

Лабораторна робота № 3
Вебтестування та чек-листи.
Кросбраузерне тестування



Мета лабораторної роботи

1. Отримання базових теоретичних та практичних навичок в розпізнаванні етапів тестування та створенні контрольного списку (checklist).
2. Отримання базових теоретичних та практичних навичок в кросбраузерному тестуванні.

Теоретичний матеріал

Вебтестування (web-testing) – тестування програмного забезпечення, сфокусоване на вебдодатках (web-applications), яке здійснюється з застосуванням систем, що використовують вебтехнології (web-based system), та вирішують такі питання як: безпека вебдодатків, базова функціональність сайту, доступність користувачам тощо.

Важливою особливістю вебтестування є часові рамки. На вебтестування може виділятися до 50% загального бюджету та часових ресурсів проєкту.

Теоретичний матеріал

Розрізняють наступні **етапи тестування вебпроектів**

(web-project stages of testing):

1. Вивчення документації (documentation analysis).
2. Тестування верстки (layout testing).
3. Функціональне тестування (functional testing).
4. Тестування зручності використання (usability testing).
5. Тестування безпеки (security testing).
6. Тестування продуктивності сайту (performance testing).

Теоретичний матеріал

Вимоги до розмітки (верстки):

- чи немає помітних оку невідповідностей: поламани блоки, не стикування кольору, некоректне відображення тексту навколо зображень;
- якщо дизайн згідно ТЗ розрахований на певну ширину, то, незалежно від браузера, не повинна з'являтися горизонтальна прокрутка;
- під час зменшення розміру вікна менше мінімального згідно ТЗ не повинно нічого ламатися, фони не повинні «розпливатися»;
- сайт повинен виглядати однаково у всіх стандартних розширеннях екрану (1024x600, 1024x768, 1152x864, 1280x800, 1280x1024, 1440x900, 1680x1050, 1920x1080): не повинно нічого ламатися, не повинні різко обриватися фони при великих розширеннях;

Теоретичний матеріал

- при відключених зображеннях, написи (особливо логотип та головне меню сайту) повинні залишатися читабельними, на всіх інформаційних картинках повинні бути підписані акуратним, невеликим, сірим шрифтом (відключення картинок: Chrome → Settings → Search settings → JavaScript → Site settings → Images → вибрати опцію «Don't allow sites to show images»; Firefox → about:config → permissions.default.image → 2);

- працездатність при вимкненому JavaScript. Увесь критично важливий функціонал сайту повинен бути доступний без JS (відключення JavaScript: Chrome → Settings → Search settings → JavaScript → JavaScript (Content) → Don't allow sites to use JavaScript; Firefox → about:config → Accept the Risk and Continue → javascript.enabled → Toggle the «javascript.enabled» preference).









Теоретичний матеріал

Перевірка однотипності/симетричності в інтерфейсі:

- однотипність/симетричність використання відступів;
- ширина колонок та контентного поля;
- позиціонування елементів верхнього та нижнього колонтитулів на всіх сторінках;
- позиціонування банерів, правого та лівого блоку;
- розмір текстів;
- відступи між абзацами;
- відстані між рядками;
- елементи управління (кнопки, чекбокси, випадаючі списки);
- розмір/колір/тип шрифту;

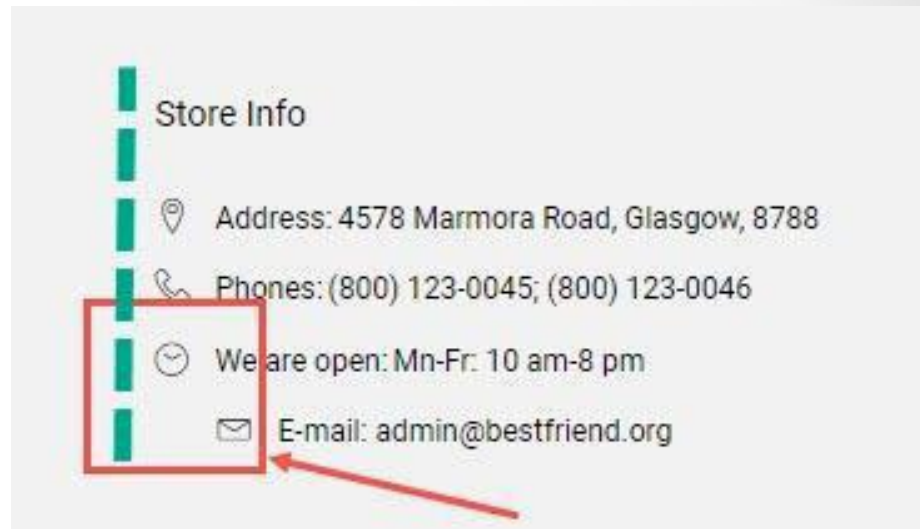
Теоретичний матеріал

- наявність стилів оформлення заголовків першого, другого та більше рівнів;
- коректне відображення тексту навколо зображень;
- напрямки тіней у всіх елементів управління;
- назви однакових елементів у різних місцях;
- чи мають елементи, що призначені для натискання, вказівник «pointer»; елементи, які можна перетягувати – вказівник «move» або «crosshair»; неактивні/недоступні елементи – вказівник «default»

Вид	Значення
	default
	crosshair
	help
	move
	pointer
	progress
	text
	wait

Приклади дефектів

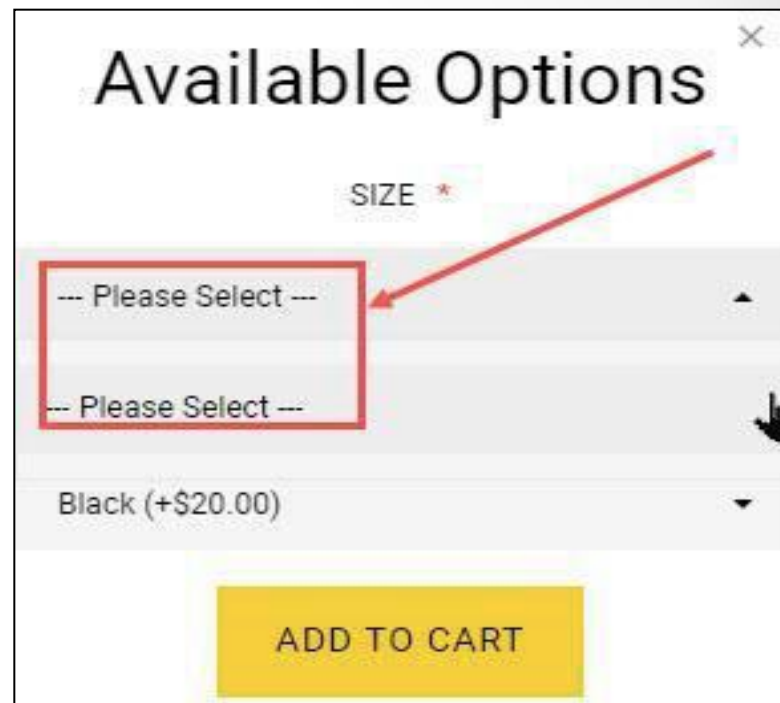
Рядки блоку не вирівняні відносно один одного



Відстані між підписами полів та міткою обов'язкового поля не збігаються. Довжина та розташування поля «Quantity» не збігається з полем «Starch»

Приклади дефектів

Не для всіх випадючих списків є значення для вибору



Приклад нецентрованих іконок випадючого меню

Приклади дефектів

Підпис не вирівняний щодо поля



A registration form with the following labels and input fields:

- Your Name: [input field]
- Your Email: [input field]
- Friend's Name: [input field]
- Friend's Email: [input field]
- Personal Note: [text area]

A red arrow points to the 'Personal Note' label, which is misaligned with the input field above it. A 'SEND' button is located at the bottom right.



Enter your phone number:

 (+3) e number without country

Continue

A red arrow points to the default text 'e number without country' in the phone number input field.

Текст за замовчуванням обрізається в полі введення номера телефону

Теоретичний матеріал

Чек-лист – це документ, який описує що має бути протестовано. При цьому чек-лист може бути абсолютно різного рівня деталізації. На скільки детальним буде чек-лист залежить від вимог до звітності, рівня знання продукту співробітниками й складності продукту.

Навіщо потрібен чек-лист?

- не забути необхідні тести;
- для поділу завдань за рівнем кваліфікації;
- для збереження звітності та результатів тестування.

Що має бути в чек-листі?

- список перевірок (з необхідним ступенем деталізації);
- оточення перевірки;
 - збірка, на якій проводилося тестування;
 - тестове оточення (якщо є);
 - інформація про тестувальників;
- результат перевірки.

Теоретичний матеріал

При проходженні чек-листів тестувальник зазначає статус навпроти кожного тестованого пункту. Можливі такі варіанти статусів:

- «*Passed*» – перевірка пройдена успішно, багів не знайдено;
- «*Failed*» – знайдений один або більше багів;
- «*Blocked*» – неможливо перевірити, тому що один з багів блокує поточну перевірку;
- «*In Progress*» – поточний пункт, над яким працює тестувальник;
- «*Not run*» – ще не перевірено;
- «*Skipped*» – не буде перевірятися з будь-якої причини (наприклад, поточний функціонал ще не реалізований).

Для більшої наочності, як правило, кожен зі статусів має свій колір.

Теоретичний матеріал

Основні моменти, які варто враховувати при роботі з чек-листами:

- після завершенні проходження чек-листа не повинно залишитися комірок зі статусом «*Not run*»;
- всі комірки зі статусом «*Failed*» і «*Blocked*» обов'язково повинні мати примітки з посиланнями на баг-репорти;
- статус «*Passed*» встановлюється тільки для пунктів, які перевірені і не містять помилок.

Правила складання чек-листів:

1. Один пункт – одна операція.
2. Починати пункти з іменника.
3. Скласти чек-лист за рівнями деталізації.

Теоретичний матеріал

Кросбраузерність – властивість сайту відображатися та працювати у всіх браузерах ідентично. Під ідентичністю розуміється відсутність розвалів верстки та здатність відображати матеріал з однаковим ступенем читабельності.

Так як вебтехнології весь час розвиваються, прийнятну кросбраузерність можна забезпечити тільки для останніх версій різних браузерів.

На практиці зазвичай обмежуються тільки найпопулярнішими браузерами, що суттєво може скоротити час на розробку сайту.

Теоретичний матеріал

Правила кросбраузерності:

- під час застосування будь-яких змін в дизайні сайту необхідно звертати увагу, який має вигляд дизайн в інших браузерах;
- потрібно тестувати сайти самостійно за допомогою встановлених на комп'ютері браузерів, інколи використовуючи спеціальні онлайн сервіси;
- необхідно здійснювати тестування кросбраузерності на всіх доступних пристроях (комп'ютер, ноутбук, планшет тощо);
- завжди повинен бути встановлений на комп'ютері комплект найбільш популярних браузерів;
- різні версії одного й того ж браузера будуть відображати сайт теж по-різному і це потрібно враховувати, але намагатися використовувати найостанніші версії.

Завдання до лабораторної роботи №3

1. Ознайомитися з матеріалом лекцій «Вебтестування та чек-листи» та «Кросбраузерне тестування».

2. Обрати реальний сайт (інтернет-магазин, сайт організації чи установи тощо), функціонуючий на даний момент для проведення тестування.

Зверніть увагу, що сайти з доменом «ru» прийматись не будуть, оскільки використання подібних інтернет-ресурсів заблоковано в Україні на рівні законодавства.

3. Завантажити файл «Чек-лист Верстка.xlsx», що знаходиться на головній сторінці дисципліни в системі Мудл у розділі «Лабораторні роботи».

4. Вказати прізвище та ім'я власника у назві чек-листа.

Завдання до лабораторної роботи №3

5. Провести тестування верстки обраного у пункті 2 сайту, перевіривши всі пункти чек-листа мінімум в трьох браузерах останніх або передостанніх версій (Google Chrome, Firefox, Edge, Opera або інші). Вказати назву та версію браузерів.

6. Результат перевірки відзначити «Passed/Failed». Для «Failed» вказати в примітці або нотатках посилання на файл зі скріншотом баг-репорту, розробленого у системі відслідковування помилок Mantis Bug Tracker (не менше 3-х). Для всіх інших багів вказати примітки з темою бага (за принципом Що? Де? Коли?).

7. У відповіді до завдання в Мудлі прикріпити:
– пройдений та повністю заповнений чек-лист;
– скріншоти всіх створених баг-репортів.

8. Опрацювати контрольні питання.

Оцінювання лабораторної роботи №3

1. Максимальна кількість балів за роботу - 100:

- чек-лист до 25 балів;

- кожен з трьох баг-репортів до 25 балів.

2. Оцінювання чек-листа відбуватиметься за критеріями, що наведені у файлі *«Критерії оцінки чек-листів.xlsx»*, баг-репортів – *«Критерії оцінювання баг репортів.xlsx»*, що наведені на сторінці дисципліни в Мудлі у розділі «Лабораторні роботи» .

3. За невідповідність кожному з критеріїв буде зніматись певна кількість балів.

4. У разі надсилання «чужої» роботи (тобто роботи з ознаками плагіату) бали не нараховуються взагалі.

Контрольні питання

1. Що відносять до дефектів розмітки (верстки)? Навести приклади та обґрунтувати відповідь.

2. Для чого в примітці до пункту чек-листа вказується посилання на звіти у системі відслідковування помилок? Обґрунтувати необхідність посилання на звіт в чек-листі.

3. Які існують переваги використання чек-листів порівняно з іншою подібною документацією для проведення тестування? Навести приклади.

4. Обґрунтувати необхідність та важливість кросбраузерного тестування.

5. За якими чинниками визначають, в яких браузерах проводити тестування?



Додаткові джерела інформації

1. Як тестувати веб-сайт: основні етапи і поради [Електронний ресурс]. – Режим доступу: <https://brainlab.com.ua/uk/blog-uk/yak-testuvati-veb-sayt-osnovn-etapi-poradi>.
2. Software Testing Help. «Web Testing: Complete guide on testing web applications» [Електронний ресурс]. – Режим доступу: <http://www.softwaretestinghelp.com/web-application-testing/>.
3. Web Application Testing: 8 Step Guide to Website Testing [Електронний ресурс]. – Режим доступу: <https://www.guru99.com/web-application-testing.html>.
4. Що таке чеклісти та як з ними працювати [Електронний ресурс]. – Режим доступу: <https://training.qatestlab.com/blog/technical-articles/work-with-checklist/>.
5. Тестування верстки. Основні вимоги [Електронний ресурс]. – Режим доступу: <https://training.qatestlab.com/blog/technical-articles/testing-layout-basic-requirements/>.
6. Як правильно описувати схожі баги або один баг у різних браузерах [Електронний ресурс]. – Режим доступу: <https://training.qatestlab.com/blog/course-materials/bug-description-browser/>.
7. Кросбраузерне тестування: навіщо та кому потрібно його проводити [Електронний ресурс]. – Режим доступу: <https://training.qatestlab.com/blog/technical-articles/cross-browser-testing/>.
8. Основні браузери для тестування [Електронний ресурс]. – Режим доступу: <https://training.qatestlab.com/blog/technical-articles/browsers-used-testing/>.
9. Інструменти для тестування кросбраузерності верстки [Електронний ресурс]. – Режим доступу: <https://training.qatestlab.com/blog/technical-articles/tools-for-testing-cross-browser-layout/>.