ключові слова: контент, інструмент, аналіз, методологія, моніторинг, реліз, тестування, обробка, збір, планування.

ВСТУП

За останні десятиріччя з’явилося багато різних підходів та методів організації роботи з розробки програмних продуктів. Внаслідок цих змін розробники стали краще розуміти який продукт і для кого вони створюють.

Однак, трапляються випадки, коли дуже якісний (з технічної точки зору) програмний продукт через деякий час занепадає. Однією з причин є те, що створений на початковому етапі контент втрачає свою актуальність, оскільки він не розвивається та не оновлюється.

Таке трапляється через те, що на етапі розробки не надають належного значення роботі з інформаційним контентом.

Провівши дослідження цієї теми я виявив відсутність вже сформованого методу та описаних і формалізованих принципів роботи з інформаційним контентом. Зважаючи на актуальність тематики, я розробив власний метод, який отримав назву Continious Content Development. Цей метод покликаний описати основні етапи та правила роботи з інформаційним контентом.

У фундамент цієї курсової роботи було покладено роботу над сайтом Факультету Інформатики Національного Університету «Києво-Могилянська академія». На етапі дослідження проблеми я з’ясував, що відвідуваність сайта нашою цільовою аудиторією: студентами, абітурієнтами, їх батьками та викладачами, була надто мала. Оскільки, сайт факультету є інформаційним джерелом, я висунув гіпотезу, що причиною низької відвідуваності була неможливість сайту до кінця виконувати свої функції, через недостатню актуальність контенту і складність його безперервного оновлення. Поставивши за мету курсової роботи розробку методу, що допоможе ефективно працювати з інформаційним контентом я зіткнувся із проблемою відсутності ресурсу, на якому можна перевіряти нові гіпотези щодо нового методу та тестувати різні підходи. Дослідивши проблему я дійшов висновку, що рішенням може бути створення спрощеної робочої версії сайту, на якій можна тестувати та перевіряти ідеї та гіпотези щодо процесу розробки сайту і методу постійного оновлення контенту.

Метою цієї курсової роботи була розробка основних етапів та принципів методу Continious Content Development, який допоможе у створенні, підтримці актуальності та аналізі контенту сайту у майбутньому. Всі основні принципи методу було використано при створенні робочого прототипу сайту факультету.

РОЗДІЛ 1. БЕЗПЕРЕРВНА РОЗРОБКА КОНТЕНТУ(CONTINIOUS CONTENT DEVELOPMENT)

В процесі дослідження було виявлено, що немає чітких методологічних вказівок як правильно працювати з інформаційним контентом.

Через це, спираючись на власний досвід, використовуючи методи, створені для роботи з програмним кодом, було розроблено власний метод роботи з інформаційним контентом.

Назва цього методу Безперервна розробка контенту (Continuous Content Development, CCD).

Застосування методу передбачає такі етапи:

1. Планування
2. Збір
3. Обробка
4. Тестування
5. Реліз
6. Моніторинг

Варто згадати, що визначені етапи варто розглядати не лінійно, а циклічно. Будь який інформаційний ресурс є динамічним і потребує постійного оновлення і актуалізації, саме тому одноразове створення контенту без подальших етапів аналізу і змін призведе до втрати актуальності контенту. Будь-який контент потребує постійного і своєчасного оновлення, яке передбачає як збір нової інформації, так і тестування, реліз і моніторинг. Цей процес є циклічним, адже без оновлення контенту інформаційний веб-ресурс занепадає, показник відвідуваності сайту зменшується, відгуки користувачів стають переважно негативними. З огляду на перелічені причини ми приходимо до поняття безперервної розробки контенту.

Рис 1. 1 Етапи CCD

Цикл починається з етапу планування і завершується етапом моніторингу звідки він знову переходить у етап планування.

РОЗДІЛ 2. ПЛАНУВАННЯ

2.1. “Content-first” підхід

Що таке контент для сайту?

Контент - це матеріали, які формують сайт: зображення, відео, музика, тексти тощо.

Для створення ефективного контенту важливо, спочатку потрібно визначитися з контентом, вибрати картинки і фото, а також зробити прототип. Після цього етапу можна переходити до створення дизайну.

Оскільки у сьогоденні доволі легко створювати сайту, тому, звичайно, хочеться одразу починати втілювати у життя неясну картинку, яка зараз у голові, не маючи перед собою ніякого прототипу або ескізу. Через це часто можна побачити незручні у користуванні сайти, які потрібно дуже довго перероблювати. Але доробки ніяким чином не зможуть швидко перетворити непродуманий з самого початку сайт на новий рівень якості. Тому в першу чергу потрібно продумати структуру сайту, до того як починати його наповнювати.

Як нам всім часто хочеться -- зробити якось, подивитися як воно виходить, і вже потім переробити. Так, на жаль, не працює. Проект є фундаментом для дизайну, тобто без проектування дизайн просто працює як декорація і просто ховає недоліки і повну відсутність структури.

“Content-first” підхід - це філософія, щоб спрямовувати веб-сайти своїх клієнтів у більш ефективне русло та, зрештою, створювати кращий досвід для користувачів (UX).

Принципи “Content-first” підходу:

Перший принцип -- створення чіткої структури.

Якщо є розуміння, з чого повинен бути зроблений сайт, то правильно буде відокремити основну частину від іншої. Ієрархія - це запорука успішного сайту. Кожен блок інформації має встати на своє місце.

Наприклад, є усвідомлення того, що сайт буде тісно пов’язаний з блогом. Набагато простіше підготувати місце для такої інформації на етапі генерального формування структури, ніж намагатися зробити готову систему з цієї інформації, яка не була достатньо досліджена і виявилась результатом роботи методом тику без попередньої підготовки. Тому без системи не буває системи.

Другий принцип -- уміння створювати способи покращення контенту.

Якщо починати будувати структури на папері з нуля, накидавши приблизну схему, то доволі швидко можна буде зрозуміти, які моменти будуть визначальними та ключовими.

Наприклад, якщо ми знаємо, що на сайті буде багато посилань на інші джерела, і про це за заздалегідь подумати, то можна передбачити проектування графічного відображення посилань, що зробить дизайн максимально ефективним в поданні інформації.

Третій принцип -- UX/UI інтерфейс відмінно реалізований, тобто сторінка є дуже чіткою і зрозумілою будь-якому користувачу.

Незалежно від віку, статі, розумових здібностей тощо кінцевий користувач має зрозуміти що це таке. Веб-сайти завжди цінувалися тим, що чим, простіше -- тим ефективніше. Але, як не парадоксально, саме простоти домогтися буває найважче. Простий і зрозумілий інтерфейс виходить, якщо приступити до його створення, вже сформулювавши для себе що, де і звідки. І найголовніше - чому.

Четвертий принцип -- низька ймовірність нескінченних переробок.

Відсутність нескінченної кількості переробок досягається збором якнайповнішої інформації про ту сторінку, яка буде сформована, враховуючи всі деталі перед тим, як приступати до створення, щоб не вносити правки вже в готову сторінку.

2.2. RACI матриця

Перед тим як переходити до збору інформації, потрібно чітко розподілити ролі у команді. Тому що для проектування або зміни процесів потрібно реалізувати відповідальність і відносини між ролями, задіяними у процесі.

Методика RACI є зручним і наочним засобом проектування і планування змін, а саме участі різних ролей в процедурах і завданнях процесу.

Термін RACI є абревіатурою:

- R - Responsible (виконує);
- A - Accountable (несе відповідальність);
- C - Consult before doing (консультує до виконання);
- I - Inform after doing (оповіщається після виконання).

Розглянемо приклад RACI матриці

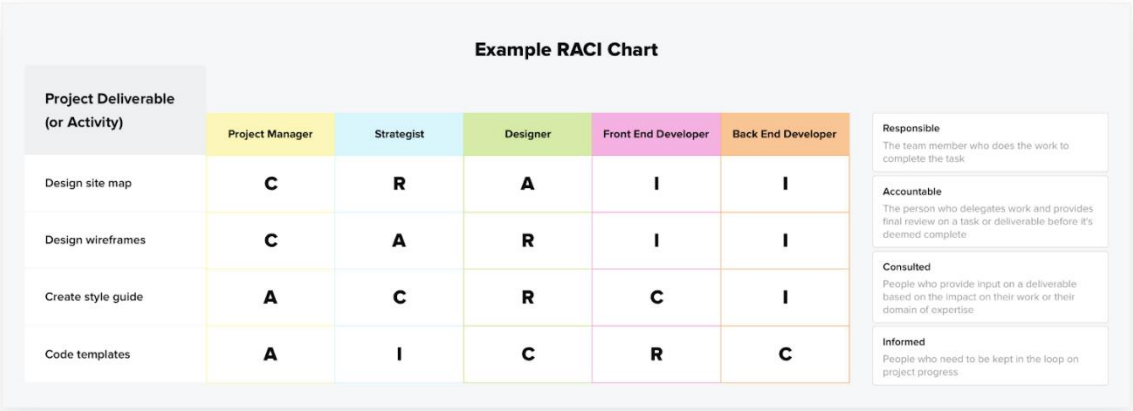


Рис 2. 1 Приклад RACI матриці

Є кілька правил, яких слід дотримуватися при побудові матриці RACI:

Accountable - повинен бути тільки один. Якщо це не так, то потрібно чітко обмежити рамки, в яких, або в даний момент по даній діяльності, або в даних умовах відповідальний тільки один, але в інших умовах з тієї ж діяльності можливо відповідальність несе інший.

Responsible - повинен бути в наявності по кожній діяльності, їх може бути кілька, причому можливі поєднання.

Кожна діяльність обов'язково повинна мати Accountable і Responsible.

Можна сміливо сказати, що матриця RACI це зручний інструмент візуалізації, що є частиною проектування будь-якого процесу, бо в будь-якому процесі є роль і діяльність, які потрібно розподілити і контролювати.

РОЗДІЛ 3. ЗБІР(КОНТЕНТ-СТРАТЕГІЯ)

Кожен з людей, які відповідають за певні аспекти проекту роблять свою конкретну роботу. Наприклад, хтось відповідає за збір даних з ЦА, хтось відповідає за дизайн, хтось за зручний UI/UX і так далі. Ці всі завдання потрібно грамотно з’єднати в один проект, тобто -- веб-продукт.

3.1. Аналіз цільової аудиторії

Першим етапом формування контент-стратегії є складання портрета цільової аудиторії (ЦА).

Орієнтуючись на портрет своєї ЦА, буде легше зрозуміти, який контент їй цікавий і корисний, чого вона потребує, яку її потребу необхідно закрити. Створюючи статті без аналізу аудиторії, керуючись виключно особистими уподобаннями, є великий ризик даремно витратити час і гроші. Потенційний клієнт найімовірніше прислухається до порад на сайті, якщо вважатиме інформацію цікавою. Таким чином, правильно підібрана тема статті або відео збільшує ймовірність зацікавати користувача.

Першим етапом при опитуванні є розділення нашої ЦА на різні групи. Це допоможе нам у створенні оптимальних питань для кожної з груп. Якщо цей етап пропустити, то питання будут більш загальні для усіх груп, які опитували і через це не дасть максимально зібрати болі користувачів.

Другим етапом у опитуванні цільової аудиторії є створення опитувального блоку питань. Це є найважливіший етап при опитуванні, бо обов’язково потрібно правильно розробити питання, які зможуть повністю розкрити болі користувачів. Питання повинні бути відкритого типу, щоб відкинути можливість коротких відповідей. У закритому ж питанні пропонується обмежена кількість варіантів відповідей. Закриті питання використовуються, коли ви вже розумієте, по яким категоріям будуть розподілятися відповіді, або вас цікавить лише частота окремих відповідей.

Розглянемо приклад нашого опитування цільової аудиторії сайту факультету інформатики. Нас цікавило, чи використовують викладачі сайт

факультетів різних університетів, при пошуку варіантів для працевлаштування.

Якби питання звучало як: “Чи використовували ви сайт факультету/університету для пошуку варіантів працевлаштування?”, то ми б отримали відповідь “Так.” або “Ні.”, що ніяким чином не допомогло би нам зрозуміти проблему і детальніше розібратися у цьому питанні.

Але в цьому випадку краще розділити одне питання на підпитання. Почати краще з: “Як ви шукаєте університет, якщо хочете працевлаштуватись?”.

З цього ми можемо зрозуміти які взагалі існують варіанти і чим вони гірші або кращі. Далі можемо перейти до уточнення і заглиблення у процес працевлаштування і перейти до проблем.

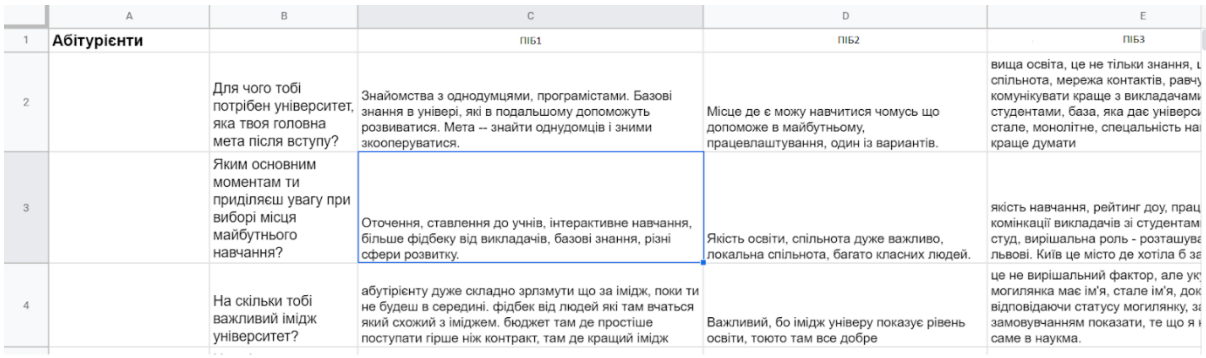
Тому наступним питанням буде: “Як зазвичай відбувається цей процес?”. І останнє питання в якому ми хочемо дізнатися про те, чи колись викладач використовував сайт для пошуку варіантів, і якщо використовував, то з якими проблеми він можливо зіткнувся. Останнє питання буде мати вигляд: “Чи використовували ви для цього сайт факультету? Якщо так, то розкажіть про цей досвід”.

Використовуючи такий підхід ми можемо гарантувати максимально детальну відповідь з якої можна зробити висновки.

Третій етап це і є інтерв’ювання цільової аудиторії. По можливості його бажано проводити вживу або онлайн, щоб повністю розкрити опитувану людину.

Оскільки ми маємо декілька груп для опитування, то бажано для кожної зробити якось зручно групувати відповіді і щоб їх можна легко було порівнювати. Гарним варіантом буде використання таблиці Excel, в якій стовпчик і рядок будуть відповідати людині, що опитують і її відповіді.

Приклад структури опитування абітурієнтів.



	A	B	C	D	E
1	**Абітурієнти**		ПІБ1	ПІБ2	ПІБ3
2		Для чого тобі потрібен університет, яка твоя головна мета після вступу?	Знайомства з однодумцями, програмістами. Базові знання в університі, які в подальшому допоможуть розвиватися. Мета -- знайти однодумців і зними зкооперуватися.	Місце де є можу навчитися чомусь що допоможе в майбутньому, працевлаштування, один із варіантів.	вища освіта, це не тільки знання, ц спільнота, мережа контактів, равч комунікувати краще з викладачам студентами, база, яка дає університ стале, монолітне, спецальність на краще думати
3		Яким основним моментам ти приділяєш увагу при виборі місця майбутнього навчання?	Оточення, ставлення до учнів, інтерактивне навчання, більше фідбеку від викладачів, базові знання, різні сфери розвитку.	Якість освіти, спільнота дуже важливо, локальна спільнота, багато класних людей.	якість навчання, рейтинг доу, прац комінкації викладачів зі студентам студ, вирішальна роль - розташува львові. Київ це місто де хотіла б за
4		На скільки тобі важливий імідж університет?	абутірієнту дуже складно зрлзмути що за імідж, поки ти не будеш в середині. фідбек від людей які там вчаться який схожий з іміджем. бюджет там де простіше поступати гірше ніж контракт, там де кращий імідж	Важливий, бо імідж університу показує рівень освіти, тоюто там все добре	це не вирішальний фактор, але ук могилянка має ім'я, стале ім'я, док відповідаючи статусу могилянку, з замовувчанням показати, те що я саме в наукма.

Рис 3. 1 Приклад оформлення зібранної інформації з ЦА

З таким підходом дуже легко орієнтуватися та отриматувати інформації по кожному питанню та порівнювати відповіді.

Четвертий етап це висновки з тої інформації, що ми отримали. Так само, потрібно робити висновки по кожній групі, з яких проводились дослідження. Після цього усі висновки можна порівняти, знайти щось спільне і зрозуміти в чому дійсно болі нашої цільової аудиторії.

Приклад висновків з кожної групи, яку опитали.

	Бізнес-партнери: • IT компанії • не-IT-компанії, але з великим штатом програмістів •	• найм: потрібні експерти, з якісними знаннями + дешеві (молоді) • можливість поспостерігати за майбутніми працівниками довгий час (довготривалі навчальні пректи/стажування…) • перекласти дослідження проблем в організації на людей, в яких є час і які зроблять це безкоштовно, можуть прийти до оригінальних рішень (формат курсової/дипломної/…) • соціальне підприємництво, піар серед клієнтів • піар серед студентів, як майбутнього роботодавця • співпраця між Research&Development Department і Могилянською Phd програмою
	Люди, що хочуть викладати	• знають і йдуть на ім'я Могилянки • цікаво попрацювати зі студентами, вплиути на щось, подивитись на процес зсередини • хочуть засвітитись у компанії та перед собою (викладати - це круто) • хочуть отримати досвід (можливо, перед викладанням на платних курсах) • випускники факультету - хочуть донести знання, які самі не отримали • в ідеалі ми підбираємо людей під курси, які потрібні факультету, а не навпаки • в ідеалі ми беремо на роботу практикуючих спеціалістів з актуальними знаннями

Рис 3. 2 Приклад висновків поопитанним группам

3.2. Складання ключових слів

Щоб контент займав перші позиції в пошукових системах в органічній видачі контент повинен бути створений з використанням ключовим слів. Ідеї ключових слів - це, припущення, як ми вважаємо користувачі можуть шукати наші товари або послуги в Google. Чим більше популярно ключове слово, тим воно цінніше. Також важлива словоформа ключового слова - чим більше точний вислів і її відміни обрані, тим більше важлива частота такої фрази. Тому перед написанням контенту необхідно зібрати всі ці можливі ключові слова. Але не головне не перестаратися з використанням ключових слів, пошукові системи дуже цього не люблять. Насамперед контент повинен бути створений для людей, бути корисним і відповідати на їхні запитання.

3.3. Планування публікацій

Контент-план є важливим не менше, ніж створення матеріалів. Він оформлюється у вигляді таблиці і доповнюється інформацією в процесі його виконання. Для кожної публікації повинна бути прописана мета її створення, формат розміщення, передбачуваний ефект (переходи на сайт, лайки, підписки, збільшення вхідних дзвінків і т.д.), дата публікації. До того ж, така таблиця дає можливість відстежити терміни виконання та ефективність кожного завдання. Задокументована інформація підвищує рівень відповідальності, і стаття, запланована на цілком конкретну дату, буде написана в термін з набагато більшою ймовірністю, ніж та, яка просто запланована на "протягом двох тижнів".

Через певний час, наприклад, місяць, можна відстежити, чи було ефективним розміщення публікації на тому чи іншому каналі.

3.4. Референси

Ні один якісний і красивий ресурс неможливо побудувати з нуля, не використовуючи референсів до інших сайтів. Референсом може слугувати будь що, що сподобалось, це можуть бути: шрифт, оформлення

кнопок,приятна палітра колорів, гарна побудова блоку, гарний футер чи може хедер чи навіть просто ідея мінімалістичного дизайну на іншому ресурсі. З кожного референсу можна взяти щось своє, що подобається і виглядає гармонійним, потім об’єднати все до купи і от, ми вже маємо з чим працювати і можемо думати як краще це оформити.

РОЗДІЛ 4. ОБРОБКА

На цьому етапі показати як роботу усіх членів команди об’єднати в один проект, тонкощі при об’єднанні, проблеми які можуть виникати і тд.

Етап обробки інформаційного коненту є містком між збором та тестування і він є дуже важливим. На цьому моменті повинні розкриватися проблеми, які, можливо, не було враховано раніше. Хоча здавалося б, макет сконструювали, контент весь зібрали, залишилося тільки наповнити макет контентом і нічого не може піти не так. Але ні. Тексти, які ми маємо, повинні пройти перевірку на орфографію та читабельність, бо навіть маленькі недоліки, як неправильно граматично написане слово, або важкість читання тексту можуть призвести до потенційної втрати користувача, який міг би в майбутньому стати клієнтом. Як раніше зазначалося, контент повинен бути простим і зрозумілим будь-якому користувачеві. Будемо вважати, що тексти пройшли перевірку на читабельність та орфографію, але далі може чекати наступний підводний камінь. Якийсь блок інформації був розрахований на N кількість символів, а наш текст вийшов трохи більшим, навіть на якісь 5-10 символів. Результатом буде неправильне розміщення контенту на сторінці, блок буде розташовуватись не так, як це планувалося раніше.

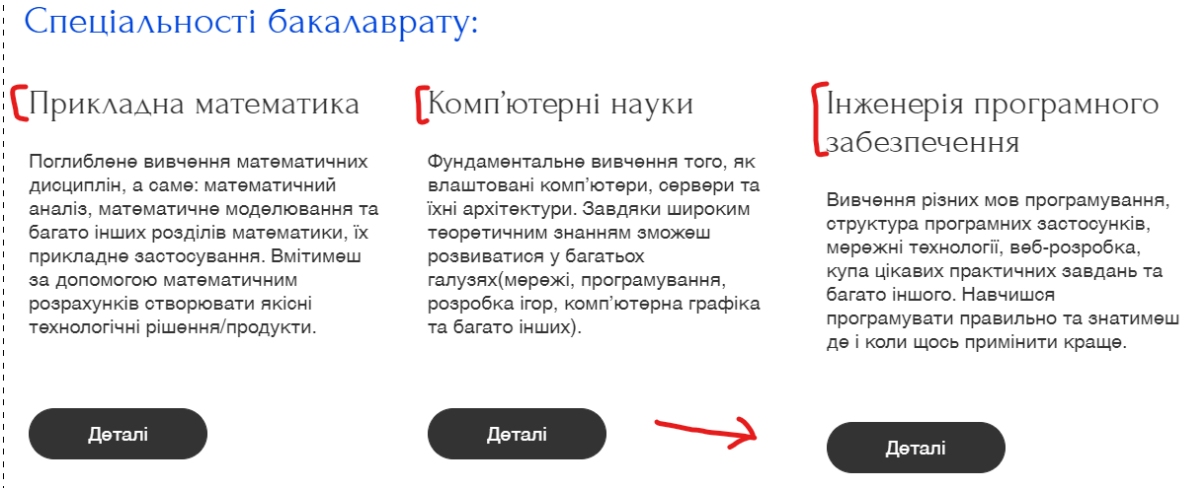


Рис 4. 1 Приклад неврахування розмірів блоку інформації

На даному прикладі, ми можемо побачити, що при створенні макету не було враховано, що назва спеціальності може займати стільки простору, звідси і маємо, що блок тепер розподілений нерівномірно і одразу кидається в очі.

Така, ситуація може виникнути не тільки з блоками текстів, але із картинками, бо їх роздільна здатність може бути замалою або завеликою для конкретного місця розташування.

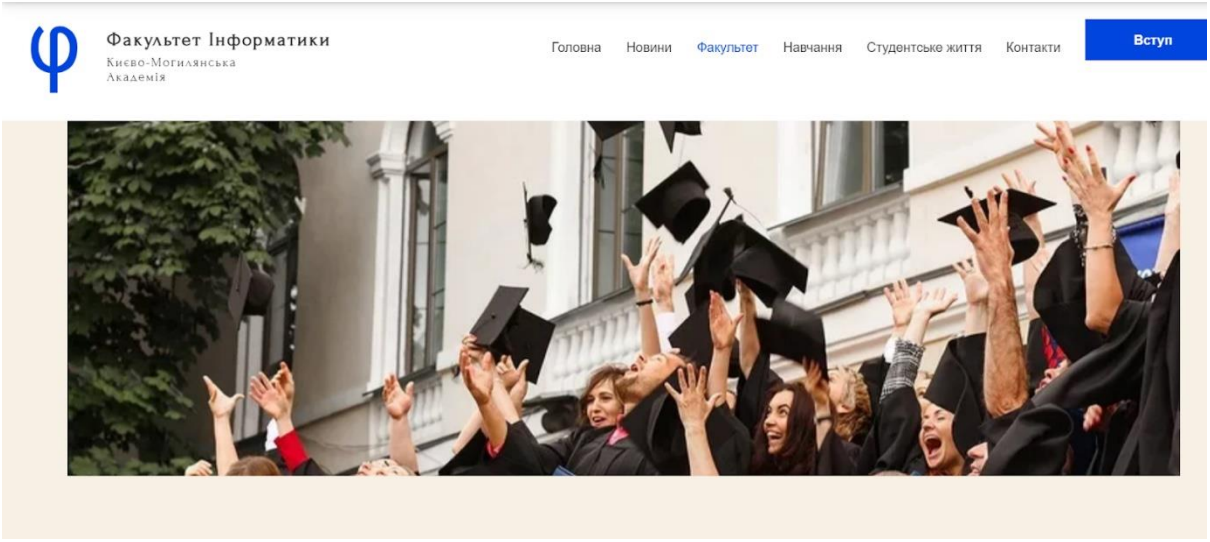


Рис 4. 2 Приклад погано підібранної фотографії

Зображення є обрізаним, бо ніяким чином не влізає в відвену для нього рамку, а також роздільна здатність виявилась замалою і зображення є доволі нечітким.

Так само не потрібно забувати про те, що якась інформація може ставити не релевантною і її потрібно буде прибрати, але оскільки структура раніше включала в себе блок з цим текстом, то може бути випадок, коли це ініціює з'їжджання блоків.

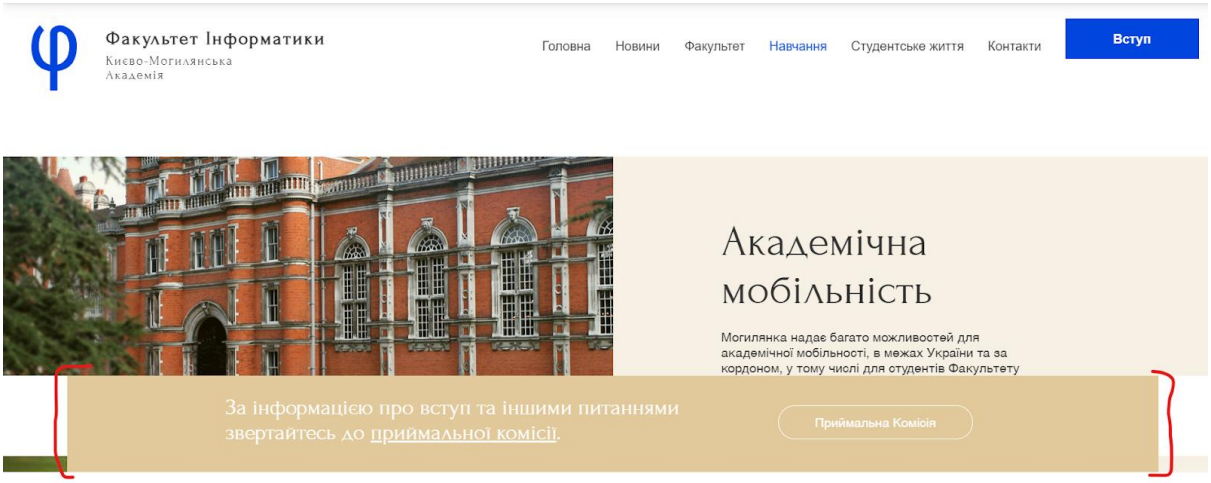


Рис 4. 3 Приклад неврахування змін при проектуванні

Тут ми можемо побачити як статичний блок наліз на інший статичний блок, через те, що динамічний блок був видалений, але це не було враховано при проектуванні.

Коли всі етапи конструювання пройдено і здається, що все готово, потрібно перейти до тестування продукту.

РОЗДІЛ 5. ТЕСТУВАННЯ

Тестування веб-додатків - це комплекс послуг, який може включати в себе різні види тестування ПО. Найголовніше в будь-якому тестуванні - це виявити всі помилки в програмному забезпеченні і побачити рекомендації щодо їх нівелюванню.

Навіщо взагалі тестувати веб-додаток?

Якщо буде потреба додати новий функціонал на сайт, поп-ап, змінити шрифт, додати кнопку, поміняти футер, має бути впевненість в тому, що в браузері користувача все відобразиться коректно.

На що ще впливають тести?

Тестування це не просто запуск програми з тестом, які або відмічаються як успішні, чи ні, це і рекомендації по усуненню якоїсь проблеми. Без тестування дуже важко знайти слабкі місця в продукті, бо не протестований продукт -- неправильно функціонуючий продукт, це просто питання часу. Коли користувач стикається з якимись проблемами на сайті, або функціонал працює не так, як було задумано, то користувач ставить під сумнів своє розуміння роботи сайту, і, що найімовірніше, покидає його. Варіантів коли користувач захоче покинути сайт -- безліч, у сьогоденні планка юзабіліті сайту та його змістове наповнення є на дуже високому рівні, тому банально не корректно розміщена кнопка, або довге підвантаження сторінки можуть змусити юзера покинути ресурс.

Загалом тестування показує наскільки логічно побудований проект та наскільки ефективно він зроблений.

Існує дві актуальні методики тестування сайтів.

Тестувати сайт можна вручну або автоматично. У кожного з підходів є свої плюси і мінуси.

5.1. Ручне тестування

В цьому режимі тестувальник не використовує програми, а проходиться по сайту як звичайний користувач, це часто відбувається на етапі “build-y”

проекту. Всі можливі помилки передбачити в автоматичному тестованні просто неможливо. Деякі нюанси тестування, особливо якісь дизайнерські моменти, виявити програма просто не може, що насамперед підсильне людському оку. Бо хто ж як не людина, краще розуміє, чого хоче інша людина. Мабуть, ще жодний проект не виходив у світ без ручного тестування, бо абсолютно все передбачити у автоматизованих тестах просто неможливо.

5.1. Автоматичне тестування

Тестувальник використовує спеціальні програми для виконання тестів, людський фактор відсутній. Їх можна використовувати безліч разів, якщо вони вже написані, щоб заощадити час. Програма визначає, чи відповідає сценарій очікуванням користувача. Автоматичне тестування особливо актуально для високонавантажених проектів з великою кількістю функціональностей.

На цьому етапі я трохи хочу заглибитись у інструменти для автоматичного тестування.

Перший інструмент, який хотілося б розглянути це - google page speed.

Google Page Speed Insights - це комплексний інструмент для визначення фактичної продуктивності і вибору ефективних шляхів оптимізації сайту. Зручний для використання як на комп'ютері, так і на мобільному пристрої. Google Page Speed Insights не показує абсолютну швидкість сторінки, а аналізує ефективність динаміки завантаження і відтворення в браузері клієнта. При цьому враховуються тільки не залежні від типу інтернет-з'єднання фактори: JavaScript, CSS, структура HTML, конфігурація сервера, розмір зображень тощо.

Для початку роботи з Google Page Speed Insight потрібно перейти на офіційний сайт: https://developers.google.com/speed/pagespeed/insights/?hl=uk.

Для того, щоб отримати результат, потрібно ввести посилання на сайт, який ми хочемо протестувати і натиснути аналізувати.

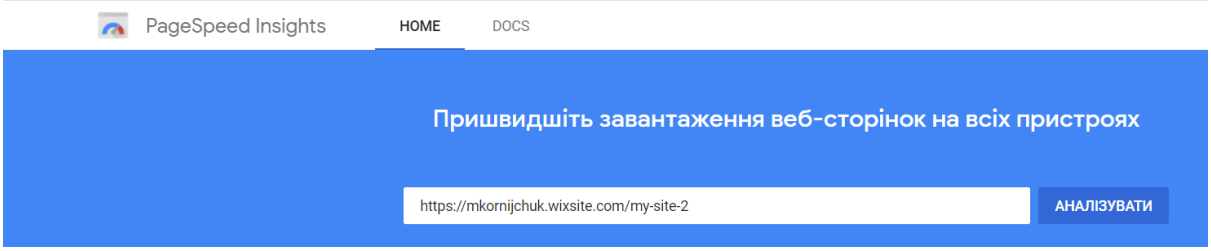


Рис 5. 1 Початкова сторінка PageSpeed

Сканування сайту займає менше хвилини і одразу йде видача результатів. І після цього користувач одразу може оцінити чи потребує сайт SEO-оптимізації. Або ж зрозуміти, що алгоритм, який використовується є доволі таки ефективним.

Звіт по сайту надається окремо для мобільної версії, а також для ПК, представлений у вигляді графічного інтерфейсу.

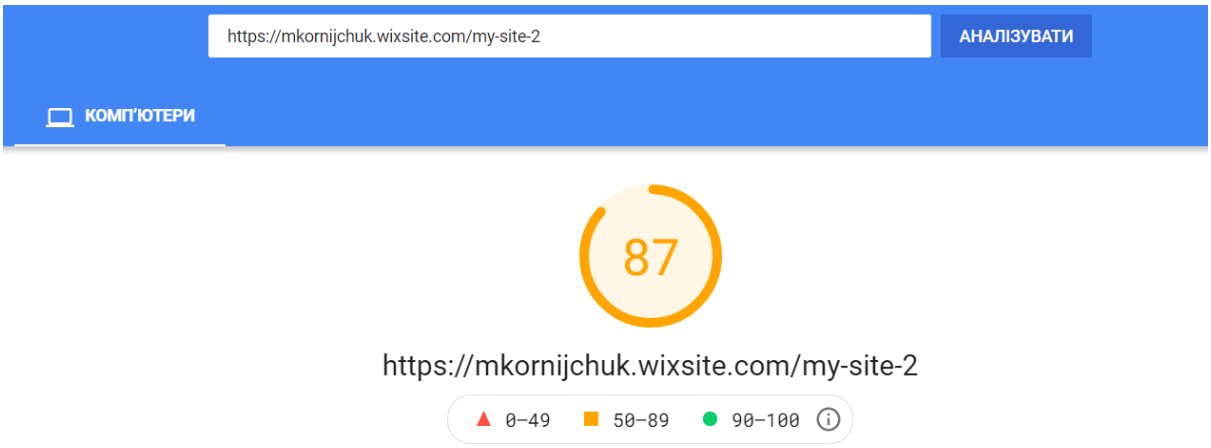


Рис 5. 2 Загальна оцінка PageSpeed

Для оцінення веб-додатку Google Page Speed Insights використовує 100-бальну шкалу оцінювання. Вони поділені на три категорії:

- Low -- 0-49
- Medium -- 50-89
- Good -- 90-100

При низькій оцінці веб-додатку майже завжди є великі недоліки, які потребують обов'язкового виправлення, бо це дуже погано впливає на рендеринг.

Google Page Speed Insights використовує більш ніж 200 факторів для того, щоб сформувати рейтинг, і після отримання загальної оцінки можна ознайомитися, що більш за все вплинуло на те, що рейтинг не є ідеальним. Вагомість помилок поділена також на три групи:

- Red -- сильно зповільнює роботу веб-додатку, потрібно виправити якомога швидше
- Yellow -- не критично, але при можливості краще виправити.
- Green -- повністю задовільняє всім вимогам.

Найголовніше, на що треба звернути увагу -- “Можливості” після Google Page Speed тесту. Можливості -- це пропозиції як зробити, щоб сторінка завантажувалася швидше.

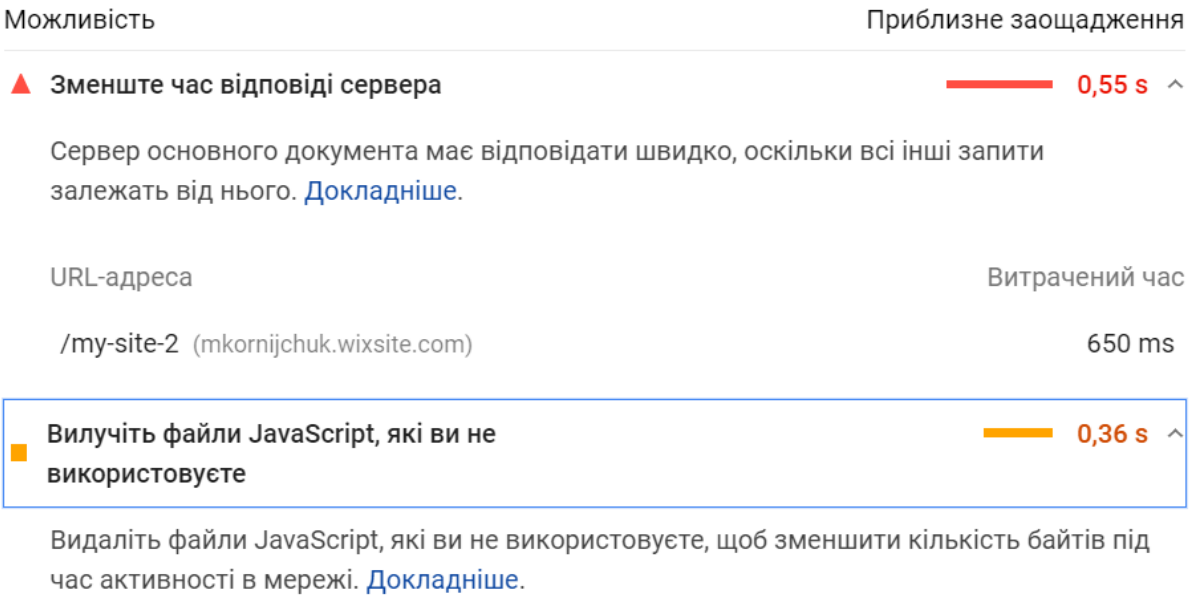


Рис 5. 3 Можливості для покращення контенту в звіті від PageSpeed

Також варту звернути увагу на пункт “Діагностика”. Діагностика дає більш детальну інформацію по кожному аспекту, який можна покращити.

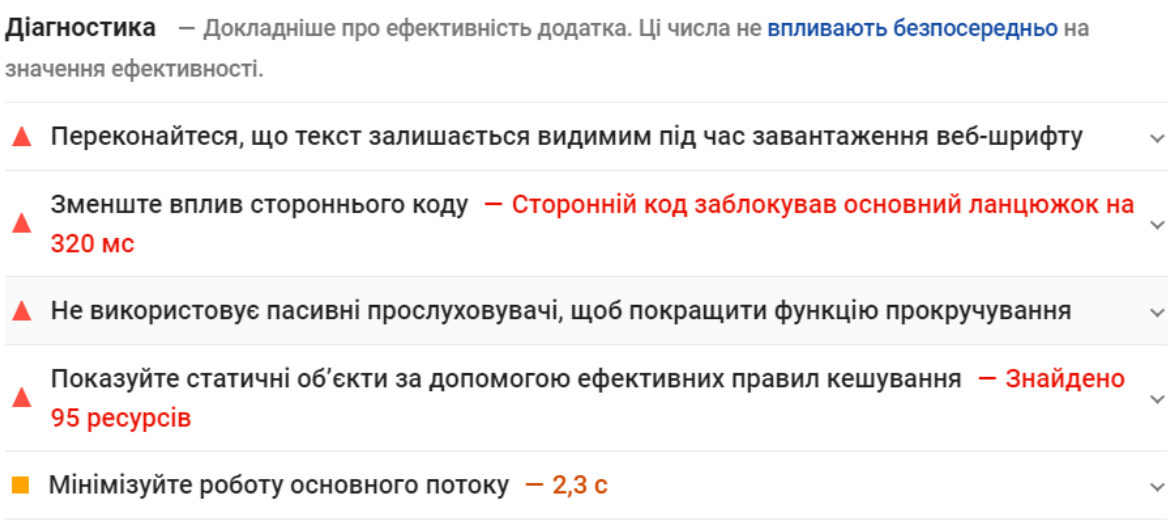


Рис 5. 4 Діагностика від PageSpeed

По кожній окремій проблемі можна ознайомитися детальніше та на цій основі зробити висновки з приводу покращення.

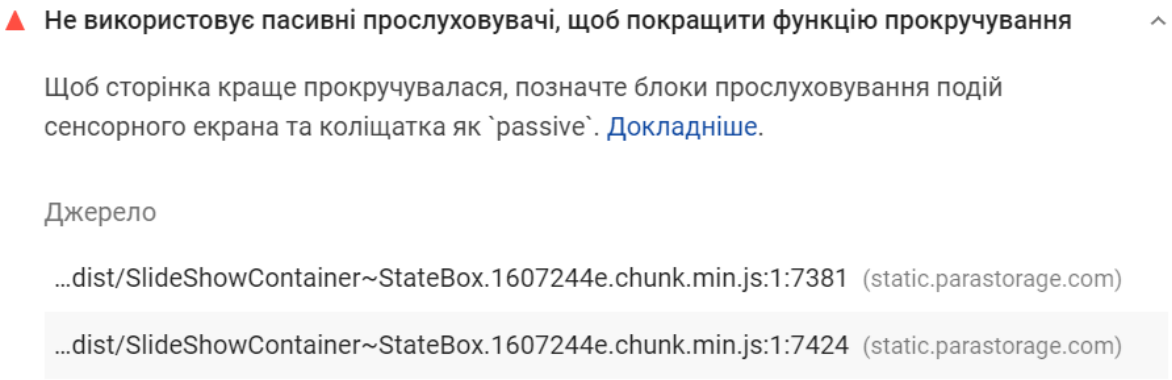


Рис 5. 5 Один з деталізованних методів у діагностиці

Загалом Google Page Speed Insights це швидкий, простий у використанні інструмент з детальним аналізом продуктивності сайту і практичними рекомендаціями щодо підвищення швидкості завантаження. Використовуючи цей сервіс, ви зможете поліпшити конверсію свого веб-ресурсу і уникнути ситуації з відходом користувача через затяжний рендеринг або тривале підвантаження зображень.

Другий дуже потужний інструмент для тестування, який хотілося б розглянути -- WebSite Auditor.

Для оптимізації сайту обов'язково потрібен аудит. Якщо проводити аудит сайтів вручну, то на це піде багато часу. Набагато ефективніше

використовувати спеціалізовані програми для аудиту, які економлять час і надають інформацію в зручній формі.

WebSite Auditor - це інструмент, який дозволяє ефективно проаналізувати весь ваш сайт, виправити помилки та перетворити його дружній для пошукової системи веб-додаток. Цей інструмент працює тільки як десктопний застосунок, завантажити його можна з офіційної сторінки https://www.link-assistant.com/website-auditor/.

За допомогою цього софта є можливим:

- Знайти технічні проблеми і недоліки сайту, а саме: дублювання метатегів або їх вітсутність, дублювання сторінок тощо. Вся ця інформація подається у вигляді докладного звіту;
- Побачити одразу рекомендації, як саме можна виправити недоліки, які було виявлено;
- Порівняти оптимізацію сторінок з сторінками провідних сайтів, які є конкурентними у цій області і на основі цієї інформації налагодити якісь процеси та поліпшити їх;
- Отримати статистику соціальних сигналів кожної сторінки сайту та порівнювати їх зі статистикою від Google Analytics;

Всі можливості охопити достатньо важко, тому що функціонал Auditora дійсно вражає і він буде корисним як і для початківцям у сфері SEO, так само і для досвідчених спеціалістів.

Перша можливість, яку надає це, звичайно, повний аудит сайту.

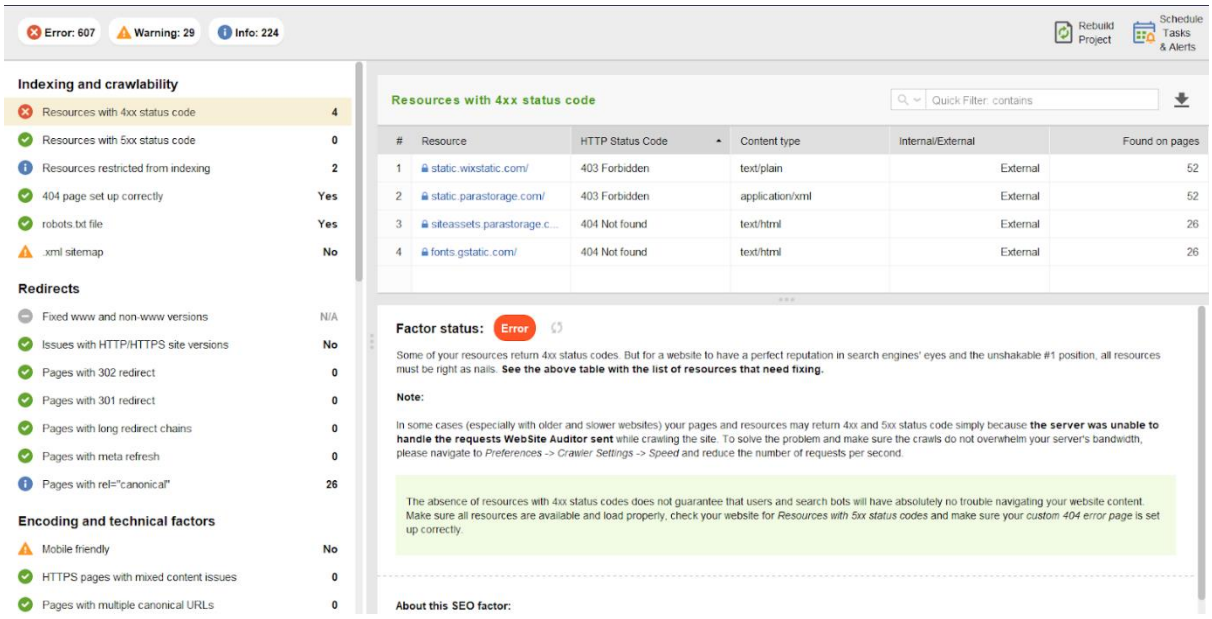


Рис 5. 6 Загальний звіт від WebSite Auditor

Інформація розподілена на три категорії: помилки, попередження та інформаційні.

З кожної з категорією можна ознайомитися окремо, почитати чому вона виникла, на що впливає, та як можна її виправити.

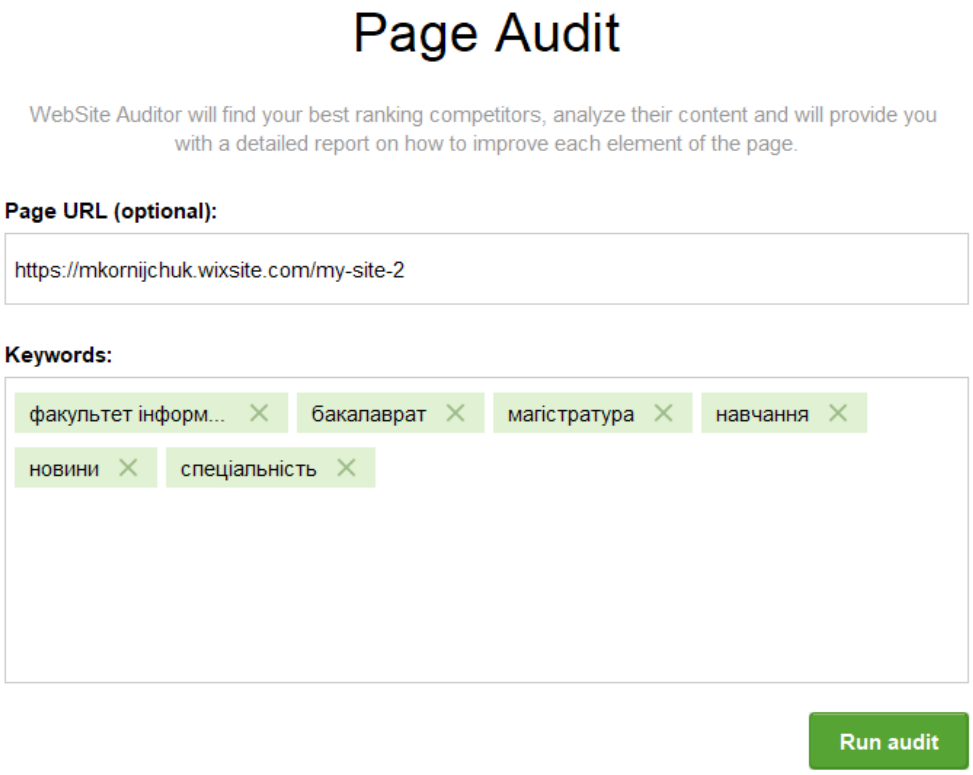


Рис 5. 7 Приклад використання функції Page Audit

Друга неймовірно важлива інформація може бути здобута за допомогою аудита певної сторінки. Прикладом буде слугувати початкова сторінка сайту

факультету інформатики. Ключові слова, як зазначалося раніше, є дуже важливими про оптимізація сайту, Page Audit дає детальну інформацію чи правильним є кількість ключових слів на сторінці та взагалі чи присутні вони.

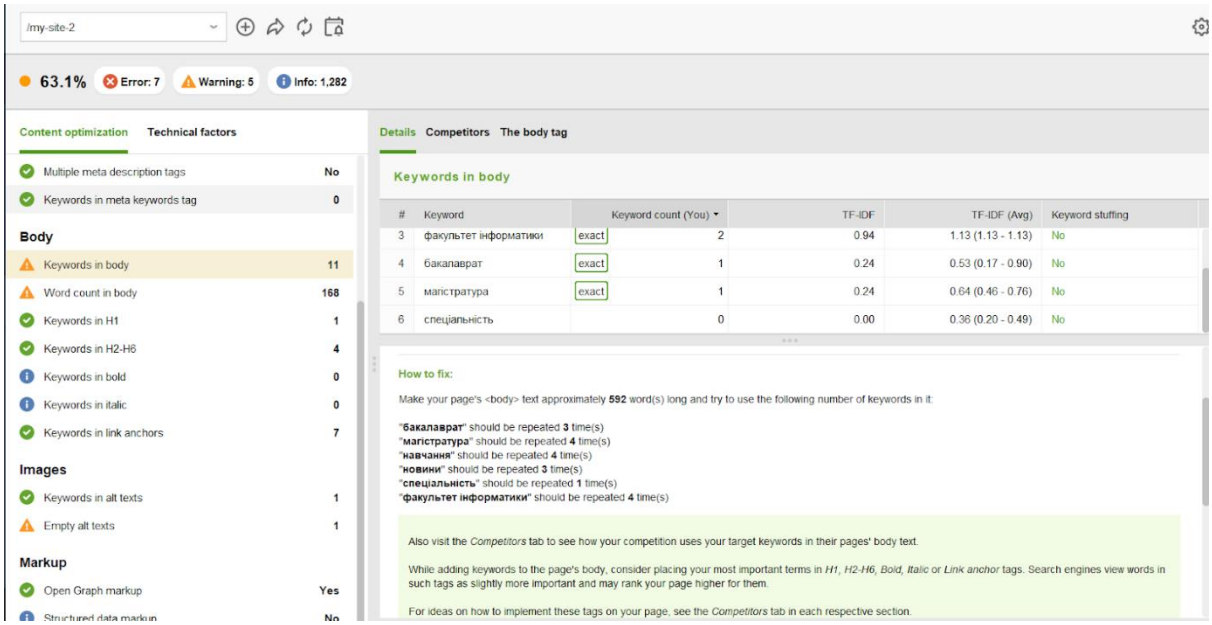


Рис 5. 8 Деталізованний звіт Page Audit

Звідси ми можемо побачити, що в нас, наприклад, немає ключового слова “спеціальність” на головній сторінці нашого сайту.

Третім інструментом, який, на мій погляд може відкрити очі на те, що здавалося б просто на поверхні -- це візуалізація проекту та зв’язки між категоріями та сторінками сайту.

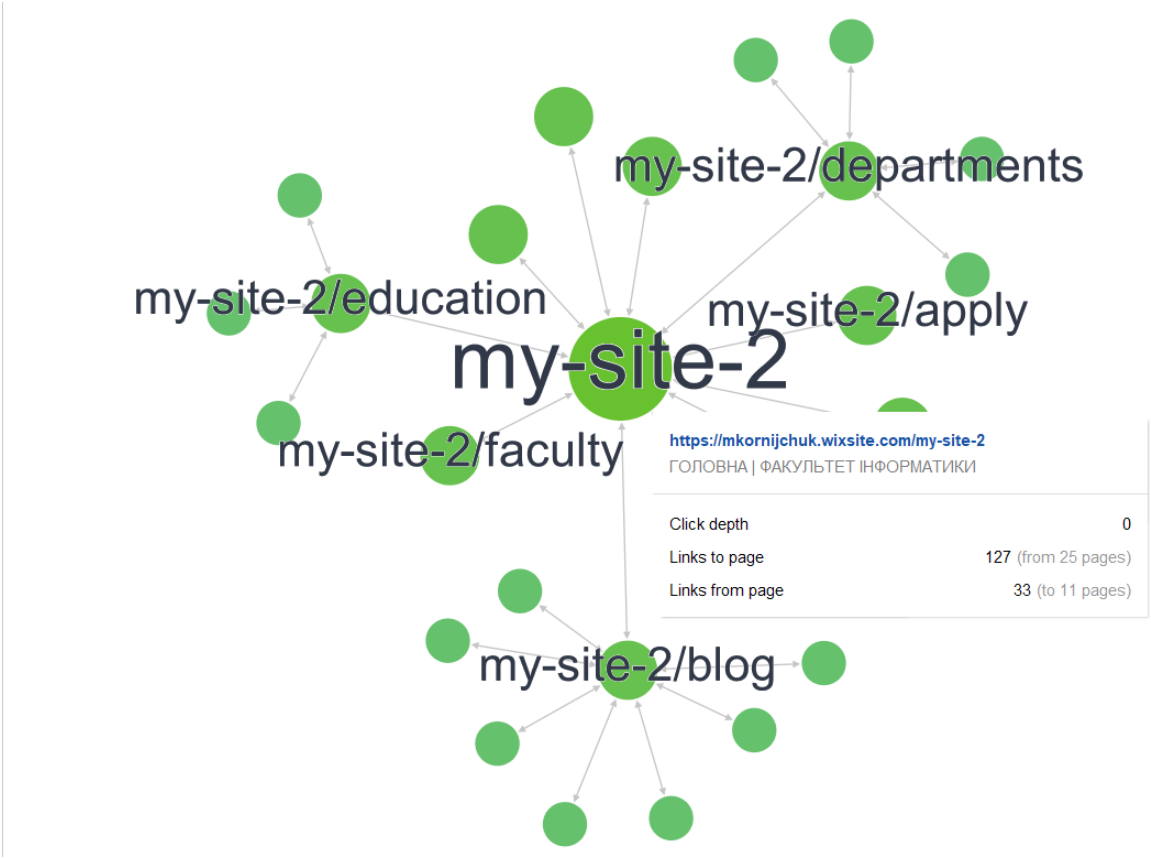


Рис 5. 9 Приклад візуалізації сайту

Часто трапляється, що з якоїсь сторінки, просто немає лінку на попередню, або про якусь сторінку зовсім забули і на неї ніяким чином неможливо потрапити. За допомогою візуалізкації це можна швидко побачити та усунути, а також побачити значимість кожної з сторінок і чи добре вони лінкуються між собою.

Також, хотілося б зазначити такі сервіси: SerpStat та Ahrefs.

Serpstat та Ahrefs - це сервіси веб-аналітики з багатим переліком інструментів для аналізу пошукової видачі, глибокого аналізу сайтів конкурентів по безлічі факторів ранжування, і масивної оптимізації свого ресурсу. В багатьох нюансах доволі схожі за функціоналом на WebSite Auditor, тому детально на ньому зупинятися немає сенсу. Можливості SerpStat і Ahrefs:

- аналіз сайту;
- аналіз домену;
- аналіз ключових запитів;
- парсер підказок;

- контент-маркетинг тощо;

З цього можна зробити висновок, що це два доволі потужних інструменти в руках SEO-спеціалістів, але є доволі дорогими у використанні.

РОЗДІЛ 6. РЕЛІЗ

Реліз -- це випуск остаточної версії програм - готового для використання продукту. У релізі зазвичай збираються всі версії та оновлення, а також випуск кінцевого продукту з усіма виправленнями.

Розгалуження(branching) - це метод, який використовується для ефективної реалізації стратегії релізу. Великі проекти вимагають заповнення багатьох ролей, включаючи розробників, тестувальників та менеджерів збірки(build). Крім того, можливо, доведеться підготувати різні релізи на різних платформах. З такою стратегією можливо досягнути підтримки кода у доступному стані, коли це потрібно.

Гілки дозволяють паралельно розробляти різні функції, відокремлюючи зміни без дестабілізації кодової бази, наприклад, нові функції, виправлення помилок та інтеграцію версій. Пізніше, коли всі тести будуть пройден, зміни можуть бути об'єднані.

6.1. Стратегія розгалуження на базі основної гілки

Модель розгалуження на основі основної гілки спрямована на підтримку єдиної гілки розвитку в здоровому стані. Це мінімізує кількість інших використовуваних гілок, оскільки всі розробники прив'язуються до однієї спільної гілки.

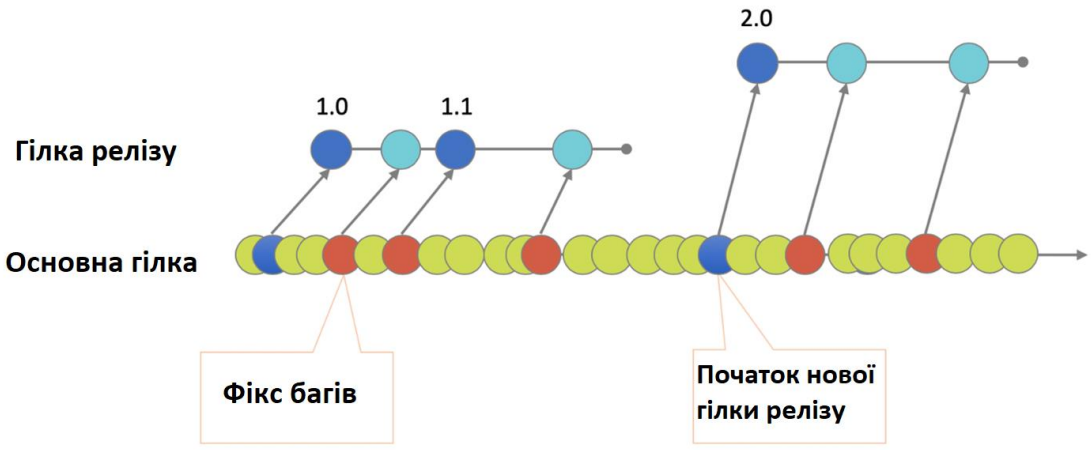


Рис. 6. 1 Стратегія розгалуження на базі основної гілки

Гілка створена для релізу, але лише інженери релізу коммітять у основну гілку. Вони також можуть вибирати окремі коміти з основної лінії до гілки випуску, якщо є бажання це зробити, наприклад у випадку виправлення помилок. Ця модель розгалуження вимагає великої дисципліни, оскільки помилковий коміт майже миттєво впливає на інших розробників. Для підтримання основної гілки у постійно здоровому стані часто використовуються автоматичні перевірки перед коммітом та огляд коду.

6.2. Модель GitFlow

Система контролю версій від Git змінила уявлення розробників про злиття та розгалуження. Злиття завжди вважалося дещо страшним, але з Git ці дії стали дешевими та швидкими.

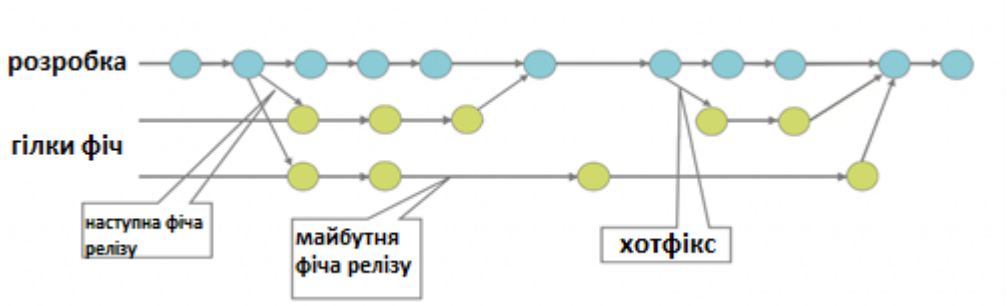


Рис. 6. 2 Модель GitFlow

Центральний репозиторій містить гілку розробки, де вихідний код відображає стан із останніми внесеними змінами розробки для наступного релізу. Гілки фіч використовуються для розробки нових функцій для майбутнього або віддаленого майбутнього випуску. Суть гілки фіч полягає в тому, що вона існує до тих пір, поки вона знаходиться в стадії розробки, але врешті-решт буде об’єднана назад у розробку або відкинута.

6.3. Реліз в моделі GitFlow

Гілки релізу створюються з гілки розробки. Ці гілки можуть існувати деякий час, доки випуск не буде розгорнуто однозначно. За цей час у цій гілці може бути фікс багів. Додавати тут великі нові функції категорично заборонено. Їх потрібно об’єднати у гілку розробки, а отже, почекати

наступного великого релізу. Основна гілка - це гілка, де вихідний код завжди відображає стан готовності до релізу.

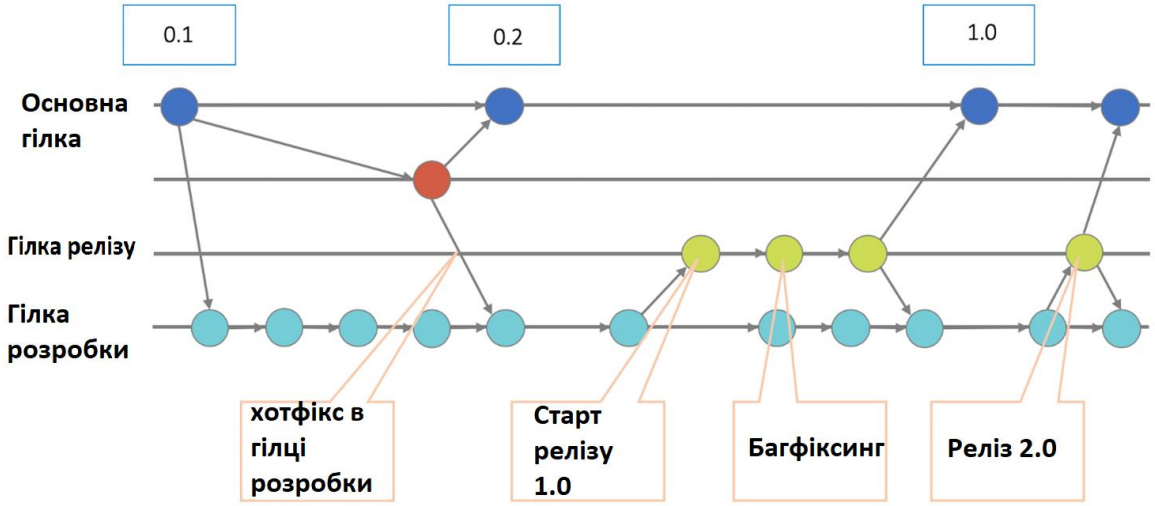


Рис. 6. 3 Реліз в моделі GitFlow

Коли стан гілки релізу готовий стати справжнім релізу, він об'єднується в основну гілку. Далі, що коміт на основну гілку повинен бути позначений для зручного подальшого посилання на цю історичну версію. Нарешті, зміни, внесені у гілку релізу, потрібно об’єднати назад у розробку, щоб майбутні випуски також містили ці виправлення багів. Гілки багів дуже схожі на гілки релізу, оскільки вони також призначені для підготовки до нового релізу, але це не є запланованою зміною. Вони виникають внаслідок необхідності негайно діяти на небажаний стан основної версії. Суть полягає в тому, що робота членів команди (над гілкою розробки) може продовжуватися, тоді як інша людина готує швидкий фікс основної гілки.

6.4. Реліз в моделі GitHub Flow

У відповідь на Git Flow була деталізована більш проста альтернатива, яка називається GitHub flow. Цей потік має лише функціональні гілки та головну гілку. Усе, що є у гілці master, можна розгорнути, тому для роботи над чимось новим створіть гілку з описовим іменем. Коли він буде готовий, ви створюєте запит на злиття або пул реквест(запит на впровадження змін із

вашої гілки в основну гілку вихідного репозиторію). Після того, як якась нова функція отримала затвердження, то вани зливається у основну гілку.

РОЗДІЛ 7. МОНІТОРИНГ

Моніторинг - це автоматична система відслідковування проекту, яка дозволяє контролювати параметри проекту в заданих рамках, оперативно усувати і запобігати збої в їхній роботі.

Моніторинг можна розділити на два типи. Перший -- моніторинг коректної роботи продукту, щоб він був доступний 24/7. Другие -- моніторинг користувацької активності, як клієнти взаємодіють з контентом, на що вони більш за все звертають увагу тощо.

UptimeRobot -- це інструмент, який дозволяє моніторити сайт кожні 5 хвилин і повідомить про недоступність, якщо така станеться. Повідомлення можна налаштувати на пошту або СМС. Також ресурс дає можливість бачити графік скільки часу витрачається на відповідь від сервера.

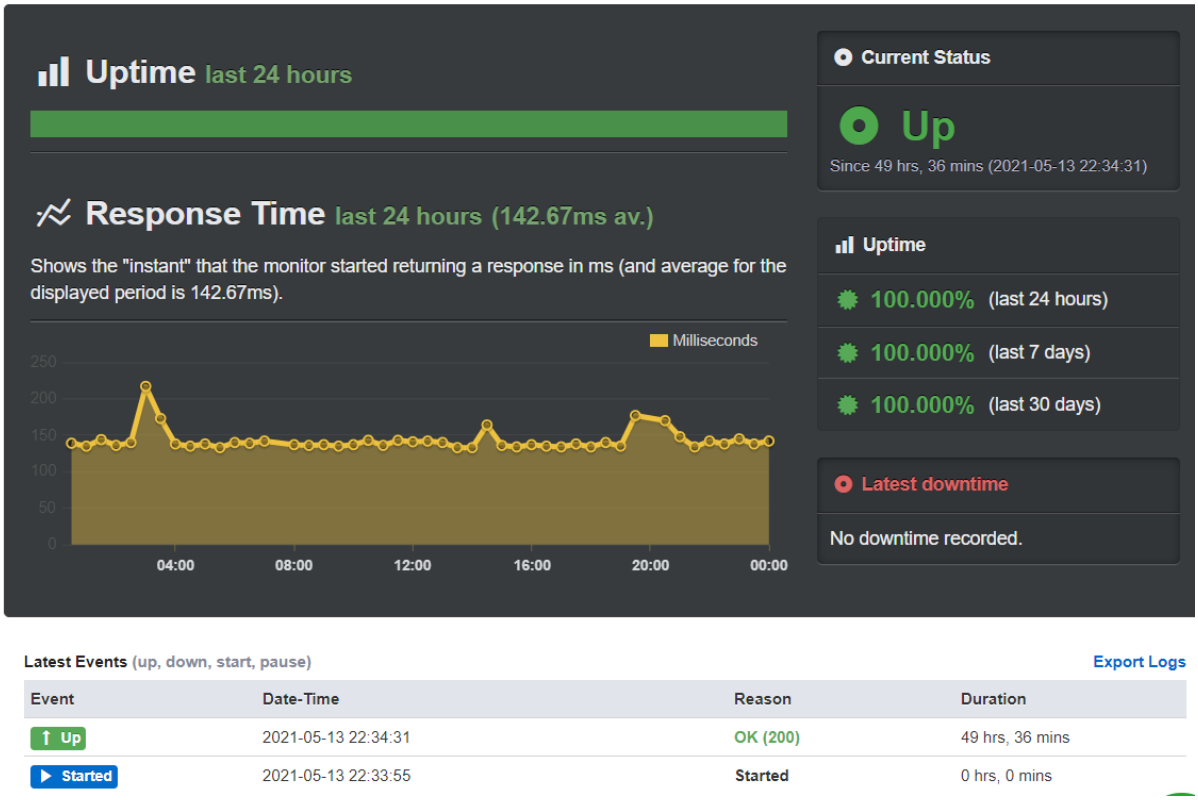


Рис. 7. 1 Приклад роботи UptimeRobot

За допомогою графіку можна відстежити коли саме відповідь від серверу йшла довше.

Google Analytics і Google Search Console це інструменти саме для відслідковування користувацької діяльності і їх взаємодії з контентом.

Функціонал Google Analytics робить його одним з лідером на ринку статистичних звітів. Головні можливості, які надає Google Analytics :

Можливість відстежувати відвідуваність за певним джерелом або за пошуковую фразою;

Відстеження подій на сторінці, наприклад: натискання на flash-елемент або завантаження файлу на комп'ютер;

Індивідуальні звіти, які налаштовуються для отримання швидкого доступу до необхідної статистики;

Статистика сторінок, за допомогою якої можна переглядати, як відвідувачі переходять зі сторінки на сторінку, і відсоток переходів;

Створення персоналізованих сегментів з метою виділення груп відвідувачів і аналізу їх взаємодії з контентом сайту, та багато інших;

За допомогою Google Search Console можна зрозуміти як сайт представлений у пошуковій системі Google, способи оптимізації контенту та, якщо помилки будуть з’являтися -- то рішення з усунення їх.

Основні дії, які можна виконати за допомогою Search Console:

- Дізнатися, які сайти ссилаються на наш ресурс;
- Діагностувати проблеми с індексуванням і надіслати запит на нове індексування при умові створення нового контенту;
- Оцінити трафік, який надходить з пошуку Google;
- Отримувати сповіщення про спам та інші несправності і багато іншого;

В купі ці два інструмента допоможуть правильно зрозуміти як сайт представлений у пошуковій системі Google, аналізувати користувацьку поведінку і що допоможе зробити висновку з приводу ефективності контенту та способів його покращення.

На основі призначеного для користувача досвіду формуються запити на новий функціонал для продукту, готується план доробок. Після цього цикл замикається і переходить в початкову стадію - планування. Далі починається нова ітерація CCD розробки.

ВИСНОВКИ

В результаті дослідження було виявлено проблему з правильним підходом до роботи з інформаційним контентом. Проблемою занепаду програмного продукту є зупинка в розвитку контенту та його подальшому оновленні. В більшості випадках, це може траплятися через відсутність правильної роботи з інформаційним контентом, чим досить часто нехтують.

Було висунуто гіпотезу, що методологія Continuous Content Development здатна вирішити цю проблему. Було проведено роботу над: створенням прототипу сайту, протестована система актуалізації контенту. В результаті цього, гіпотезу було підтверджено, методологія є робочою, але потребує подальших досліджень та деталізації кроків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. RACI Matrix Template [Електронний ресурс] – Режим доступу до ресурсу: https://www.vertex42.com/ExcelTemplates/raci-matrix.html.
2. Asimos T. A Content-First Approach to Redesigning Your Firm's Website [Електронний ресурс] / Tim Asimos. – 2020. – Режим доступу до ресурсу: https://www.circlesstudio.com/blog/content-first-approach-redesigning-website/.
3. Vincent D. Удачная модель ветвления для Git [Електронний ресурс] / Driessen Vincent – Режим доступу до ресурсу: https://habr.com/ru/post/106912/.
4. What is CI/CD [Електронний ресурс] – Режим доступу до ресурсу: https://www.redhat.com/en/topics/devops/what-is-ci-cd.
5. WebSite Auditor and Link Assistant [Електронний ресурс] – Режим доступу до ресурсу: https://seo-hacker.com/seo-powersuite-review-part-2-website-auditor-linkassistant/.
6. How to Do a Basic Website Audit to Improve SEO and UX (in 10 Steps) [Електронний ресурс] – Режим доступу до ресурсу: https://ahrefs.com/blog/website-audit/.

ДОДАТКИ

ДОДАТОК 1. Основна сторінка сайту факультету інформатики