

3 ЛАБОРАТОРНА РОБОТА №3

ВЕБТЕСТУВАННЯ, ЧЕК-ЛІСТИ ТА КРОСБРАУЗЕРНЕ ТЕСТУВАННЯ

Мета лабораторної роботи: отримання базових теоретичних та практичних навичок в розпізнаванні етапів тестування та створенні контрольного списку (checklist), отримання базових теоретичних та практичних навичок в кросбраузерному тестуванні.

3.1 Короткі теоретичні відомості

Чек-ліст – це зручний і структурований інструмент, який допомагає тестувальникам у проведенні перевірки програмного забезпечення. Він являє собою список завдань, кроків і критеріїв, які необхідно виконати для ретельної перевірки функціональності або інших аспектів ПЗ.

Використання чек-ліста в процесі тестування ПЗ надає кілька переваг. Ось деякі з них:

- **Нічого не пропустити!** Уявіть, що у вас є величезний застосунок із безліччю функцій і можливостей. Без чек-ліста, ви можете забути перевірити якісь важливі аспекти і упустити помилки. Чек-ліст допомагає не загубитися і бути впевненими, що покриті всі основні перевірки. Він нагадує про кожен крок, який потрібно виконати, щоб нічого не пропустити.
- **Структурованість і організація.** У роботі тестувальника дуже важливо мати структуру й організацію. Чек-ліст це і надає: допомагає розбити процес на більш дрібні завдання і слідувати за порядком. Це особливо корисно, коли у є великий обсяг роботи або обмежений час. Чек-ліст дає нам чітке уявлення про те, що потрібно робити і в якій послідовності.
- **Уніфікація та консистентність.** У команді QA може працювати кілька тестувальників, і кожен із них може мати свої вподобання та методи роботи. Чек-ліст допомагає уніфікувати процес і встановити єдині стандарти для всіх. Коли кожен дотримується одного й того самого чек-ліста, ми можемо бути впевнені, що всі основні перевірки будуть виконані. Це підвищує консистентність і якість роботи команди.

З яких ключових моментів складається чек-ліст тестування? Їх кілька і вони допомагають організувати та впорядкувати процес перевірки. Ось деякі з них:

- **Завдання та функціональність.** Чек-ліст має містити список завдань і функціональності, які слід перевірити. Це може бути щось на кшталт “перевірити, що кнопка ‘Відправити’ надсилає дані форми” або “переконатися, що функція пошуку знаходить усі результати, які повинна”. Простіше кажучи, це список дій і перевірок, які слід виконати, щоб переконатися, що все працює належним чином.
- **Критерії оцінки** або очікувані результати, які порівнюються з фактичними результатами перевірок. Наприклад, якщо перевіряється функція пошуку, критерієм оцінки може бути те, що за правильного запиту мають бути видані всі відповідні результати. Якщо це відбувається, відзначається завдання як виконане успішно. Якщо ні, значить є недолік, якому потрібно приділити увагу.
- **Пріоритети.** У чек-лісті також можна вказати пріоритети для кожного завдання. Це допомагає визначити, які з них слід виконати першими, щоб зосередитися на найважливіших і критичних аспектах ПЗ. Пам’ятайте, може бути обмежений час або ресурси, тому вміння пріоритизувати – ключова навичка!

При проходженні чек-лістів 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.

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).

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;
- 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» (рисунок 3.2).







































default		
none	-	-
context-menu		
help		
pointer		
progress		
wait		
cell		
crosshair		
text		
vertical-text		
alias		
copy		
move		
no-drop		
not-allowed		
all-scroll		
col-resize		
row-resize		
n-resize		

Рисунок 3.2 – Форма вказівника при наведенні на відповідний елемент

Кросбраузерність – властивість сайту відображатися та працювати у всіх браузерах ідентично. Під ідентичністю розуміється відсутність розвалів верстки та здатність відображати матеріал з однаковим ступенем читабельності. Поняття «кросбраузерність» дуже часто плутають з попиксельною відповідністю, що насправді є різними поняттями. 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.

Правила кросбраузерності:

- під час застосування будь-яких змін в дизайні сайту (чи то новий інформер, рекламний блок, новий шаблон або ділянка шаблонів) – 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 та ін.);
- завжди повинен бути встановлений на комп'ютері комплект найбільш популярних браузерів, які не вичерпують ресурси жорсткого диска, але якщо буде потрібно перевірити кросбраузерність сайту – вебмастер запускає їх та проводить необхідний аналіз;
- різні версії одного й того ж браузера будуть відображати сайт теж по-різному і це потрібно враховувати, але намагатися використовувати найостанніші версії.

Основні проблеми кросбраузерності сайту лежать на вищих етапах його створення – під час створення шаблону. Існує багато готових шаблонів, з яких можна вибрати той єдиний, який буде однаково (або з незначними змінами) відображатися у всіх браузерах, а потім допрацювати його під свій проєкт, але з постійним орієнтиром на різні браузери.

Інструменти тестування кросбраузерності верстки:

1. ***Browsershots***, ймовірно, володіє найширшим набором браузерів серед безкоштовних інструментів тестування, включає в себе браузери, що працюють в Linux, Windows та BSD. Серед них є такі, про які взагалі ніколи не чули (наприклад, Galeon, Iceape, Kazehakase або Eriphany). Browsershots дозволяє тестувати як в останніх версіях кожного браузера, так і в застарілих версіях. Хоча Browsershots дозволяє тестувати у величезній кількості браузерів, варто пам'ятати, чим більший набір браузерів для тестування, тим довше доведеться чекати результату. Так що варто зупинитися на основних браузерах.

2. ***Browser Sandbox*** – інструмент, який буде корисним тільки користувачам Windows. Він підтримує такі браузери, як Firefox, Chrome, ChromiumCanary, Firefox Mobile, Safari, Opera та

FirefoxNightly. В безкоштовному варіанті існує можливість тестування тільки в останній версії певного браузера. Для тестування в старих версіях необхідно користуватися платною версією.

3. Microsoft Edge – це справжня платформа для тестування сайтів в ІЕ. В ньому можна отримати скріншот сайту в багатьох браузерах, в тому числі і мобільних, але на основі платформи Microsoft Edge можна розгорнути віртуальну машину тільки для тестування в ІЕ7 і новіших.

4. CrossBrowserTesting – сервіс, який використовує реальні пристрої для тестування сайту. Сервіс платний, але надає можливість тестувати більш як в 900 браузерах і приблизно в 40 операційних системах. Ще одна його особливість полягає в режимі live testing, в якому можна тестувати сайт в реальному оточенні, перевіряючи дієздатність AJAX, HTML-форм, JavaScript, Flash тощо.

5. Sauce Labs (безкоштовна та комерційна версії) – надає доступ до безлічі браузерів в різних ОС та встановлює з'єднання браузера з налагодженою віртуальною машиною. Також записується відео всієї сесії тестування. Sauce надає 200 хвилин безкоштовного тестування на місяць і дозволяє створювати тести автоматизованого тестування в браузерах (використовується Selenium).

6. Browsera (безкоштовна та комерційна версії) – забезпечує автоматизацію тестування сумісності. Він автоматично визначає відмінності в відображенні сторінок браузерами, тим самим спрощуючи процес тестування. Також визначаються помилки JavaScript, а в комерційній версії дозволяє тестувати сторінки за передплатою та сторінки, що вимагають авторизації. Також можна протестувати динамічні сторінки. Безкоштовна версія включає в себе достатньо обмежене число браузерів та низький дозвіл. Існують різні комерційні версії, які підтримують більшу кількість браузерів, забезпечують високий дозвіл та дозволяють тестувати «закриті» сторінки.

7. BrowserStack – ще один з найвідоміших сервісів для кросбраузерного тестування. Він також надає реальні пристрої для тестування і підтримує більш як 700 браузерів. Існує можливість локального тестування та швидкого отримання скріншотів на різних розширеннях екранів від 800-600 до 2048-1536. Сервіс платний, але для деяких open source проєктів пропонуються безкоштовні послуги.

3.2 Завдання до роботи

3.2.1 Завантажити шаблон чек-ліста «Чек-ліст Верстка.xlsx» (з навчальної системи Moodle). Вказати прізвище та ім'я власника у назві контрольного списку.

3.2.2 Провести тестування верстки сайту <http://prestashop.qatestlab.com.ua/en/>, перевіряючи всі пункти контрольного списку на вкладці «Верстка» у 3 браузерах (Firefox, Google Chrome, Edge, Opera або інший) в останній або передостанній версії.

3.2.3 Результат перевірки відзначити «Passed/Failed». Для «Failed» вказати в примітці або нотатках посилання на баг-репорти (2 шт.), якщо багів буде більше 2, то додати в примітках тему багу за принципом «Що? Де? Коли?».

3.2.4 Створити 2 баги-репорти для багів, знайдених під час проведення тестування верстки, оформити їх та додати до звіту.

3.2.5 Скриншот пройденого чек-ліста додати до звіту.

3.2.6 Оформити звіт з лабораторної роботи та надати його на перевірку.

3.3 Зміст звіту

3.3.1 Тема та мета роботи.

3.3.2 Завдання до роботи.

3.3.3 Короткі теоретичні відомості.

3.3.4 Хід роботи з виконання завдання до лабораторної роботи.

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

3.3.6 Перелік використаних джерел.

3.4. Контрольні запитання

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

3.4.2. За яким принципом були додані нові пункти до чек-ліста? Обґрунтувати важливість нових пунктів при тестуванні сайту <http://prestashop.qatestlab.com.ua/en/>.

3.4.3. Для чого в примітці до пункту чек-ліста вказується посилання на звіти у системі відслідковування помилок? Обґрунтувати необхідність посилання на звіт в чек-лісті.

3.4.4. Обґрунтувати необхідності створення пунктів та підпунктів в чек-лісті.

3.4.5. Які існують переваги використання чек-лістів порівняно з іншою подібною документацією для проведення тестування? Навести приклади та обґрунтувати відповідь.

Обґрунтувати необхідність та важливість кросбраузерного тестування.

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

3.4.7. Перерахувати існуючі розширення у різних браузерах для тестування вебінтерфейсів, вказати доцільність та ефективність застосування.

3.4.8. Для чого здійснюється та як впливає на роботу очистка історії (включаючи cookies, cache та ін.) у браузері? Навести приклад роботи браузера до очистки та після.

3.4.9. Назвати програму, яка була використана для тестування кросбраузерного тестування у лабораторній роботі, обґрунтувати вибір даної програми.