

Тестування програмного забезпечення

Викладач: к.т.н. Шитікова Олена Вікторівна

Лекція 6

ТЕСТУВАННЯ
ВЕБПРОЄКТІВ

Тестування вебпроектів

Тестування, як завершальний етап розробки **вебсайту**, відіграє життєво важливу роль в процесі створення якісного програмного забезпечення.

Чим складніший сайт, тим більше часу потрібно на його перевірку та налагодження. Залежно від специфіки проєкту, на тестування може виділятися до 50% загального бюджету і часових ресурсів.



Тестування вебпроєктів

Тестування сайту вирішує кілька основних завдань:

- дає впевненість у якості продукту, підтверджує, що всі заявлені функціональні вимоги реалізовані, вебсайт їм відповідає і не має помилок у програмному кодї;
- підтверджує, що сайт здатний виконуватися у всіх заявлених режимах і на всіх підтримуваних ОС або браузерах коректно;
- гарантує, що дані, які зберігаються та оброблюються, надійно захищені від стороннього доступу;
- дозволяє переконатися в тому, що користувач може «інтуїтивно» використовувати продукт, не плутаючись в складному інтерфейсі.

Тестування вебпроєктів

Етапи тестування вебпроєктів

1. Вивчення документації (підготовчі роботи).
2. Тестування верстки.
3. Функціональне тестування.
4. Тестування зручності використання (юзабіліті).
5. Тестування безпеки.
6. Тестування продуктивності сайту

Функціональне тестування вебпроєктів

Тестування інтеграції даних для вебдодатків

Інтеграція даних в контексті перевірки вебдодатків відноситься до того, як введені на сайті дані сприймаються, перетворюються, відслідковуються (моніторяться), передаються по каналах передачі даних та зберігаються в базі даних в реальному часі.

Перевіряється на коректність введення максимально можливої величини текстових полів, від'ємні значення, дійсні числа тощо.



Функціональне тестування вебпроєктів

Тестування полів даних для вебдодатків

При перевірці полів даних в вебдодатках необхідно переконатися, що дані, які вводяться, відповідають необхідному для цього поля типу даних, і дані в поле можна вводити тільки певної довжини.

Перевіряється відсутність орфографічних помилок у назвах полів, коректний запис інформації про авторське право, правильність введення дат високосних років та числа 28-29 лютого, а також 30-31.



Функціональне тестування вебпроєктів

Тестування числових полів для вебдодатків

Перевірка числових полів виконується для перевірки того, що:

- найбільше та найменше значення оброблені коректно;
- порожні числові поля в додатку повинні оброблятися у вигляді появи підказки або повідомлення про помилку;
- додатні (+) та від'ємні (-) числові значення обробляються коректно;
- верхні та нижні значення обробляються коректно;
- забезпечується перевірка при діленні на нуль (0).

Функціональне тестування вебпроєктів

Тестування буквено-цифрових полів

а) Перевірка буквено-цифрових полів виконується для перевірки того, як обробляються:

- дані з пропуском та без пропуску (пробіл);
- найбільше та найменше значення;
- валідні букви для поля вебформи;
- невалідні букви для поля вебформи;
- невалідні символи для поля вебформи;
- дані з пропуском в першій позиції;
- дані з пропуском в останній позиції;

Функціональне тестування вебпроєктів

- мінімальна довжина допустимих символів для введення;
- максимальна довжина допустимих символів для введення;
- мінімальна кількість слів допустимих для введення;
- максимальна кількість слів допустимих для введення;
- слова з «чорного списку»;
- звичайний текст;
- регулярні вирази.

Функціональне тестування вебпроєктів

b) Тестування працездатності користувальницьких форм на сайті.

Форми призначаються для обміну даними між клієнтською частиною (браузером) і вебсервером або поштовим сервером.

Форма може містити інформацію у вигляді тексту, розмітку, а також спеціальні елементи, які називаються елементами управління

Google
Створити обліковий запис
Google
Введіть своє ім'я

Ім'я

Прізвище (необов'язково)

Далі

Українська Довідка Конфіденційність Умови

Google
Основна інформація
Введіть дату народження й стать

День Місяць Рік

Стать

Далі

Українська Довідка Конфіденційність Умови

Приклад форми реєстрації

Функціональне тестування вебпроєктів

1) Кнопки (*Buttons*).

Кнопки бувають різних типів:

- тип кнопки відправки (Submit) відповідає за відправку форми;
- активізація кнопки з типом скидання (Reset) призводить до скидання на формі всіх елементів управління в їх початкові значення;
- кнопки з типом натискання (Push) не мають поведінки за замовчуванням, і кожна така кнопка може мати свій клієнтський сценарій.

Функціональне тестування вебпроєктів

2) Прапорці – чекбокси (Checkboxes), можуть бути переключені користувачем у стан включити (On) / виключити (Off), і тільки включені прапорці беруться до уваги при відправці форми.

3) Перемикачі (Radio buttons) працюють, як і прапорці, тільки при виборі одного із значень в стан включити, інші елементи перемикаються в стан вимкнути.

Функціональне тестування вебпроектів

4) Поле введення тексту (Text input) призначене для введення тексту в однорядковому і багаторядковому режимах.

5) Поле/кнопка вибору файлу (File select) дає користувачеві можливість вибрати файл, який буде відправлений з формою, наприклад Ваше прикріплене резюме на сайті роботодавця.

6) Пароль (Password) – в цьому елементі вводиться текст таким чином, щоб символи приховувалися від «випадкових» спостерігачів, наприклад, за допомогою ряду зірочок *********.

Функціональне тестування вебпроєктів

При заповненні і відправленні форми необхідно, щоб на формі була обробка таких подій:

- прийняття допустимих вхідних значень;
- блокування неприпустимих вхідних значень;
- обов'язкові поля для введення повинні бути заповнені;
- необов'язкові поля у формі можуть залишатися незаповненими;
- перемикачі дозволяють вибрати тільки одне значення;
- прапорці / чекбокси можуть бути в станах включити / виключити;

Функціональне тестування вебпроєктів

- чи з'являється смуга прокрутки (Scrollbar) при необхідності?
- послідовне перемикання між елементами форми можливо за допомогою клавіші Tab на клавіатурі;
- чи всі дані в списках розташовані в хронологічному порядку?
- чи розташовані всі елементи форми відповідно до затвердженого дизайну?

Функціональне тестування вебпроєктів

с) Перевірка роботи пошуку (включаючи релевантність результатів).

Шукане значення:

- Аббревіатури
- Одне неповне слово / Кілька неповних слів
- Словосполучення
- Спецсимволи
- Цифри (позитивні, негативні, нуль, цілі, дробові тощо)
- Регістр введення
- Різні мови / розкладки
- Пусте значення
- Морфологія.

Функціональне тестування вебпроєктів

Додатково:

- Аналіз граничних значень і класи еквівалентності для довжини.
- Пошук по всіх полях.
- Пошук по деяких полях.
- Пошук без заповнення обов'язкових полів.
- Пошук по обов'язкових полях.
- Врахування різноманітних параметрів.

Функціональне тестування вебпроєктів

Варіанти введення:

- З клавіатури.
- З буфера обміну (через контекстне меню або поєднання гарячих клавіш).
- З екранної клавіатури.

Очищення поля:

- Через клавішу <Esc>.
- Через спец. клавішу додатки.
- Через клавіші <Delete> та <Backspace>

Функціональне тестування вебпроектів

d) Перевірки гіперпосилань, пошук неробочих посилань

Посилання (Links) або **гіперпосилання** (Hyperlinks) дозволяють в електронному документі активувати або відображати інший документ або програму.

Кожне посилання на сайті повинне бути в певному місці, його назва повинна відповідати вимогам і бути без граматичних помилок. Також, воно повинно правильно виконувати покладені на нього навігаційні властивості - вести на певну іншу сторінку або файл.

Перевіряються на сторінці також всі поштові посилання (Mail to links) і чи досягає відправлений з сайту електронний лист адресата.

Функціональне тестування вебпроєктів

Коли існує проблема при роботі з посиланнями, то браузер обробляє цю подію та показує повідомлення про помилку у вигляді коду стану НТТР (НТТР status code).

Коди відповіді сервера поділяються на

5 основних класів:

- 1xx Інформаційні коди (1xx Informational);
- 2xx Успішно (2xx Success);
- 3xx Перенаправлення (редирект) (3xx Redirection);
- 4xx Помилки клієнта (4xx Client Error);
- 5xx Помилки сервера (5xx Server Error).

Функціональне тестування вебпроєктів

е) Перевірки загрузки файлів на сервер

- Відкриття вікна провідника для вибору файлу для завантаження;
- Можливість завантаження файлу з локальних дисків комп'ютера;
- Можливість завантаження файлу зі знімних дисків;
- Можливість завантаження декількох файлів;
- Завантаження файлів різних форматів і розмірів;
- Завантаження файлів, що містять у назві спеціальні символи (! * <>? / \ тощо).

Функціональне тестування вебпроєктів

f) Перегляд на відповідність вмісту сторінок сайту вихідного контенту, наданого замовником

Контент – це все, що присутнє на сайті: текстовий зміст, зображення, аудіо, відео та інші файли будь-яких розширень. Для користувача контент може бути представлений для перегляду, скачки або інших дій. Все це перевіряється порівнянням з матеріалами наданими замовником.

Функціональне тестування вебпроєктів

г) Тестування взаємодії вебдодатків з сервером

Для коректної роботи вебдодатки на стороні сервера перевіряють конфігураційні налаштування, які повинні відповідати вимогам до ПЗ, що розробляється.

При перевірці вебдодатків необхідно враховувати, як:

- поведе себе додаток, якщо сервер «впаде»;
- відреагує вебдодаток після спрацьовування механізму перезапуску сервера;
- відреагує вебдодаток після неспрацьовування механізму перезапуску сервера;

Функціональне тестування вебпроєктів

- стабільно відновлює свою роботу вебдодаток при періодичному перезапуску сервера;
- повинна бути оброблена транзакція при передачі даних, якщо був збій при з'єднанні з сервером;
- обробити відключення від інтернету;
- реалізована інтелектуальна обробка помилок та наскільки інформативні для користувача повідомлення про помилки.

Тестування верстки вебсайтів

При перевірці верстки першим ділом тестувальник перевіряє розташування елементів, відповідність їх позицій наданим макетам, а так само перевіряє оптимізацію зображень і графіки.

Далі здійснюється перевірка валідності коду. У процесі верстки важливо дотримуватися коректної ієрархії об'єктів, і важливо упевнитися в її валідності за фактом завершення робіт.

Перевірка на кроссбраузерність

Перевірка на кроссбраузерність передбачає перевірку працездатності сайту в різних браузерах, а також при різних параметрах настройки екрану.

Кожен з браузерів дотримується загальних рекомендацій візуалізації розмітки сторінки, проте в той же час кожен обробляє код відповідно до особливості власного движка.

В процесі тестування враховується перелік браузерів, підтримка яких обумовлювалася із замовником на ранніх етапах обговорення проєкту.

Етап перевірки сайту на кроссбраузерність при різних розширеннях досить довгий за часом, але результат того вартий.



Usability тестування вебсайтів

Проводиться для оцінки зручності продукту у використанні, часто засноване на залученні користувачів в якості тестувальників і аналіз отриманих результатів.

Незважаючи на той факт, що опрацювання зручності використання ресурсу здійснюється в процесі складання технічного завдання, розробки макетів, бувають ситуації, коли отриманий результат не є оптимальним.

Тому найкраще рішення в даному випадку - внести зміни до реалізованого продукту.



Тестування безпеки вебсайтів

На даній стадії тестування фахівець перевіряє – чи немає у користувачів доступу до службових/закритих сторінок, а також проводить перевірку захисту всіх критично важливих сторінок (наприклад, розділу адміністрування сайту) від зовнішнього впливу.

На відміну від інших тестів, тестування безпеки слід проводити регулярно. Крім того, тестуванню піддається не тільки сам сайт або вебдодаток, а весь сервер повністю – і вебсервер, і операційна система, і всі мережеві сервіси.

Тестування продуктивності сайту

Проводиться з метою визначення швидкодії сайту або частини під певним навантаженням.

Тестування продуктивності включає в себе наступні **види тестування продуктивності:**

1. Навантажувальне тестування;
2. Тестування швидкодії

Тестування продуктивності сайту

1. Навантажувальне тестування – найпростіша форма тестування продуктивності.

Зазвичай проводиться для того, щоб оцінити поведінку сайту (або програми) під заданим **очікуваним навантаженням**. Цим навантаженням може бути, наприклад, очікувана кількість одночасно працюючих користувачів на сайті, що вчиняють задане число транзакцій за інтервал часу.

Такий тип тестування зазвичай дозволяє отримати час відгуку всіх найважливіших бізнес-функцій.

Тестування продуктивності сайту

Основні цілі навантажувального тестування:

- оцінка продуктивності і працездатності програми на етапі розробки і передачі в експлуатацію;
- оцінка продуктивності і працездатності програми на етапі випуску нових релізів, інформації, призначеної для автоматизованого внесення певних змін в комп'ютерні файли (patch);
- оптимізація продуктивності додатку, включаючи налаштування серверів та оптимізацію коду;
- підбір відповідної для цього додатку апаратної (програмної платформи) і конфігурації сервера.

Тестування продуктивності сайту

2. Тестування швидкодії – перевірка швидкості завантаження сайту для визначення швидкості відпрацювання скриптів, завантаження зображень і контенту.

Цей тест проводиться з метою оптимізації процесу завантаження сайту, а також визначення оптимальності налаштувань сервера.

У такому тесті перевіряється не тільки і не стільки сам сайт, скільки спільна злагоджена робота всього комплексу – апаратної частини сервера, вебсерверу, програмного ядра (engine) та інших компонентів сайту.



Тестування продуктивності сайту

Залежно від спрямованості тестування, перевіряється та чи інша особливість вебсайту: корпоративного сайту, інтернет-магазину, сайту-візитки тощо.

Зазвичай, процес тестування документується у вигляді тест-плану та тест-кейсів.

Тест-план описує стратегію тестування, методи та засоби тестування, порядок та інші особливості.

Тест-кейси описують послідовні покрокові операції перевірки функціоналу програми або вебсайту.



Обробка результатів

Протягом усього етапу тестування, фахівець створює та доповнює звіт про виявлені помилки. Даний звіт передається учасникам проєкту, після чого керівник проєкту визначає відповідального за виправлення кожної з помилок.

Далі визначається графік виправлення помилок, після чого проводиться повторне тестування з метою контролю якості виправлення помилок, а також відсутності нових.

По завершенню тестування проєкт готовий до розміщення на сервері і повноцінної роботи, ефективно і стабільно виконуючи покладені на нього бізнес-функції.



Інструменти розробника (Developer Tools)

Інструменти розробника – це потужний інструмент для налагодження коду вебсайтів, який за замовчуванням встановлений в багатьох браузерах (Firefox, Google Chrome, Edge та в інших).

Крім веброзробки, даний інструментарій буде дуже корисний тестувальникам при тестуванні вебсайтів і створенні баг-репортів.

Інструменти розробника (Developer Tools)

Для того, щоб в браузері **Google Chrome** відкрити вкладку Console можна:

- скористатися гарячими клавішами *Command - Option - I* (Mac) або *Control - Shift - I* (Windows/Linux);
- вибрати пункт меню *Додаткові інструменти > Інструменти розробника > Console*;
- натиснути клавішу F12.

Ці способи Developer Tools також працюють і в інших браузерах, таких як **Mozilla Firefox, Edge**.

Інструменти розробника (Developer Tools)

Для того, щоб відкрити консоль в браузері **Safari**, необхідно в налаштуваннях браузера активувати меню розробника (якщо воно відсутнє), натиснувши

Safari > Preferences > Advanced, і відмітити *Show Develop menu in menu bar*, після цього визвати консоль за допомогою поєднання клавіш

Command+Shift+I або *Develop > Show Web Inspector*.

Інструменти розробника (Developer Tools)

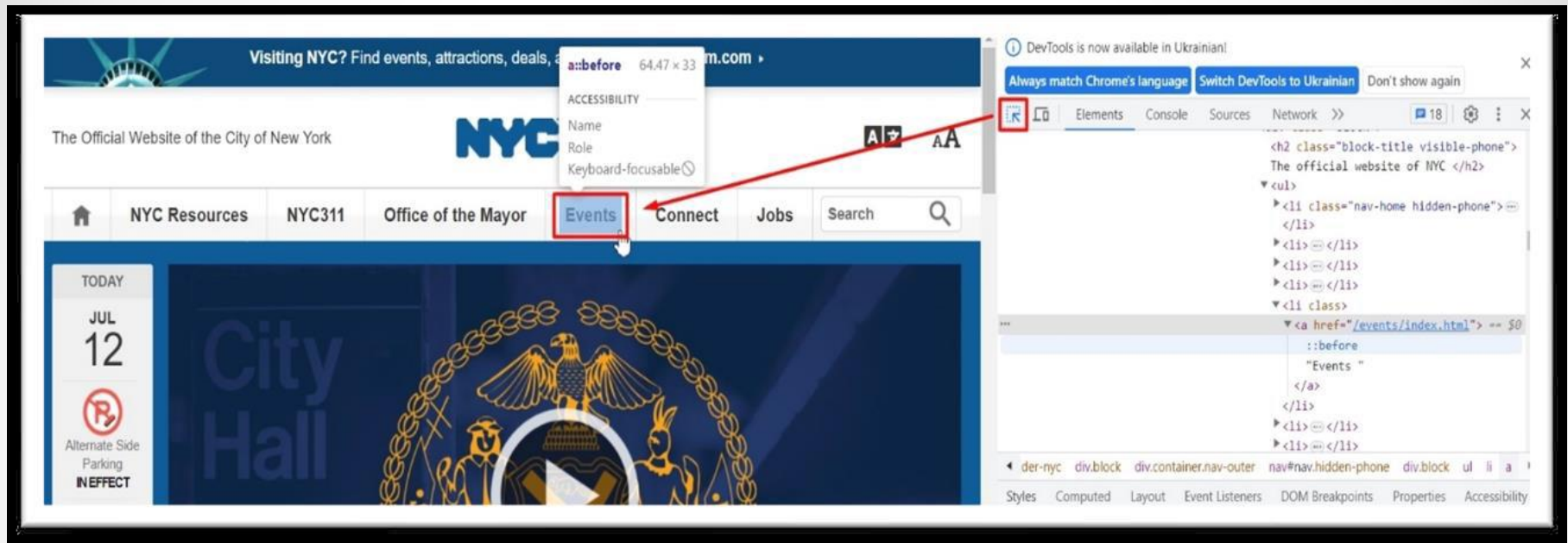
Функції *Developer Tools*:

- перегляд HTML та CSS-коду елементів вебсторінки (вкладка «Elements»);
- емуляція відображення вебсторінки на мобільному пристрої (вкладка-іконка «Toggle device Toolbar»);
- моніторинг подій на сторінці з відображенням попереджень і помилок (вкладка «Console»);
- виконання JavaScript-команд (вкладка «Console»);
- перегляд всіх ресурсів, що завантажуються на сторінці: картинок, скриптів, таблиць стилів, також можна правити скрипти і можна завантажити на ПК файли (вкладка «Sources»);
- перегляд статистики завантаження елементів, а також запитів, що відправляються сторінкою (вкладка «Network»);

Інструменти розробника (Developer Tools)

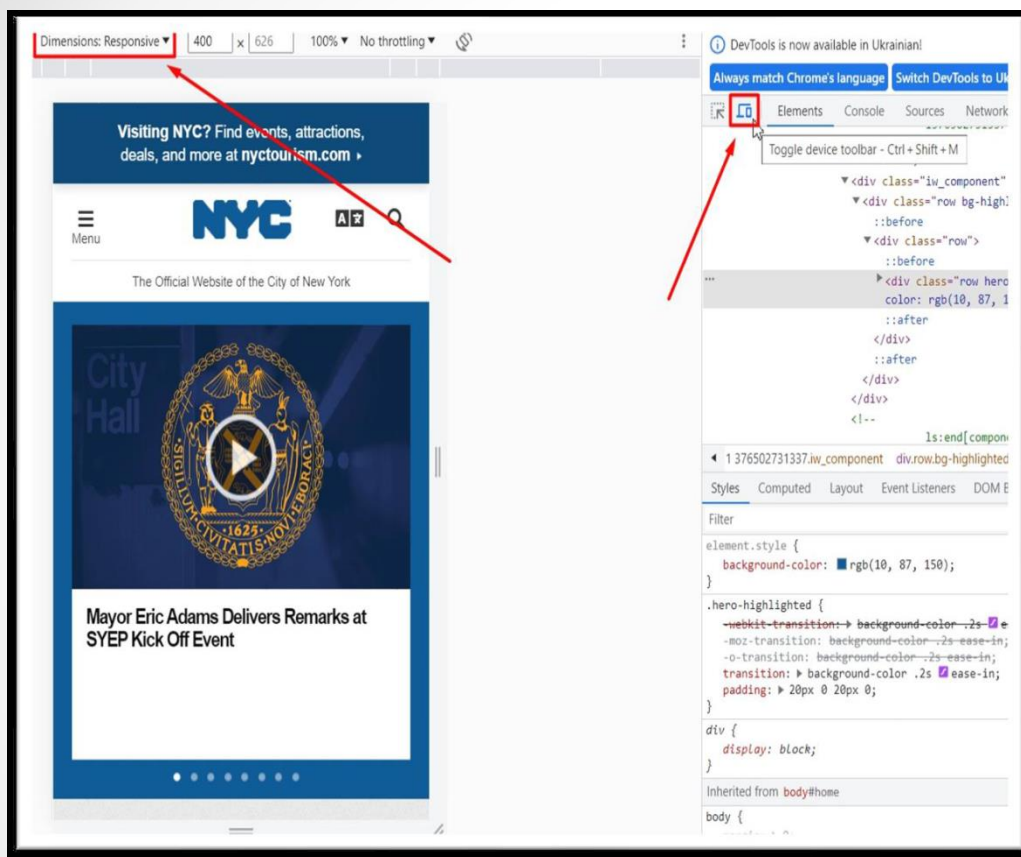
- емуляція завантаження сторінки із штучним обмеженням швидкості (вкладка «Network»);
- оцінка навантаження вебресурса на систему користувача для оптимізації (вкладка «Performance»);
- перегляд поточного використання пам'яті вебсторінкою (вкладка «Memory», наявна не у всіх версіях браузерів);
- перегляд та редагування даних сесій, cookie-файлів та інших локальних сховищ вебсторінки (вкладка «Application»);
- перегляд звіту про безпечність ресурсів (вкладка «Security»);
- перегляд рекомендацій для оптимізації сторінок (вкладка «Audits»).

1. Дослідження елементів



Крайня ліва кнопка в інспекторі активує інструмент для виділення елементів. Якщо навести вказівник миші на будь-який елемент, можна дізнатися його назву чи ID, розміри, внутрішні і зовнішні відступи, межу. Якщо клікнути на виділений елемент, у вікні нижче можна побачити його виділеним всередині структури DOM

2. Адаптивний дизайн



Наступна кнопка (Toggle device Toolbar) дозволяє вибрати режим емуляції мобільного пристрою. Це зручний інструмент для тестування в різних розширеннях екрану і перевірки відображення сайту на мобільних пристроях.

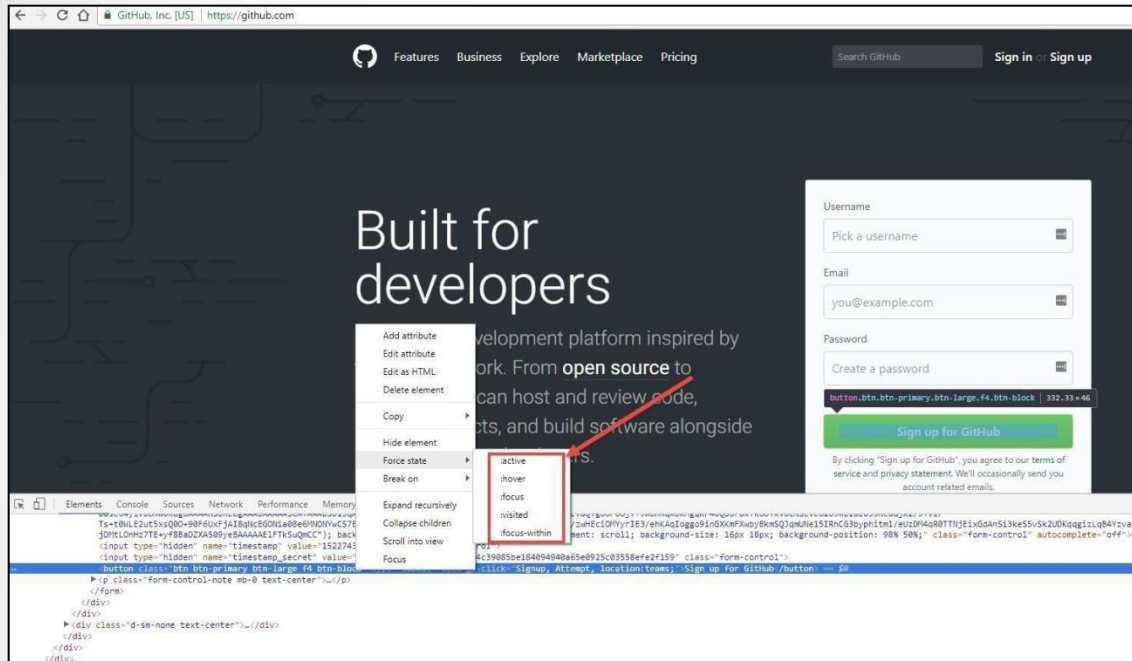
У верхньому меню можна вибрати тип пристрою для емуляції, встановити розширення екрану вручну, змінити орієнтацію з вертикальної на горизонтальну та навпаки.

3. Elements (Елементи)

The image shows a screenshot of the Facebook registration page with the DevTools 'Elements' panel open. The page content includes the Facebook logo, a blue button labeled 'Створити обліковий запис', and various form fields for name, date of birth, and gender. The 'Elements' panel on the right displays the DOM tree, with a red arrow pointing to the 'Elements' tab. The selected element is a button with the text 'Створити обліковий запис'. The panel also shows the 'Styles' pane for the selected element, displaying its default styles.

Використовується для перегляду коду сторінки, відображається у вигляді DOM – дерева елементів.

3. Elements (Елементи)



Після кліку правою клавішею миші на будь-який елемент можна додати атрибути, редагувати їх, видалити елемент із структури сторінки або приховати його, чи додати елементу псевдокласи.

Для зміни HTML-елемента необхідно вибрати потрібний тег, натиснути F2 і в блоці, що відкриється, можна буде змінити його зміст, додати інші атрибути цьому тегу, а також додати інший тег перед чи після даного.

4. Console (Консоль)

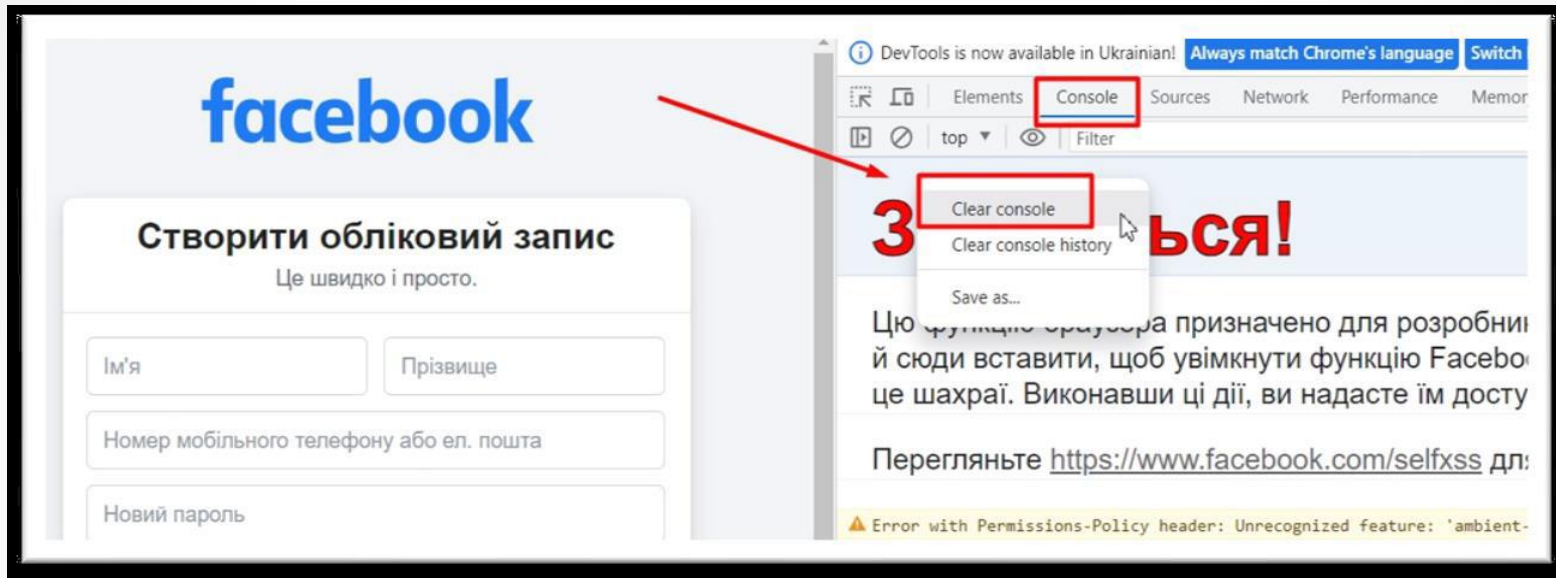
В цій вкладці відображаються важливі повідомлення про роботу сайту, помилки і попередження, логи для розробників, а також в консолі можна виконувати JavaScript код.

Очищення консолі:

Для очищення консолі необхідно:

- Клікнути правою клавішею миші і в контекстному меню вибрати пункт Clear Console.
- Ввести команду `clear()` з Command Line API в консолі.
- Визвати команду `console.clear()` із Console API із скрипта.
- Використати гарячі клавіші: $\text{⌘}K$ або $\text{^}L$ (Mac) Control – L (Windows і Linux).

4. Console (Консоль)



Якщо клікнути правою клавішею миші, то відкриється контекстне меню, в якому можна очистити консоль (Clear Console), очистити історію введення команд і параметрів в консоль (Clear Console History), зберегти лог консолі в окремий файл.

4. Console (Консоль)

Необхідно звернути увагу на те, що логи зберігаються в рамках поточної сторінки з моменту відкриття консолі, а не завантаження сторінки. Тому, якщо спочатку відкрити сторінку, а потім відкрити консоль, то помилки не будуть відображатися, хоч і будуть на сторінці.

Також варто звернути увагу на те, що консоль відображає також помилки різних встановлених плагінів і розширень для браузера, враховуючи віруси (вони також прописуються як плагіни або розширення).

5. Sources

Вкладка, в якій відображаються всі файли, що підключені до поточної сторінки, також можна переглянути їх зміст.

Якщо це зображення, то буде відображене воно саме, якщо HTML-сторінка, можна буде переглянути її код, якщо файли CSS або JavaScript - то можна буде не тільки переглядати їх зміст, але й редагувати його, або, за необхідності, скопіювати код чи зберегти його як новий файл.

Також вкладка дозволяє створювати точки зупинки (Breakpoints) для налагодження коду.

6. Network

Ця вкладка дозволяє відстежити завантаження сторінки і всіх файлів, які ця сторінка підтягує при завантаженні (назва файлів, статус (відповідь сервера), тип файлу, ініціатори файлу, розмір файлу, час, за який він був завантажений).

Timeline – послідовність завантаження всіх файлів на сторінці. В нижньому рядку відображається кількість запитів, що були надіслані серверу, загальна кількість переданих даних, загальний час завантаження всіх файлів і всіх запитів.

DOMContentLoaded – час, за який буде завантажено весь HTML-код і побудоване DOM-дерево сторінок, **Load** – завантаження усіх необхідних ресурсів, що впливають на відображення сторінки.



6. Network

Секція «Request/response»:

- Request sent - час, за який запит було відправлено на сервер.
- Waiting TTFB (Time To First Byte) - час, за який сервер опрацював запит і відправив перший байт клієнту.
- Content Download - час завантаження контенту.

Секція «Connection Start»:

- Stalled - запит може бути зупинено з будь-якої причини, що описані в Queuing.

Секція «Resource scheduling»:

- Черга запитів в браузері, яка може утворитися через наступні причини:
 - Існують запити з більш високим пріоритетом.
 - Для цього джерела вже відкрито шість TCP-з'єднань, що є межею. Використовується тільки для HTTP / 1.0 и HTTP / 1.1.
 - Браузер короткочасно розподіляє час для кешу.

6. Network

Якщо клікнути на файл із списку у вкладці Name, то відкриється додаткове вікно, в якому відображені властивості обраного елемента.

Коди відповіді сервера поділяються на 5 основних класів:

1xx – інформаційні коди, що відповідають за процес передачі даних. Вони тимчасові, інформують про те, що запит прийнято і обробка триває.

2xx – успішна обробка. Запит було отримано і успішно оброблено сервером.

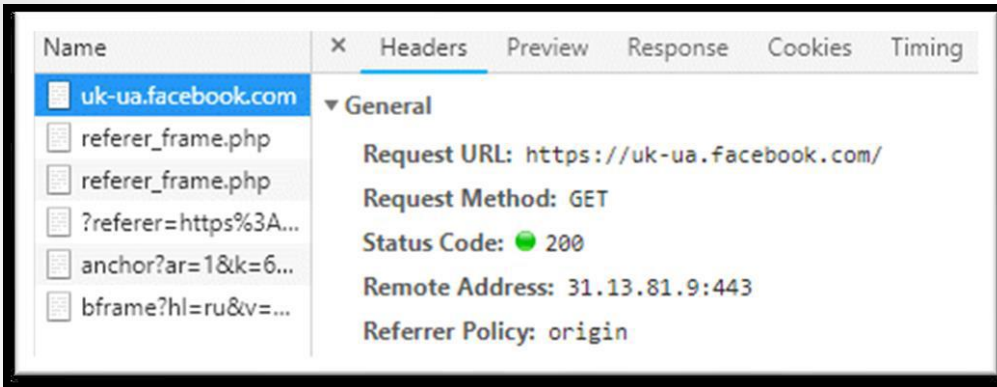
3xx – перенаправлення (редирект). Ці відповіді сервера повідомляють про те, що необхідно виконати подальші дії для виконання запиту.

4xx – помилка користувача. Це означає, що запит не може бути виконаний з його вини.

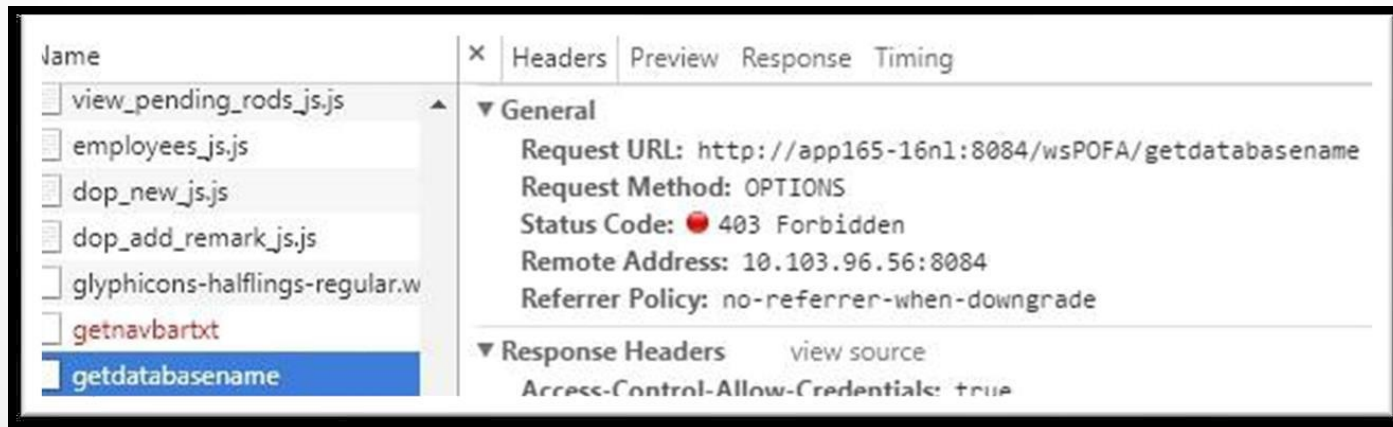
5xx – помилка сервера. Обов'язково відображається повідомлення, що не може опрацювати запит і з якої причини.

6. Network

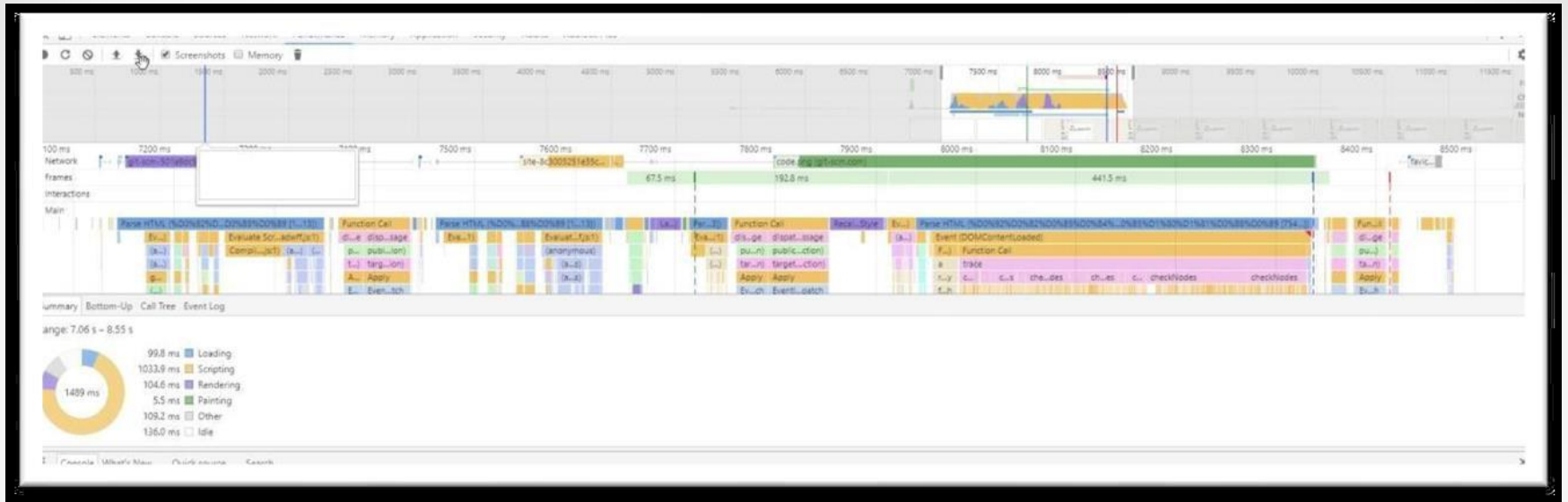
Наприклад, код зі статусом 200 ОК означає, що запит виконано успішно відповідно до очікувань користувача – потрібні дані існують і доступні для перегляду.



Код 403 Forbidden (Відмовлено у доступі) повертається у тому випадку, коли користувачу заборонений доступ до документа.

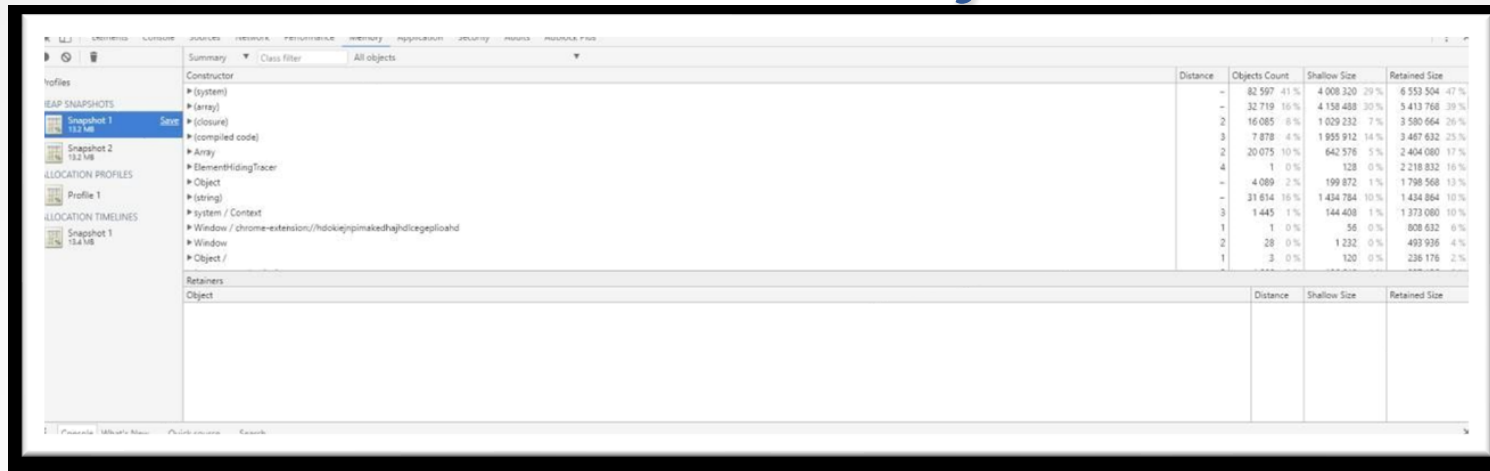


7. Performance



В цій вкладці також дублюється рядок Timeline, як і у вкладці Network, але в більш розгорнутому варіанті. Окрім мережі в цій вкладці можна відстежувати виконання JavaScript-коду і завантаження пам'яті, скільки процесорного часу це займає. Може використовуватися тестувальниками для перевірки продуктивності, швидкості завантаження сайту, пошуку слабких місць і можливих шляхів оптимізації.

8. Memory



The screenshot shows the Chrome DevTools Memory tab. The left sidebar has a tree view with categories: Profiles, HEAP SNAPSHOTS (Snapshot 1: 13.1 MB, Snapshot 2: 13.2 MB), ALLOCATION PROFILES (Profile 1), and ALLOCATION TIMELINES (Snapshot 1: 13.4 MB). The main area is split into two panes. The top pane is titled 'Summary' and shows a table of memory usage statistics. The bottom pane is titled 'Retainers' and shows a table of objects that retain other objects.

Distance	Objects Count	Shallow Size	Retained Size
-	82 597 41%	4 008 320 29%	6 553 504 47%
-	32 719 16%	4 158 488 30%	5 413 768 39%
2	16 085 8%	1 029 232 7%	3 580 664 26%
3	7 878 4%	1 959 912 14%	3 467 632 25%
2	20 075 10%	642 576 5%	2 404 080 17%
4	1 0%	128 0%	2 218 832 16%
-	4 089 2%	199 872 1%	1 798 568 13%
-	31 614 16%	1 434 784 10%	1 434 864 10%
3	1 445 1%	144 408 1%	1 373 080 10%
1	1 0%	56 0%	808 632 6%
2	28 0%	1 232 0%	493 936 4%
1	3 0%	120 0%	236 176 2%

Object	Distance	Shallow Size	Retained Size
--------	----------	--------------	---------------

Вкладка, за допомогою якої можна відслідковувати навантаження виконання JavaScript коду на пристрій, що тестується. Має три профілі:

- Heap snapshot – відображає розподіл пам'яті серед об'єктів в кодї сторінки, а також пов'язані з ними DOM-елементи.
- Allocation instrumentation on timeline – відображає розподіл пам'яті серед змінних в кодї сторінки, цей метод застосовується для ізоляції витоків пам'яті.
- Allocation sampling – відображає розподіл пам'яті на виконання JavaScript-функцій.

9. Application

Вкладка, в якій відображаються доступні сховища:

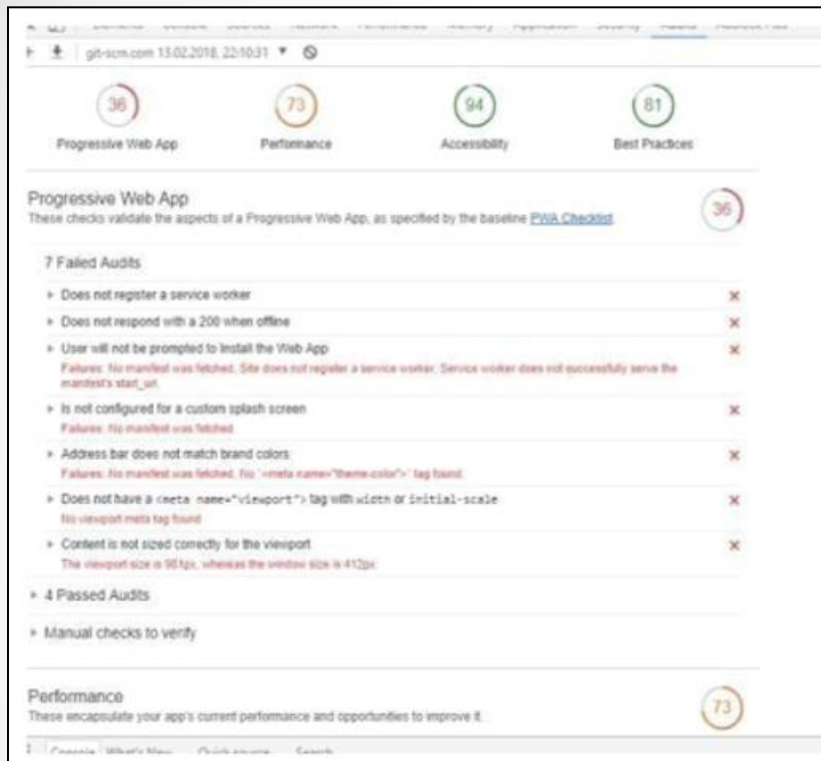
- Local storage – відображає, де можна переглянути, які змінні зберігає сторінка та їх значення, які можна змінити у разі необхідності.
- Session Storage – зберігаються дані поточної сесії.
- Cookies – зберігаються всі дані cookies сайту.

Очистити сховища сайту можна за допомогою кнопки Clear. Очищення необхідно виконувати після оновлень версій продукту, оновлення файлів конфігурації, стилів, додавання/зміни функціональностей вебсайту та інших змін на сервері.

10. Security

Відображає протокол безпеки, якщо він існує, а також дані про сертифікат даного підключення.

11. Audits



Вкладка, яка допомагає визначити деякі проблеми сайту, що тестується.

Відбуваються перевірки того, чи ввімкнене стиснення, кешування, визначається, чи є невикористовувані CSS правила.

За допомогою цього інструмента можна провести аудит сайту і переглянути

проблемні місця, що впливають на швидкість роботи. Можна провести аудит як для десктопної, так і для мобільної версії сайту, обравши відповідні параметри в налаштуваннях перед запуском.