

5 ЛАБОРАТОРНА РОБОТА №5 ТЕХНІЧНЕ ТЕСТУВАННЯ

Мета роботи: отримання базових теоретичних та практичних навичок з технічного тестування, ознайомлення з функціональними можливостями Інструментів розробника та *Xenu Link Sleuth*.

5.1 Короткі теоретичні відомості

Технічне тестування – група видів тестування, що проводиться на завершальному етапі готовності програмного продукту (сайту) й спрямована на перевірку технічних характеристик, таких як:

- здатність сервера витримувати навантаження;
- швидкість завантаження сторінок;
- трастовість (такі показники як вік, PR);
- цілісність посилань;
- індексація сторінок, наявність в каталогах пошукових машин;
- мета-теги, HTML-теги;
- відповідність стандартам W3C і багато іншого.

Основні програми для технічного тестування:

1. *Інструменти розробника (Developer Tools)* – це потужний інструмент для налагодження коду вебсайтів, який за замовчуванням встановлений в браузерах Firefox, Google Chrome та в інших браузерах. Крім веброзробки, даний інструментарій, а також накопичені знання з HTML, CSS будуть дуже корисні тестувальникам при тестуванні вебсайтів і створенні баг-репортів. Визначити колір, погратися зі шрифтами, виміряти піксельну позицію елемента на сторінці або протестувати адаптивність сайту для різних пристроїв – тепер все це можна зробити не встановлюючи окремих додатків.

До основних можливостей Інструментів розробника відносять:

- зручний перегляд HTML-коду сторінки: функція Inspect дозволяє точно визначити місцезнаходження тега того чи іншого елемента, переглянути всі «прив'язані» до нього властивості та стилі;
- редагування HTML та CSS безпосередньо в браузері: можна змінювати атрибути тегів та значення властивостей для того, щоб

спостерігати зміни. Це зручно для тих випадків, коли потрібно шляхом експериментів знайти найбільш прийнятний варіант оформлення створеної сторінки;

- налагодження JavaScript;
- відстеження процесу завантаження сторінки. Існує можливість урізати швидкість з'єднання, щоб подивитися як буде поводити себе контент сторінки при повільному інтернеті або коли взагалі не буде інтернету;

- перегляд HTTP-заголовків звичайних та AJAX-запитів;
- редагування або видалення cookie. Вкладка Sources;
- зміна даних геолокації – це досить зручно при тестуванні різних сервісів з урахуванням карт і маршрутів. У цьому режимі можна й провести тестування юзабіліті мобільної версії сайту. А підтримка емуляції відразу пари десятків пристроїв на адаптивність, зробить перевірку ще більш надійною.

2. *Xenu Link Sleuth* – це один з найбільш корисних інструментів в пошуковій оптимізації.

До основних завдань, які виконує програма відносять:

- шукати биті (непрацюючі) посилання на заданому ресурсі: своєчасне виявлення несправних посилань дозволяє не тільки поліпшити показники сайту в пошукових системах, але й вочевидь підвищити інтерес та лояльність користувачів сайту;

- складати карту сайту: для динамічних сайтів скласти карту не складає проблеми, однак, для статичних HTML ресурсів створювати карту сайту вручну вельми довго й трудомістко. *Xenu* вирішує цю задачу за кілька хвилин в залежності від розміру сайту та швидкості інтернетз'єднання;

- шукати сторінки з великим часом віддачі: вимірювання швидкості завантаження вебсторінок – дуже важливий процес тестування. Але заміряти вручну кожен сторінку багатосторінкового сайту занадто затратно по ресурсам. За допомогою *Xenu* можна побачити картину відгуку всіх сторінок в цілому та визначити проблемні місця;

- знаходження неунікальних заголовків: кожен заголовок на сторінці повинен бути унікальним, тоді жоден з них не буде знаходитися в додаткових результатах пошуку та фільтруватися, як дубльований контент.

- знаходження сторінок з великим рівнем вкладеності: всі сторінки на сайті по можливості повинні знаходитися не далі, ніж у двохтрьох рівнях від головної. Чим далі знаходиться сторінка, тим складніше до неї добратися як користувачам, так і пошуковим системам. Якщо знайшлися подібні сторінки, які являються достатньо важливими, але знаходяться далі, ніж в 3-х рівнях від головної, варто прийняти які-небудь заходи для поліпшення навігації;
- виявляти, які зі сторінок мають найбільшу та найменшу кількість внутрішніх посилань: перевірити внутрішню перелінковку в числовому вигляді. Які з сторінок заслужили більше уваги, а які менше (виходячи з внутрішніх посилань);
- знаходження картинок з відсутнім атрибутом «alt»: атрибут «alt» є важливим при оптимізації сайту або окремих сторінок під певні запити. Перевірити, можливо відсутня важлива інформація чи опис зображень на вашому сайті.

3 Тестування навантаження (Load Testing) – дозволяє зрозуміти чи здатен сайт/додаток стабільно працювати з навантаженням в допустимих межах і трохи перевищуючи ці межі. При цьому тестуванні перевіряється поведінка з високим навантаженням. Навантаженням може бути, наприклад, певна кількість одночасно працюючих користувачів додатка в певний проміжок часу. Такий тип тестування дозволяє визначити час відгуку важливих бізнес-транзакцій. Спостерігаючи за базою даних, сервером додатка і мережею можна визначити слабкі місця програми.

Навантажувальне тестування найчастіше застосовується для дослідження багатокористувацьких ресурсів і різноманітних загальних систем, але інші види програмних продуктів також можуть бути протестовані цим методом. Наприклад, можна перевірити, чи зможе графічний редактор впоратися з відкриттям зображення великого обсягу. Також можна з'ясувати, чи здатна система сформувати фінансовий звіт, проаналізувавши дані декількох місяців, або навіть років. Якісно створений навантажувальний тест забезпечить найбільш достовірними результатами.

Основною ідеєю навантажувального тестування є створення певного навантаження за допомогою однакових апаратних і програмних забезпечень, щоб відстежити індекс продуктивності продукту. Найефективнішим цей метод буде на ранніх етапах

розробки, тому що такий підхід дозволить отримати оптимальні результати вимірювання показників продуктивності системи.

Основні принципи навантажувального тестування:

1. Унікальність запитів. Під час складання сценарію слід враховувати реальні статистичні дані, вимоги, очікувану поведінку системи.

2. Час відгуку системи. Керуючи певним собі числом вимірів, можна визначити до якого інтервалу часу потрапить той чи інший запит.

3. Розподіл системи залежить від часу відгуку. Кількість вузлів впливає на розкид часу відгуку системи, кожен з яких збільшує величину затримки при скануванні запитів. Цей факт варто врахувати під час складання вимог до продуктивності продукту.

4. Коректність відтворення навантажувальних профілів. Складність програмного забезпечення вимагає значних витрат часу і ресурсів на проектування, програмування і подальшу підтримку. Розробка тестів і покриття функціоналу системи повинні бути збалансованими для отримання найбільш

Системи управління контентом (CMS – Content Management System)

CMS – інформаційна система або комп'ютерна програма, яка використовується для забезпечення та організації спільного процесу створення, редагування та управління контентом.

Основні функції CMS:

- надання інструментів для створення наповнення (даних), організації спільної роботи над наповненням;
- управління даними: зберігання, контроль версій, дотримання режиму доступу, управління потоком документів та ін.;
- публікація наповнення;
- представлення інформації у вигляді, зручному для навігації, пошуку.

Joomla! – система управління даними (CMS), написана на мовах PHP і JavaScript, що використовується в якості сховища бази даних СУБД MySQL або іншої індустріально-стандартної реляційної СУБД. Є вільним програмним забезпеченням, що поширюється під ліцензією GNU GPL.

CMS Joomla! включає в себе мінімальний набір інструментів при початковій установці, який доповнюється в міру необхідності. Це знижує заповнення адміністративної панелі непотрібними елементами, а також знижує навантаження на сервер і економить місце на хостингу.

Drupal – система управління вмістом, що також використовується як каркас для вебдодатків (CMF). Система написана на мові PHP та використовує як сховище даних реляційну базу даних (підтримуються MySQL, PostgreSQL та інші). Drupal є вільним програмним забезпеченням,

5.2. Завдання до роботи

5.2.1. Дослідити 3 зображення на будь-яких вебсайтах, визначити розмір цих зображень за допомогою Інструментів розробника. У звіті описати проведену роботу та додати скріншот з посиланнями на сторінки цих сайтів, вказати розміри зображень (рисунок 5.1).

5.2.2. Описати та додати до звіту 2 скріншота з помилкою в консолі Web Developer при перегляді різних сайтів (рисунок 5.2).

5.2.3. Змінити властивості шрифту на сторінці будь-якого сайту (колір та розмір), ширину поля. У звіті описати проведену роботу та додати до звіту скріншот з посиланнями на сторінки (рисунки 5.3, 5.4).

5.2.4. Встановити програму *Xenu Link Sleuth* для пошукової оптимізації.

Провести технічне тестування пошукової оптимізації будь-якого сайту за допомогою *Xenu Link Sleuth*. Для чого обрати пункт меню File → Check URL. У вікні ввести адресу сайту та виділити чекбокс «Check external links» (рисунок 5.5).

У програмі передбачено багато налаштувань, доступних в пункті меню Options → Preferences (рисунок 5.6).

Функції, які виконують відповідні елементи пункту меню «Options»:

1. «Parallel threads» встановлює кількість синхронних (паралельних) запитів;

2. «Apply to all jobs» дозволяє зберігати налаштування для всіх проєктів та звіт дозволяє включати наступні пункти:

- «Broken links, ordered by links» – биті посилання, відсортовані за адресою;
 - «Broken links, ordered by page» – список битих посилань, відсортованих по сторінках;
 - «Broken local links» – биті внутрішні посилання;
 - «Redirected URLs» – адреси з 301/302 перенаправленням;
 - «Ftp and gopher URLs» – ФТП й інші не HTTP адреси;
 - «Valid text URLs» – список робочих текстових посилань;
 - «Site Map» – карта сайту;
3. «Statistics» – статистика.

5.2.5 Після закінчення перевірки сайту за допомогою *Xenu Link Sleuth*, в головному вікні виводиться детальна таблиця, в якій можна відсортувати по кожному із стовпців: адреса посилання, статус, міме-тип, розмір, заголовок, рівень вкладеності, кількість зовнішніх та внутрішніх посилань, час віддачі сторінки.

Файл звіту роботи *Xenu* дуже великий й містить всю необхідну інформацію (рисунок 5.7).

Описати проведену роботу та додати скріншот звіту *Xenu* (рисунок 5.7).

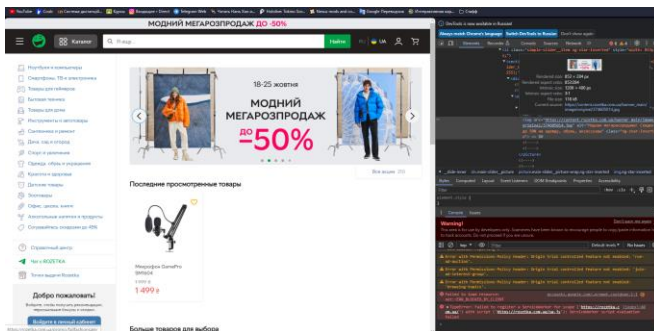


Рисунок 5.1 – Розмір картинки: 850 x 402 рх. Посилання: <https://prom.ua>

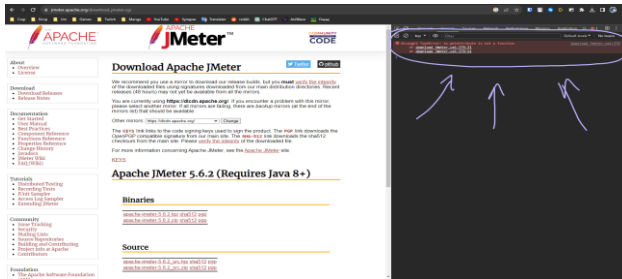


Рисунок 5.2 – Помилки на сайті
https://jmeter.apache.org/download_jmeter.cgi

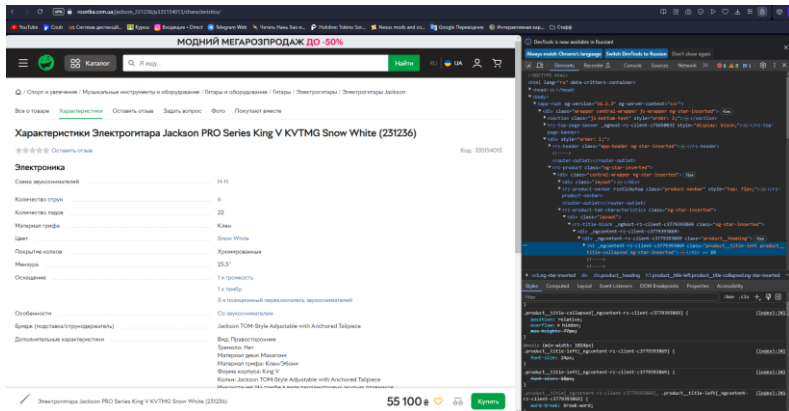


Рисунок 5.3 – Зображення до зміни параметрів:
https://jmeter.apache.org/download_jmeter.cgi

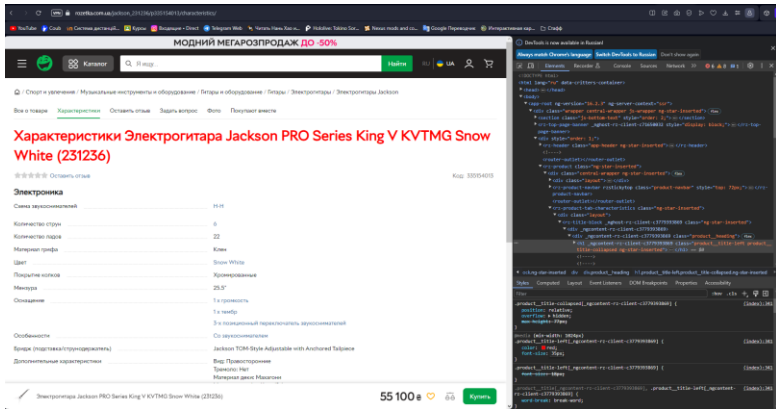


Рисунок 5.4 – Зображення після зміни параметрів:
https://jmeter.apache.org/download_jmeter.cgi

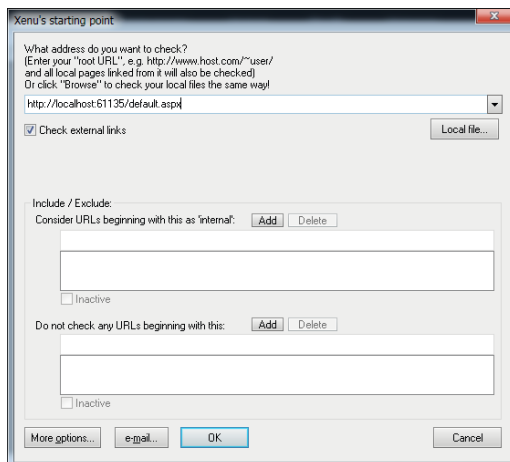


Рисунок 5.5 – Стартове вікно програми *Xenu Link Sleuth*

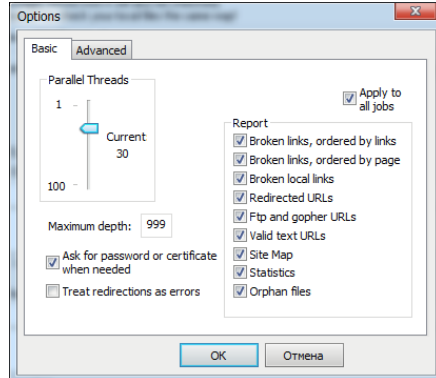


Рисунок 5.6 – Вікно налаштувань програми *Xenu Link Sleuth*

Address	Status	Type	Size	Title	Date	Level	Out L...	In L...	Server	E...	Duration
http://localhost:61135/media/1283250318...	not found			backstage		3		1	ASP.NET Develo...		00:00.186
http://localhost:61135/Ombre.L2.jpg	not found			backstage		3		1	ASP.NET Develo...		00:00.456
http://localhost:61135/media/7/Emplu.L.23...	not found			aspasovsk		3		1	ASP.NET Develo...		00:00.456
http://localhost:61135/About.aspx	ok	text/html	13386	About Us		1	28	22	ASP.NET Develo...		00:01.327
http://localhost:61135/Account/Login.aspx	ok	text/html	7095	Log In		1	24	22	ASP.NET Develo...		00:01.079
http://localhost:61135/Account/Login.aspx	ok	text/html	7154	Log In		2	25	1	ASP.NET Develo...		00:02.316
http://localhost:61135/Account/Record.aspx	ok	text/html	171	Заявка к мастеру		1	1	23	ASP.NET Develo...		00:02.923
http://localhost:61135/Account/Register....	ok	text/html	9404	Register		2	24	1	ASP.NET Develo...		00:03.237
http://localhost:61135/Account/Register....	ok	text/html	9508	Register		3	24	1	ASP.NET Develo...		00:00.973
http://localhost:61135/Contacts.aspx	ok	text/html	6459	Контакты		1	24	22	ASP.NET Develo...		00:01.285
http://localhost:61135/default.aspx	ok	text/html	7651	Home Page		0	25		ASP.NET Develo...		00:00.409
http://localhost:61135/Default.aspx	ok	text/html	7651	Home Page		1	25	22	ASP.NET Develo...		00:01.185
http://localhost:61135/img/levask.jpg	ok	image/jpeg	86148	tt		3		1	ASP.NET Develo...		00:00.936
http://localhost:61135/img/1391579914...	ok	image/jpeg	63567	tt		3		1	ASP.NET Develo...		00:01.523
http://localhost:61135/img/102911-25617...	ok	image/jpeg	13257	no foto		2		1	ASP.NET Develo...		00:02.522
http://localhost:61135/img/1391579914...	ok	image/jpeg	19816	no foto		2		1	ASP.NET Develo...		00:02.485
http://localhost:61135/img/221.jpg	ok	image/jpeg	68794			2		1	ASP.NET Develo...		00:02.748
http://localhost:61135/img/221.jpg	ok	image/jpeg	58171	backstage		2		1	ASP.NET Develo...		00:02.618
http://localhost:61135/img/323.jpg	ok	image/jpeg	199395	no foto		2		1	ASP.NET Develo...		00:02.509
http://localhost:61135/img/432.jpg	ok	image/jpeg	80918	backstage		2		1	ASP.NET Develo...		00:02.455
http://localhost:61135/img/512.jpg	ok	image/jpeg	11165	no foto		2		1	ASP.NET Develo...		00:02.525
http://localhost:61135/img/art1 team gr.c...	ok	image/png	46151			1		2	ASP.NET Develo...		00:01.936
http://localhost:61135/img/le-v-massazh....	ok	image/jpeg	102833	tt		3		1	ASP.NET Develo...		00:00.844

Рисунок 5.7 – Вікно звітної інформації роботи *Xenu Link Sleuth*

5.3 Зміст звіту

5.3.1. Тема та мета роботи.

5.3.2. Завдання до роботи.

5.3.3. Короткі теоретичні відомості.

5.3.4. Хід роботи з виконання завдання до лабораторної роботи.

5.3.5. Висновки, що відображують результати виконання роботи та їх критичний аналіз.

5.3.6 Перелік використаних джерел.

5.4 Контрольні запитання

5.4.1. Яким чином можна оцінити проблемні частини сайту за допомогою Dev Tools? Відповідь обґрунтувати та навести приклади.

5.4.2. За допомогою якого програмного інтерфейсу можливо змінити «JavaScript» сценарії у реальному часі на сайті?

5.4.3. Як визначити за допомогою Dev Tools, коли почалося та закінчилося завантаження файлів на сайті? Відповідь обґрунтувати та навести приклади.

5.4.4. В якій послідовності завантажуються «JavaScript»? Відповідь обґрунтувати та навести приклади.

Перерахувати, які помилки на сайті можна знаходити за допомогою інструментів розробника, навести приклади.

5.4.5. Навести приклади інструментів для навантажувального тестування, коротко розповісти про них.

5.4.6. Розповісти для чого проводиться навантажувальне тестування?