ЗМІСТ

ВСТУП

Актуальність дослідження. У зв’язку зі зростанням інтересу та поширенням використання різноманітних вебдодатків та сайтів не лише представників молодшого покоління та спеціалістів, а й широкої громадськості, стрімко зростає і кількість робіт, присвячених проблемам їх функціонування, зручності використання та подальшого розвитку.

Водночас методи оцінки зручності використання (usability) подібних систем переживають період незворотних трансформацій внаслідок формування абсолютно нових типів аналізу на основі даних користувацьких логів. Усе це характеризується виробництвом знань, інтеграцією технологій та розвитком інформаційних децентралізованих мереж, які дозволяють, використовуючи Інтернет як основу для створення віртуальних систем, проводить чисельні дослідження дистанційно, у режимі реального часу та без необхідності мануальної перевірки кожного етапу тестування вебсайту чи додатку. При цьому подібні системи мають більший рівень надійності та валідації у порівнянні з традиційними методами оцінювання зручності сайтів та технологічних продуктів загалом.

Сутність автоматизованої оцінки usability сайту визначає основні переваги методів, які безпосередньо розроблені на основі цієї концепції: низька вартість обслуговування, відсутність територіальних обмежень, робота у режимі реального часу та високий рівень об’єктивності результатів. В результаті моделі автоматизованої оцінки зручності використання сайтів отримують все більшого поширення, зокрема серед експертів та дизайнерів інтернет-магазинів. Однак, проблеми та ризики використання подібних технології зумовлюють необхідність проведення подальших досліджень у цій сфері. В цьому контексті, актуальним стає дослідження автоматизованої оцінки зручності використання сайтів на основі аналізу даних користувацьких логів, інструментів подібного тестування та потенціалу до розвитку.

Мета дослідження – покращення якості автоматизованої оцінки usability сайту на основі аналізу даних користувацьких логів.

Об’єкт дослідження: процес автоматизованої оцінки usability сайту.

Предмет дослідження: метод автоматизованої оцінки usability на основі аналізу даних користувацьких логів.

Для досягнення поставленої мети необхідно вирішити наступні **завдання:**

- встановити сутність поняття «usability» і визначити завдання, цілі та стандарти тестування сайтів;
- висвітлити особливості оцінки зручності використання вебсайтів на основі користувацьких логів;
- проаналізувати доступні методи автоматизованої оцінки сайтів;
- розробити метод автоматизованої оцінки на основі логів користувачів інтернет-магазину Etsy;
- виокремити загрози, ризики та контекст використання розробленого методу.

Об'єктом дослідження є методи автоматизованої оцінки usability сайтів.

Предметом дослідження є обмеження автоматизованої оцінки зручності використання вебсайту Etsy на основі даних користувацьких логів.

Методи дослідження: математичні – статистичні методи; емпірико-теоретичні: абстрагування, аналіз, синтез, методи математичного моделювання, емпіричні методи, евристики, експертні оцінки, візуалізація; методи проектування та розробки програмного забезпечення.

Наукова новизна магістерської роботи полягає в розробці методу автоматизованої оцінки usability сайту, який грунтується на даних користувацьких логів.

Практичне значення одержаних результатів дослідження полягає в можливості використання методу автоматизованої оцінки usability сайту на основі користувацьких логів для цілей подальшої оптимізації сторінок сайту по основним показникам, що впливають на досвід та враження користувача від використання сайту.

1. ТЕОРЕТИЧНІ ОСНОВИ ОЦІНЮВАННЯ ЯКОСТІ ІНТЕРФЕЙСУ САЙТУ

1.1. Завдання, цілі та стандарти тестування usability сайтів

За словами Шакеля (1984), перше зареєстроване використання терміну usability датується 1842 роком. З того часу це поняття увійшло в повсякденну мову і є повсюдним у взаємодії людини з інформаційними системами [1].

Однак, концепція автоматизованої оцінки на основі користувацького досвіду є новішою. Хоча, сьогодні поняття usability та користувацького досвіду часто використовуються в тандемі. Ці дві концепції багато в чому збігаються, але підкреслюють різні аспекти взаємодії користувачів із продуктом.

Міжнародна організація стандартизації надає широко використовувані визначення обох понять [2]:

- зручність використання: ступінь, до якої система, продукт або послуга можуть використовуватися певними користувачами для досягнення визначених цілей з ефективністю, ефективністю та задоволенням у визначеному контексті використання;
- взаємодія з користувачем: сприйняття та реакції людини в результаті використання та/або очікуваного використання продукту, системи чи послуги.

У тесті на usability користувачі використовують певний продукт, вирішуючи заздалегідь визначені завдання, тобто вони працюють над певними цілями. Єдиним елементом, яким можна пожертвувати, є контекст використання, який здебільшого відсутній, коли тест на зручність використання проводиться в лабораторних умовах, далеко від реального середовища користувачів.

Враховуючи, що більшість людей прагнуть використовувати продукти, які легко зрозуміти, тобто працюють, як очікується, і в кінцевому підсумку приносять користь, розробка зручності використання системи відіграє вирішальну роль у формуванні сприйняття якості її використання користувачами.

Однак, варто зазначити, що не бракує і альтернативних варіантів тлумачення зручності використання та взаємодії з користувачем.

Хоаші К. та інші у своїй роботі 2018 року стверджували, що usability є загальною концепцією, а отже, розпливчасто вільною або парасолькою (відсутня єдина дослідницька парадигма). Однак, вони вважають таку ситуацію незадовільну і у своїй роботі використовували концепцію usability, яка визначала точні атрибути концепції та забезпечувала перевірені інструменти для її вимірювання [3].

На основі цієї концепції Хертзум у 2010 році презентував шість типів usability (рис. 1.1) [4].

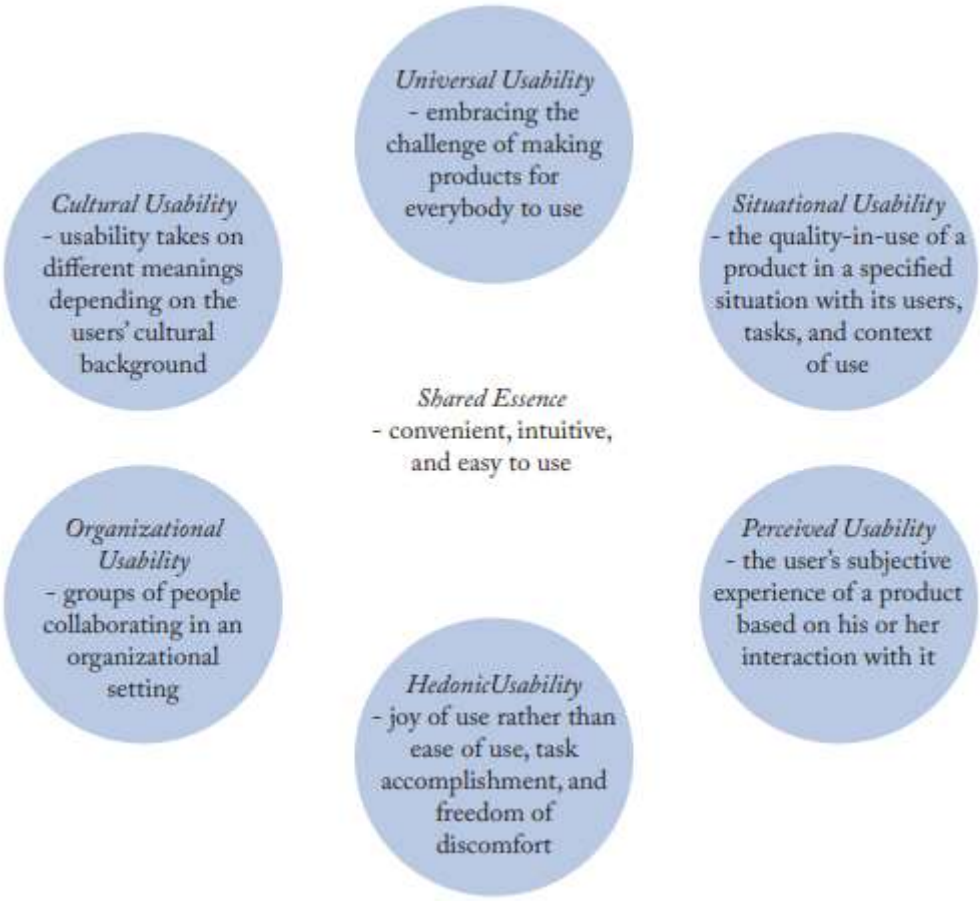


Рисунок 1.1 – Шість типів usability

Джерело: [4]

Універсальна зручність використання – це зображення зручності використання охоплює завдання створення продуктів для всіх. Здібності людей,

походження, особисті стилі, технологічне середовище тощо. Проте всім людям може знадобитися отримати доступ до інформації чи деяких інших можливостей, які надають вебсайти та інші продукти. Послідовне виключення певних груп людей із цих можливостей несумісне із загальними уявленнями про справедливе суспільство. Виключення значних груп людей із використання окремих продуктів може зменшити доцільність розробки цих продуктів. Тому важливою метою є розробка продуктів, придатних для використання всіма – ціль, що зовсім відрізняється від «визначених користувачів» у визначенні ISO 924-11.

Універсальність використання – це також велике завдання, оскільки воно передбачає, що продукти мають бути такими ж інклюзивними, наскільки люди різноманітні. Універсальна зручність використання є особливо актуальна у зв'язку з тестуванням продуктів загального призначення, систем швидкого доступу та використання різноманітних вебдодатків, таких як електронна комерція, електронний уряд та електронна охорона здоров'я.

Ситуаційна зручність використання відповідає визначенню зручності використання ISO 9241 [5]. Цей тип usability заснований на передумові, що користувачі відчувають продукти не окремо, а як частину ситуації (контексту). Отже, usability слід розуміти стосовно конкретних ситуацій з їх користувачами, завданнями та ширшим контекстом використання. Простіше кажучи, деталі ситуації використання є обов'язковими для визначення придатності продукту.

Однак, ця розміщеність переважає загальні принципи зручності використання. Дрейпер вважає ситуаційний образ зручності використання песимістичним, оскільки він відкидає узагальнення за межами конкретних ситуацій використання. Хоча це значною мірою означає, що універсальне використання є недосяжною метою, це узгоджується з основними принципами маркетингу такими як «знай свого користувача» [6]. Ситуаційна зручність безпосередньо застосована до тестування продуктів, які розроблені з урахуванням конкретних ситуацій використання, а не для ринку різноманітних клієнтів і ситуацій використання. До таких продуктів належать продукти на замовлення, які

замовлені клієнтом і створені на замовлення або налаштовані відповідно до конкретної ситуації клієнта.

Уявна зручність використання стосується суб'єктивного досвіду користувача продукту. Сприймання зручності справді орієнтоване на користувача, на відміну від орієнтованого на використання, оскільки воно робить окремого користувача остаточним арбітром щодо usability. У цьому образі немає особливого акценту на задоволенні, а лише фокус на суб'єктивних оцінках, на відміну від показників ефективності. Намір людей використовувати продукт значною мірою залежить від їх сприйняття корисності, простоти використання та задоволення, пов'язаного з використанням продукту.

Ці три суб'єктивні критерії приблизно відповідають концепції ефективності та задоволення, розглянутої вище, однак вони демонструють, як usability може виходити за межі задоволення та, наприклад, впливає на намір користувачів використовувати продукт, їхні способи взаємодії з продуктом і майбутні рішення про покупку. Уявлення про зручність використання особливо важливо під час оцінки продуктів, використання яких є дискреційним.

Використо́вність usability відповідає радість, а не простоту чи ефективність використання. Використо́вність usability пов'язане з компонентом задоволення у визначенні usability ISO 9241, але хоча цей компонент здається упередженим у бік уникнення негативних емоцій, використо́вність usability – це лише створення позитивних емоцій. Це розрізнення є важливим, оскільки якості, які допомагають уникнути негативних емоцій, відрізняються від тих, які викликають позитивні емоції.

Використо́вність usability подібне до сприйнятого usability своєю зосередженістю на суб'єктивному досвіді, але воно відрізняється від інших образів винятковою зосередженістю на задоволенні та емоціях. Використо́вність usability особливо актуальна для оцінки побутової техніки, онлайн-ігор, соціальних медіа та інших продуктів, які передбачають інтерактивний досвід або самовираження. Крім того, використо́вність usability актуальна для електронної

комерції, оскільки наявність використо́вність якостей впливає на рішення про покупку.

Організаційна зручність передбачає співпрацю груп людей в організаційному середовищі. На жодному з чотирьох попередніх типів-образів не згадується співпраця чи організації, хоча сайтів та вебдодатків, які використовують як внутрішні інформаційні системи велика кількість.

Нерівномірний розподіл роботи та переваг у таких системах (керівництво та підлеглі) означає, що різні групи користувачів можуть сприймати продукт зовсім по-різному. Елліотт і Клінг (1997) пропонують оцінювати функціональність організації на трьох рівнях [7]:

- відповідність користувача продукту;
- відповідність організації продукту;
- відповідність середовища продукту.

Другий і третій рівні, зокрема, визнають, що на організаційну зручність використання впливають способи, якими продукти роблять деякі компетенції застарілими, перенаправляють інформацію, створюють нові ролі тощо. Цей образ зручності використання має відношення до оцінки продуктів, які варіюються від групового програмного забезпечення, яке використовується організаційними підгрупами на власний розсуд, до корпоративних систем, якими щодня користуються всі співробітники.

Культурна зручність – цей тип підкреслює, що usability набуває різного значення для користувачів із різним культурним походженням. Культурне usability можна визначити як ступінь, до якої комп’ютерна система, особливо в міжкультурному контексті використання, відповідає культурному фону її користувачів, таким чином, що вона підтримує їхню діяльність ефективно, результативно та приємно. Багато елементів продукту можуть залежати від культури, включаючи кольори, графіку, мову та макет.

Так, наприклад, у США червоний колір асоціюється з небезпекою, а в Китаї – із щастям; в Єгипті колір щастя асоціюється з жовтим, що у Бразилії вказує на

боягузтво. Окрім таких відмінностей на рівні інтерфейсу, культурне походження людей впливає на їхні когнітивні процеси – те, як вони пізнають світ. Ці відмінності в когнітивних процесах створюють міжкультурні відмінності в тому, що становить корисний дизайн. Культурна зручність особливо актуальна для оцінки продуктів для міжнародної аудиторії, включаючи сайти.

Так, попри спільну сутність, ці типи (образи) відрізняються за світоглядом і перспективою. Кожне зображення пропонує інший напрямок, у якому слід дивитися, і надає лише часткове уявлення про зручність використання. Лише використовуючи всі типи одночасно, можливо повністю оцінити досвід користувача.

Повертаючись до традиційних концепцій варто зазначити, що у 1991 році Організація зі стандартизації (ISO), у відповідь на потребу співтовариства програмного забезпечення стандартизувати деякі аспекти вебсайтів, оприлюднила стандарт 9126, який визначає зручність використання як набір атрибутів програмного забезпечення, які стосуються зусиль, необхідних для використання, і на індивідуальній оцінці такого використання заявленою або неявною групою користувачів [8]

Потім, у 2001 році, ISO переробила визначення usability в стандарті ISO 9241-1, яке фактично стверджує, що зручність використання – це «ступінь, до якої продукт може використовуватися певними користувачами для досягнення визначених цілей з ефективністю, результативністю та задоволенням у визначений контекст використання» [9].

Інше визначення зручності використання можна знайти в ISO/IEC 25010, який замінив стандарт ISO/IEC 9126 від 2001 року, і визначає зручність використання як «ступінь, до якої продукт або система можуть використовуватися певними користувачами» досягти визначених цілей з ефективністю, результативністю та задоволенням у визначеному контексті використання» [10].

Таким чином, з моменту створення першого офіційного визначення usability можна стверджувати, що велика кількість атрибутів usability була взята до уваги щодо можливості використання конкретних програмних продуктів, починаючи від монолітних систем і закінчуючи класичними вебсторінками. З огляду на атрибути зручності використання, які сприяють якості програмного забезпечення для настільних комп'ютерів та мобільних пристрої, показуючи, що найбільш поширеними є ефективність, задоволення, можливість навчання та ефективність.

1.2. Класифікація методів проведення оцінки usability сайтів

На основі вищенаведеного аналізу історії розвитку та підходів до визначення поняття usability, робимо висновок, що usability (зручність використання) – це дослідження перетину систем і користувачів, завдань і очікувань у контексті використання. Більшість досліджень оцінки usability концентруються на найкращому розумінні та об'єктивних методах вимірювання з метою охопити всі дійсні явища в одній системі чи моделі.

Однак, варто зазначити, що тестування зручності використання датується ще ранніми 1970-ми роками. Раннім і впливовим описом тесту зручності використання був методом роздумів вголос. По суті, usability-тест складався з користувача, який використовує продукт, розмірковуючи вголос, і оцінювача, який спостерігає за користувачем і прислухається до його думок. Це базове налаштування допускає численні варіації. На цьому етапі ми просто зауважимо, що тест usability складається з чотирьох основних компонентів.

У цій моделі користувачі взаємодіють із продуктом на основі набору інструкцій і завдань, підготовлених перед тестом. В інструкції є пояснення того, як думати вголос; завдання вказують, чого користувачі повинні намагатися досягти за допомогою продукту. Таким чином, завдання гарантують, що користувачі використовують продукт у конкретних деталях.

Розв'язуючи завдання, користувачі вербалізують свої думки – думають вголос. Вербалізація показує, як користувачі розуміють і відчувають продукт.

Якщо користувачі замовкають на тривалий період часу, їм пропонується відновити вербалізацію. Користувачів також можуть попросити пояснити, чому вони вагаються, чого вони очікують і як вони оцінюють свій досвід.

Оцінювач або група оцінювачів спостерігає за взаємодією користувачів із продуктом і прислухається до їхніх думок. На цій основі оцінювач аналізує, наскільки добре продукт підтримує користувачів у виконанні завдань. Результатом цього аналізу є ідентифікація, опис і звіт про ряд проблем зручності використання. Оцінювач також відповідає за встановлення ситуації, в якій користувач може використовувати продукт і відчуває себе вільним робити як позитивні, так і негативні коментарі. Чотири основні компоненти взаємопов'язані та передбачають, що оцінювач знайомий із продуктом та його (передбачуваним) використанням. Ці передумови означають, що тест на зручність використання охоплює попередній аналіз і проєктування, а також подальший повторний аналіз і перепроєктування. Таким чином, перевірка usability не проводиться окремо від процесу розробки.

Щоб зібрати всі необхідні дані для покращення якості окремих аспектів програмного забезпечення, було розроблено та емпірично протестовано різноманітні методи оцінки зручності використання (UEM) [11]. Одні з найбільш визнаних UEM стосуються сімейства методів тестування користувачів, зокрема:

- протоколу мислення вголос;
- протоколу запитання;
- вимірювання ефективності;
- аналіз журналів;
- відстеження очей;
- дистанційне тестування.

По-друге, методи перевірки, призначені для використання експертами стосуються:

- евристичної оцінки;
- когнітивного проходження;

- перспективної перевірки;
- огляду настанов.

По-третє, методи опитування, призначені для збору суб’єктивних даних від користувачів, використовують як кількісні (анкети), так і якісні (інтерв’ю та фокус-групи) методи.

Баст’єн пропонує загальну структуру для проведення тестів зручності використання вебсайтів шляхом обговорення існуючих методологій і атрибутів зручності використання [12]. Як виклики вони вказують на унікальні особливості вебсайтів і бездротових мереж загалом, які впливають на зручність використання, а саме:

- контекст;
- мультимодальність;
- підключення;
- розмір екрана;
- роздільна здатність дисплея;
- обмежені можливості обробки та потужність;
- обмежувальні методи введення даних.

У випадку дослідницьких методологій для тестування зручності використання вони вказують на контрольовані лабораторні експерименти та польові дослідження.

Тоді як обмеженнями для оцінки usability виступають:

- незнання контексту;
- відсутність достатнього контролю над учасниками дослідження та розв'язання таких питань, як вибір умов середовища, ефективність оцінки, збір даних і контроль стану.

Вони також ідентифікують дев'ять загальних атрибутів usability:

- легкість використання;
- ефективність;
- можливість запам'ятовування;

- помилки;
- задоволеність користувача;
- результативність;
- простота;
- зрозумілість (читабельність);
- ефективність навчання.

Фабіл у 2009 році представив нову ієрархічну модель GQM (метрика цільових запитань) для оцінки зручності використання вебсайтів. На найвищому рівні вони розміщують дві характеристики якості: ефективність і задоволення [13].

На середньому рівні концептуалізовано шість керівних принципів: простота, точність, витрачений час, характеристики, безпека та привабливість.

Зрештою, внизу з'являється відображення між запитаннями та показниками, що дає змогу збирати кількісні дані для оцінки зручності використання. Наприклад, гібридна модель для автоматизації оцінки зручності використання вебсайтів. Гібридний підхід поєднує два методи збору даних, а саме журналювання та ESM (метод вибірки досвіду). Перший базується на зборі даних, пов'язаних із взаємодією користувача з програмою. Використання датчиків, користувацьких даних, що дозволяє виконувати статистичний аналіз щодо зручності використання.

Другий заснований на зборі почуттів користувачів до конкретного продукту через запитання. Ці два методи відповідно використовуються для вимірювання ефективності та задоволеності.

Таким чином, кожен із наведених вище підходів є прикладом використання певної форми аналізу для формулювання або перевірки проєктних рішень. Водночас фактична поведінка вебкористувачів у вебпереглядачі створює численні помітні сліди даних, пов'язані з низкою різних змінних, висновки з яких пропонують багате джерело інформації про зручність використання. Такі дані про фактичний досвід користувачів на конкретному вебсайті необхідні для перевірки

та вдосконалення дизайну вебсайту, щоб збільшити простоту використання та бізнес-цінність.

1.3. Особливості оцінки зручності інтерфейсу на основі користувацьких логів

Сучасні інформаційні системи все більше використовують переваги персоналізації як життєво важливих засобів оптимізації взаємодії з користувачем, що серйозно ускладнюється як через зростаючу складність додатків, так і через постійне збільшення розміру доступного інформаційного простору. Попит на загальні адаптивні системи призводить до попиту на загальні методи та програмні засоби, які виконують автоматичне конструювання та підтримку індивідуальних моделей користувачів. Своєю чергою ці методи автоматичного збору характеристик користувача з мінімальною участю людини можливо успішно використовувати для персоналізованої навігації, рекомендацій вмісту та/або фільтрації в певному домені (наприклад, для співробітників чи наукових публікацій).

Одним з таких питань, які потребують подальшого дослідження є можливі критерії та джерела вказівок для досягнення, які стосуються планування та дизайну вебсайтів. Ключ до хорошого вебдизайну полягає в розумінні користувачів вебсайту та їхніх завдань.

Враховуючи, що більшість інструкцій з вебдизайну, як правило, представлені не в контексті цілей використання, а як універсали – більшість таких методів оцінки сумнівно базується на традиційних, а не емпіричних знаннях.

Серед найпопулярніших таких настанов – Якоб Нільсен. Нільсен, високоповажний дизайнер вебінтерфейсу користувача, надає цінні вказівки щодо вебдизайну у своїй щомісячній колонці «Alertbox». Він опублікував дві колонки «Десять найпоширеніших помилок у вебдизайні» і «Десять найпоширеніших нових помилок у вебдизайні», в яких описуються помилки, які зазвичай роблять

вебдизайнери. Серед помилок, яких варто уникати вебдизайнерам, можна віднести [14]:

- елементи сторінки, які знаходяться в постійному стані анімації;
- сторінки з довгим прокручуванням;
- нестандартні кольори посилань;
- довгий час завантаження;
- запуск нових вікон браузера;
- відсутність інформації про автора;
- переміщення сторінок на нові URL-адреси;
- дизайни, які виглядають як реклама.

Нільсен також запропонував «Десять добрих вчинків у вебдизайні», в яких він окреслив «обов'язки» вебдизайну. До них належать [15]:

- розміщення назви та логотипу на кожній сторінці;
- забезпечення пошуку на вебсайті з понад 100 сторінками;
- використання простих заголовків і назв сторінок;
- використання структури сторінки, яка полегшує сканування;
- надання сторінок, доступних для користувачів з обмеженими можливостями;
- спостереження за тим, що роблять великі вебсайти.

На додаток до порад, отриманих із принципів зручності використання та риторичних схем, невелика кількість емпіричних досліджень використання Інтернету також пропонує альтернативні дизайнерські рішення. Наприклад, коли користувачі шукають інформацію, вони зосереджені, а підходи до дизайну, які призначені для «серферів» (наприклад, реклама), відволікають шукачів інформації.

При цьому вісімдесят відсотків запитів від сервера зазвичай належать до типу http (на відміну від інших протоколів, таких як ftp). Крім того, користувачі віддають перевагу методу навігації за гіперпосиланнями, де гіперпосилання становлять 52% усіх запитів документів. Таким чином, значною мірою

гіперпосилання є кращим методом навігації вебсайту. На другому місці після гіперпосилань за популярністю серед користувачів була команда браузера «Назад», на яку припадає 41% усіх запитів документів. Тому користувачі зазвичай переміщуються двома рівнями сайту, перш ніж повернутися до точки, з якої вони зайшли на сайт. Відповідно, кнопка «Назад» була і є важливим інструментом навігації для користувачів [16].

Щодо реєстрації активності користувачів, то існує два основних підходи – стандартні логи (журнали) вебсерверу та логи на стороні клієнта, створені спеціальними програмними засобами, розгорнутими на комп'ютері користувача. Журнали вебсерверу зазвичай використовуються як вхідні дані для різних методів інтелектуального аналізу даних, результатами яких є здебільшого звичайні послідовні шаблони або кластери користувачів і сторінок. На жаль, журнали вебсерверу не надають достатньо інформації через механізми кешування веббраузерів і основні принципи протоколу HTTp, який є низькорівневим протоколом без збереження стану та без чітко визначеної семантики виконуваних дій.

Крім того, адаптивна вебсистема з динамічною генерацією сторінок може створювати два подібні або навіть однакові записи в журналі, що призводить до абсолютно різних відповідей системи через зміни в моделях користувача та домену. Щоб усунути деякі недоліки цього підходу та отримати інформацію про взаємодію лише на стороні клієнта (наприклад, наведення курсора на спливаючі підказки), необхідно використовувати методи, які можуть забезпечувати високий рівень деталізації, але становлять серйозну загрозу конфіденційності користувачів.

Тому використання стандартних клієнтських вебтехнологій, таких як JavaScript, видається більш придатним через обмежений обсяг роботи і більше сприйнятний користувачем. Таким чином, щоб бути корисними, логи часто мають відповідати різним вимогам. Тут ми перерахуємо найпоширеніші, які відповідають стандарту ISO 9126 від 2001 року:

- повнота: всі події зазначені в логах можна відновити з його журналу;

- ретроспективна незалежність – минуле та майбутнє не впливають на логи та можливості їх відновлення;
- точність – відновлення журналу трасування є однозначним: заданий журнал містить унікальний слід подій, які могли його створити;

Крім того, події в логах зазвичай мають атрибути (наприклад, дата, ім’я користувача, адреса).

Однак, з цих принципів є і виключення. Прикладом системи логів, яка не є ретроспективно незалежною, є, наприклад, Linux, яка записує ім’я користувача разом із призначеним псевдонімом (однак, варто зазначити, що в Linux псевдоніми використовуються для зручності, а не для збереження конфіденційності користувачів).

Існують також логи переходів, які надають інформацію про те, які вебсторінки, як із самого сайту, так і з інших сайтів, містять посилання на документи, що зберігаються на сервері. Журнал містить таку інформацію, як URL-адреси сайтів і сторінок на сайтах, які спрямовували відвідувачів на певну сторінку. Наприклад, користувачі часто можуть потрапляти на певний вебсайт через пошукову систему, а пошукову систему, яка спрямовує, разом із ключовими словами, використаними в початковому запиті, можна отримати з журналу переходів.

Користувацькі логи зберігаються, як правило, у загальному форматі файлу журналу або в розширеному форматі файлу журналу. Загальний формат файлу журналу містить дату (дата, час і часовий пояс запиту), IP-адресу клієнта (IP-адреса віддаленого вузла та/або запис DNS), ім’я користувача (ім’я віддаленого журналу користувача), передані байти, ім’я сервера, запит (URI). запит) і статус (повернено код статусу http). Розширений формат Logile включає надіслані та отримані байти, сервер (ім’я, IP-адреса та порт), запит (запит URI та стовбур), ім’я запитуваної служби, час, необхідний для завершення транзакції, використовувану версію протоколу передачі, агент користувача (програма

здійснення запиту, як-от браузер Google Chrome або «павук» пошукової системи), ідентифікатор cookie та реферер.

Інструменти реєстрації вебсервера, також відомі як аналізатори вебтрафіку, аналізують файли журналу вебсервера та створюють звіти з цієї інформації (HTTP-ANALYZE). Журнали вебсервера є вимірюванням вебсервісів на основі користувачів і можуть використовуватися, щоб «почати розуміти шлях, який проходять користувачі через сервер, проблеми, з якими вони стикаються під час сеансу, і технології, які користувачі використовують під час навігації сайт». Вони також вважають, що ці дані можна використовувати при плануванні та проектуванні вебсайтів.

При цьому дані журналу сервера розділені на три групи:

- навігація та діяльність;
- демографічні дані;
- продуктивність.

Дані логів навігації та активності сервера надають інформацію про аспекти взаємодії користувача з вебсайтом, наприклад, шляхи навігації, кількість звернень, час, проведений на сторінці тощо.

Різні інструменти журналювання сервера надають масив інформації про дії користувачів на вебсайті. Інформація, зібрана цими інструментами, і терміни, що використовуються для опису даних, не завжди узгоджуються між серверними інструментами журналювання. Крім того, логи користувачів широко використовуються в різноманітних завданнях для забезпечення надійності, оскільки вони часто є єдиними доступними даними, які записують інформацію про час виконання того чи іншого елементу програмного коду. Логи також відіграють незамінну роль у прийнятті рішень на основі даних у бізнесі.

Однак, існує багато проблем, пов’язаних з повнотою, точністю та репрезентативністю даних журналу сервера. Зокрема, до них належать кешування та унікальна ідентифікація користувача.

Так, веббраузер може надсилати так званий «умовний запит» вебсерверу. В умовному запиті браузер запитує документ або вбудований об’єкт із сервера, лише якщо сторінка ще не збережена в «дисковому кеші» браузера. Цей спосіб зменшує мережевий трафік. Однак, з точки зору журналювання вебсервера, сторінки, які обслуговуються з кешу браузера, не реєструватимуться в журналі вебсервера. Таким чином, дані користувача не будуть зафіксовані в цій ситуації.

Водночас, проксі-сервери, як використовуються постачальниками послуг Інтернету, а також приватними та державними установами з великою базою користувачів, щоб захистити мережу від неавторизованих сторін та/або зменшити мережевий трафік. Щоб зменшити мережевий трафік, сторінки, які запитуються та завантажуються в браузер через проксі-сервер, зберігаються в «дисковому кеші» проксі-сервера. Ідея полягає в тому, що документи, які часто запитуються користувачами, можуть бути доступні з кешу проксі-сервера, а не з вебсервера, де документ спочатку знаходився. Як і у випадку з кеш-пам’яттю браузера, з точки зору журналювання вебсервера, сторінки, які обслуговуються з кешу проксі-сервера, не записуватимуться в журнал вебсервера. Тому дані користувача в цій ситуації не будуть зафіксовані.

Таким чином, з журналів сервера можна отримати усього два відносно надійних типи інформації:

- кількість отриманих звернень;
- кількість унікальний користувачів.

На основі вищезазначеної інформації можемо зробити висновок, що логування – це завдання створення операторів журналювання з належним описом і необхідними програмними змінними та вставлення операторів журналювання в потрібні позиції у вихідному коді. Логування привернуло увагу як наукових кіл, так і бізнесу, оскільки журналювання є фундаментальним кроком для всього подальшого видобутку інформації.

Механізм журналювання – це набір операторів журналювання та код їх активації, реалізований розробниками або даною програмною платформою.

Найбільш поширений шаблон кодування, який використовується для активації журналювання оператори if (умова) then log error().

Водночас, щоб підвищити гнучкість, бізнеси часто використовують бібліотеки журналювання, які є компонентами програмного забезпечення, які полегшують журналювання та надають розширені функції (наприклад, безпеку потоків, конфігурацію архіву журналу та розділення API). З цією метою було розроблено багато відкритих бібліотек журналювання (наприклад, Log4j, SLF4J, AspectJ, spdlog тощо).

Таким чином, логи загалом – це напівструктурований текст, надрукований операторами журналювання (наприклад, printf(), logger.info()) у вихідному коді. Наприклад, на рис. 1.2 ми можемо побачити як два повідомлення журналу друкуються двома операторами журналювання у вихідному коді.

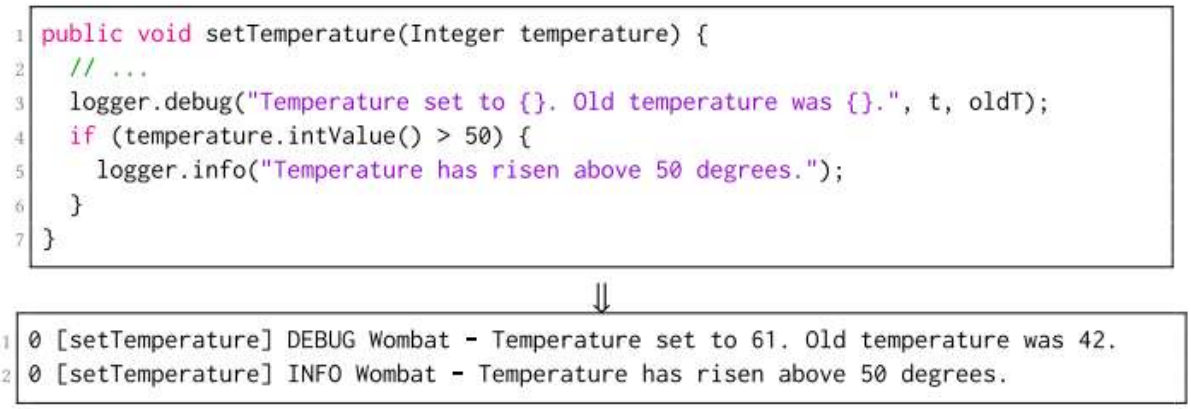


Рисунок 1.2 – Структура журналювання SLF4J

Джерело: [17].

Перші кілька слів повідомлень журналу визначаються відповідною структурою журналювання (наприклад, SLF4J), і вони мають структуровану форму. Решта слів (наприклад, «50 градусів») є неструктурованими, оскільки вони написані користувачами для опису конкретних подій під час виконання системи.

1.4. Переваги та недоліки використання користувацьких логів для проведення оцінки usability вебсайтів

Традиційна практика журналювання (наприклад, які змінні друкувати) в основному покладається на знання домену розробників. Під час виконання системи журнали програмного забезпечення збираються та стискаються як звичайні файли за допомогою інструментів стиснення файлів (наприклад, WinRAR). Крім того, розробники використовують зібрані журнали для виявлення аномалій. Десятки років тому ці процеси базувалися на певних правилах, визначених розробниками. Наприклад, щоб отримати певну інформацію, пов’язану із завданням (ідентифікатор потоку), розробникам потрібно було розробити правила регулярних виразів для автоматизованого аналізу журналу. Традиційний процес виявлення аномалій також спирався на правила, створені вручну. Ці методи аналізу логів були ефективними на початку, оскільки більшість широко використовуваних програмних систем були невеликими та простими. Однак, оскільки сучасне програмне забезпечення навіть самого простого сайту стало набагато більшим за масштабом і складнішою за структурою, традиційний аналіз логів, який в основному базується на спеціальних знаннях домену або вручну створених і підтримуваних правилах, стає неефективним і неможливим.

Це створює чотири основні проблеми для сучасного аналізу та оцінки зручності веб сайтів на основі даних користувачів, а саме:

1. У багатьох практиках, хоча кілька старших розробників у тій самій групі поділяють деякі найкращі практики журналювання, більшість нових розробників з інших проектів пишуть заяви про журналювання на основі знань домену та спеціальних проектів. Як наслідок, якість журналів виконання значною мірою змінюється.

2. Обсяг логів програмного забезпечення швидко зріс, тому набагато складніше вручну сформулювати правила (наприклад, шаблони подій журналу або аномальні моделі).

3. З поширеністю вебсервісів і платформ для обміну вихідним кодом (наприклад, Github) програмне забезпечення можуть писати тисячі розробників з усього світу. При цьому розробники, які мають підтримувати правила, часто не мають уявлення про початкову мету журналювання, що ще більше ускладнює мануальне обслуговування їхніх правил.

4. Через широке впровадження концепції гнучкої розробки програмного забезпечення (Agile) нова версія програмного забезпечення часто з'являється в дуже короткий термін. Таким чином, відповідні заяви журналювання також часто оновлюються (наприклад, сотні нових операторів журналювання на місяць).

Щоб вирішити ці проблеми, за останні десятиліття як теоретиками так і практиками було виконано величезну роботу. Зокрема, починаючи з 2003 року, ряд дослідницьких зусиль було спрямовано на автоматизовану конструкцію правил і витяг критичної інформації з журналів програмного забезпечення, включаючи перші фрагменти роботи з розбору журналів, виявлення аномалій і прогнозування збоїв. Хатонен та інші запропонував першу методику стиснення, розроблену спеціально для логів. Пізніше було проведено багато емпіричних досліджень як перші кроки до деяких складних проблем автоматизованого аналізу журналів, включаючи перше дослідження діагностики помилок [18].

Крім того, сучасні алгоритми машинного та глибокого навчання були широко прийняті в найсучасніших документах (наприклад, Deeplog) для виявлення та аналізу аномалій [19]. Крім машинного навчання, розпаралелювання використовувалося в різних нещодавніх роботах, таких як Logzip для стиснення журналів [20]. Ці масштабні дослідження автоматизованого аналізу журналів у кількох основних напрямках значною мірою підвищили ефективність і доступність систематичного використання блогів для аналізу usability сайтів.

У цьому контексті також варто згадати про інструменти робототехнічної автоматизації процесів (RPA), які дозволяють автоматизувати процедури, записуючи сценарії та кодувати послідовності взаємодії з вебпрограмами та програмами (наприклад відкриття файлу, вибір поля у формі або клітинки в

електронній таблиці та копіювання даних між полями/комірками). Ці методи використовують логи користувачів як вхідні дані [21].

Відповідно, кожен лог (рутинна траса) складається з послідовності взаємодій (далі по тексту скорочено називаються діями). Кожна дія має тип (наприклад, вибрати, скопіювати, вставити тощо) і набір параметрів (наприклад, ідентифікатор елемента інтерфейсу користувача, над яким виконується дія, а також вхідні та вихідні дані дії). За наявності журналу інтерфейсу користувача RPA виводить набір рутинних специфікацій.

Специфікація процедури – це основа, що складається з умови активації та послідовності специфікацій дій. Специфікація дії своєю чергою складається з типу дії та набору функцій для обчислення параметрів дії з параметрів попередніх дій.

Метод починається зі стиснення журналу інтерфейсу користувача в детермінований ациклічний кінцевий автомат (DAFSA). Потім він застосовує алгоритм для декомпозиції до зв'язних графів на регіони Single-Entry Single-Exit (SESE). Деякі з цих регіонів відповідають послідовності дій. Для кожної такої послідовності метод перевіряє, чи кожна дія є детермінованою. Якщо так, він намагається виявити умову активації за допомогою методу аналізу правил.

З іншого боку, якщо дія в середині послідовності не є детермінованою, послідовність розбивається на підпослідовності, для яких метод намагається окремо виявити умови активації. Специфікація процедури генерується для кожної (під)послідовності, для якої знайдено умову активації. Таким чином, ці масштабні дослідження автоматизованого аналізу логів у кількох основних напрямках значною мірою підвищили ефективність оцінки usability сайтів.

У табл. 1.1. представлено сценарій із реального життя, що є прикладом використання логів користувачів для автоматизації різноманітних процедур. Цей приклад натхненний роботою, виконаною групою покращення послуг Мельбурнського університету, яка застосовує RPA для автоматизації

різноманітних процесів, пов’язаних із студентами, наприклад прийому абітурієнтів.

Дані студентів зберігаються як у системі управління студентами (доступною через вебінтерфейс), так і в локальних файлах Excel для цілей резервного копіювання.

Табл 1.1 – Відстеження та автоматизація на основі аналізу логів користувачів, студентський сценарій

	Дія	Параметр 1	Парметр 2	Парметр 3
1	Натиснути кнопку	Ціль: Інтернет	Атрибут: Студенти	
2	Заповнити текстове поле	Ціль: Інтернет	Атрибут: ID студента	
3	Натиснути клавішу	Ціль: Інтернет	Атрибут: ENTER	Значення: 010234
4	Натиснути кнопку в рядку	Ціль: Інтернет	Атрибут: Оновити	Ряд ID: 010234
5	Заповнити текстове поле	Ціль: Інтернет	Атрибут: Адреса	Значення: 19 Parkville St, Burnley VIC 3121
6	Заповнити текстове поле	Ціль: Інтернет	Атрибут: Країна	Значенян: Австралія
7	Відкрити файл	Ціль: Excel	Назва: 010234	Path: C:/Students/Australia/
8	Копіювати	Ціль: Інтернет	Від: Адреса	Значення: 19 Parkville St, Burnley VIC 3121
9	Вставити	Ціль: Excel	Рядок: 5	Колонка: A
10	Зберегти як	Ціль: Excel		

11	Натиснути кнопку	Ціль: Інтернет	Атрибут: Підтвердження бекапу	

Джерело: складено автором за [22].

Витяг фіксує послідовність дій, які виконує один користувач. Так, співробітник університети уже ввійшов у систему управління студентами, і починає завдання, натиснувши кнопку студентів у вебінтерфейсі. Потім він вводить ідентифікатор студента в текстове поле ID студента та натискає клавішу введення для підтвердження. У вебінтерфейсі відображається список студентів, включно з тим, кого шукали. Біля кожного запису студента доступні дві опції: оновити та відкрити. Співробітник натискає на кнопку оновлення, оскільки він має намір оновити дані про місце проживання студента. Відкривається нове вікно з інформацією про студента, включаючи дані про місце проживання, тобто адресу та країну. Співробітник вводить нову адресу та країну, а у нашому випадку відкриває відповідний резервний файл Excel, щоб скопіювати частину адреси (лише вулицю) з вебінтерфейсу до Excel файл. Після чого файл змінюється, а співробітник підтверджує оновлення у вебінтерфейсі, натиснувши кнопку підтвердити резервне копіювання.

Цей приклад показує, що певне завдання (наприклад, оновлення даних про проживання студента) може виконуватися за допомогою різних послідовностей дій (процедур). Для автоматизації завдання потрібно ідентифікувати межі кожної процедури з логів інтерфейсу користувача, а в межах цих процедур визначити, які дії є детермінованими та можуть бути автоматизовані. Таким чином, надавши як вхідні дані журнал інтерфейсу користувача, на першому кроці аналізується журнал інтерфейсу користувача в DAFSA.

На другому кроці кожна з потенційних автоматизованих процедур аналізується шляхом перевірки того, чи кожна з її дій є детермінованою чи ні, тобто ця дія може бути виконана систематичним способом за допомогою сценарію RPA (наприклад, програмного бота). Результатом другого кроку є набір

специфікацій, що складається з: дії та набору функцій для автоматичного визначення всіх значень параметрів дії.

Кінцевим результатом зазначеного підходу є набір рутинних специфікацій. Кожна специфікація підпрограми складається з умови активації для автоматизації підпрограми і послідовність специфікацій дій.

Є кілька документів, де подібний підхід використовується як основа для тестування зручності використання вебсайтів загалом. Наприклад, використовувати користувацькі логи для віддаленого тестування usability сайтів [23].

Дистанційне тестування usability – це метод, при якому оцінювач не спостерігає безпосередньо за поведінкою учасника, а лише за результатами тесту. Результати в нашому випадку – це журнали, записані у вигляді послідовності клацань миші та натискань клавіш, пов’язаних із певним завданням. Журнали будуть показувати, які вікна відкривав учасник і які дії виконував. Для віддаленого використання модель завдання може бути дуже деталізованою. Подібна модель завдання також підходить для якісних тестів на usability.

Модель також може бути використана для проектування інтерфейсу користувача, що призводить до більш зручного програмного забезпечення внаслідок візуалізації анотацій, завдань та події, щоб допомогти спостерігачам отримати загальне уявлення про результати тестування зручності використання.

З іншого боку, у автоматизованої оцінки usability на основі користувацьких логів є деякі недоліки. По-перше, немає систематичного представлення завдань, тому складніше відокремити кожне завдання. Другий недолік полягає в тому, що за один раз можна спостерігати лише анотації від одного учасника. Усунення цих двох недоліків суттєво допоможе оцінювачу або групі оцінювачів у спостереженні за динамікою тесту. Результат інтерпретації журналу у подібній моделі доступний у вигляді шкали часу (рис. 1.3).

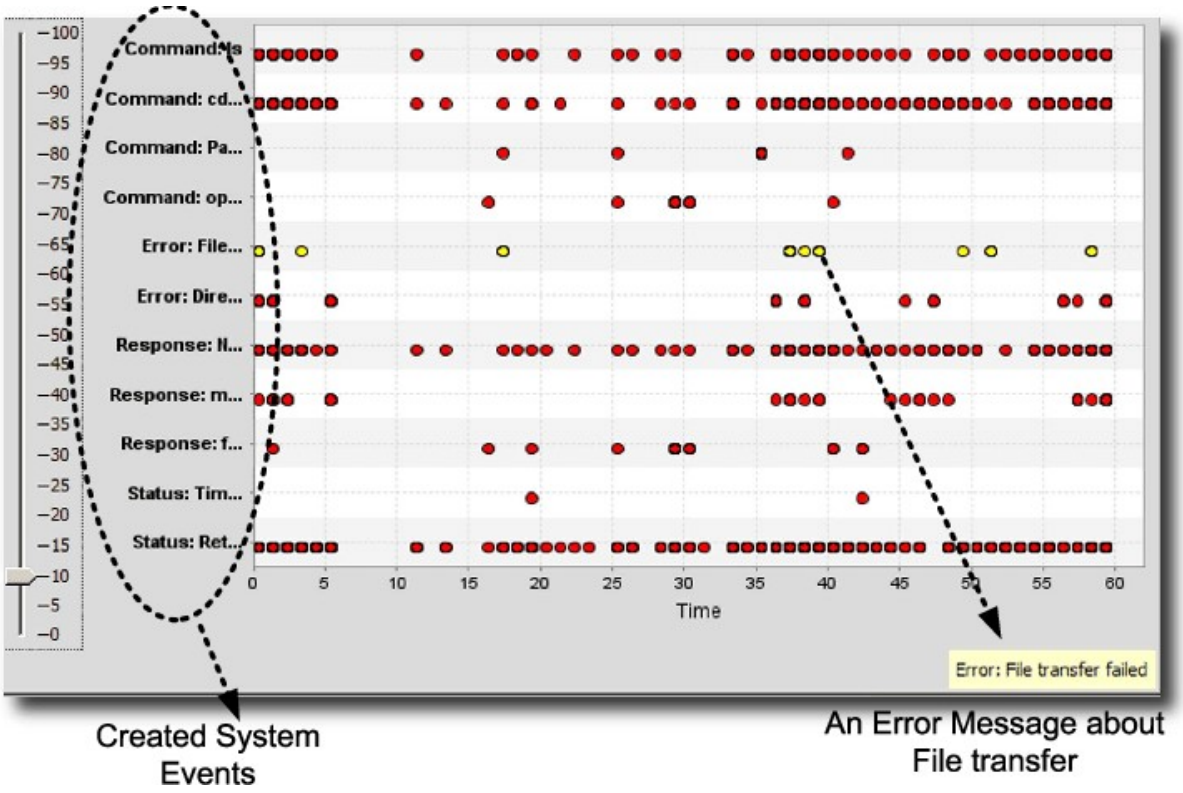


Рисунок 1.3 – Шкала часу для візуалізації користувацьких логів

Джерело: [24].

Текстова форма журналу (надається у структурній формі – XML) автоматично перетворюється на горизонтальну лінію (шкала часу), де окремі підзавдання можуть бути легко розпізнаними, включаючи інформацію про необхідний час і проблеми, які виникли в деяких конкретних завданнях. Ці проблемні частини позначаються за допомогою кольорових прямокутників (також званих гліфами), які відповідають анотаціям. Часові шкали або подібна візуалізація залежних від часу даних є інтуїтивно зрозумілими та доступними до інтерпретації.

Крім того, перетворення даних можна використовувати для виявлення нової інформації, прихованої в класичних тестових спостереженнях. Спостерігачі можуть спостерігати за кореляція між певними помилками в режимі реального часу та переглядати динамічні аспекти тесту, такі як залежність тривалості

виконання завдання від кількості помилок або появи однакових помилок між учасниками.

Безпосередньо інструменти автоматизованої оцінки usability являють собою різноманітні модулі, які разом підтримують повну перевірку зручності використання, але при цьому можуть функціонувати автономно. Наприклад, модуль завдання використовується для підготовки моделей завдань. Модуль журналювання анотацій використовується для створення анотацій під час тестування, а модуль візуалізації – графічно відображає зібрані логи для спостерігачів.

Подібні модулі дозволяють показувати результати кількох учасників одночасно. Цю функцію можна використовувати для пошуку схожої поведінки учасника. Завдяки використанню моделі задач ми можемо чітко розрізняти задачі, оскільки кожна з них представлена власним графічним елементом. Цей функціонал дозволяє спостерігачеві глибше аналізувати кожне завдання для всіх учасників, оскільки йому не потрібно шукати завдання кожному учаснику.

Щоб проаналізувати результати автоматизованого тесту, оцінювач повинен проаналізувати сеанси даних. Проблеми, що були знайдені на цьому, визначаються шляхом повністю або частково, шляхом аналізу саме даних без необхідності комунікації з користувачами.

І хоча наведені вище інструменти підтримують автомазивону оцінку usability-тесту, у багатьох аспектах є місце для подальшого розвитку. Один із них полягає в тому, як реєструються анотації та особливо категорії анотацій. Відомо, що анотації зазвичай реєструються після виконання учасником дії. Цей дрейф часу невеликий і майже постійний, і його навряд чи слід пропускати. Але категорія анотації, яка іноді є важливішою, повинна реєструватися швидше. Це має особливо допомогти, коли на часовій шкалі представлено два рівні ієрархії.

Другий аспект – це фільтрація анотацій. Іноді трапляються деякі анотації, такі як збій програми або подібні порушення, які погано впливають на динаміку тесту, тому їх слід видалити з візуалізації. Деякі зміни можна також внести в

часові шкали та анотації. Зазвичай події, які тривають лише кілька секунд (зазвичай автоматично зібрані дані), важливіші за події, які тривають довше. Іноді важливо також знати тривалість підзавдань, тому сьогодні питання візуалізації користувацьких логів для подальшого аналізу даних залишається відкритим питанням та актуальною темою досліджень.

2. РОЗРОБКА МЕТОДУ ОЦІНКИ USABILITY НА ОСНОВІ КОРИСТУВАЦЬКИХ ЛОГІВ

2.1. Аналіз інструментів для збору та оцінки інформації про дії користувача сайту

На сьогоднішній день існує багато інструментів для покращення процесів підтримки сайтів, оскільки не потрібно постійно слідкувати за рейтингом сайту чи кількістю помилок на ньому, на даний момент це все в більшості автоматизовано і службам технічної підтримки чи розробникам не потрібно витрачати на ці задачі багато часу. За допомогою більшості інструментів можна генерувати лог файл, який відображає певні показники ваб-ресурсу, цікаві для подальшого аналізу. В загальному сенсі, лог файл – це створений комп'ютером файл даних, який надає інформацію про шаблони використання, дії та операції операційної системи, програми, сервера або іншого пристрою. Лог файл містить в собі багато корисної інформації, як наприклад помилки на сервері, які дії були виконані на сайті, кількість кліків, подій, переходів і багато іншого. Розглянемо кілька існуючих інструментів, що дозволяють відстежувати та аналізувати дії користувачів на веб-сторінках, зокрема це:

- Google Analytics [41]
- Mixpanel [42]
- Fullstory [43]
- Mouseflow [44]

Підключення інструментів до сайтів чи окремих сторінок сайтів відбувається за схожою схемою. Більшість з них мають передбачають доступ до підключення сайту та інструментів аналітики тільки для зареєстрованих та авторизованих користувачів. Можуть бути присутні платні опції, які визначаються обраним тарифним планом (крім безкоштовного сервісу Google

Analytics). Практично кожен інструмент має засоби для зручного відображення логів та можливості для вивантаження даних у файл визначеного формату.

2.1.1 Google Analytics

Google Analytics - один з кращих безкоштовних інструментів, який може використати будь-яка людина для відстежування і аналізу даних, які отримані з веб-сторінок. За допомогою цього інструменту можливо визначити ключові слова, які притягають найбільше людей на веб-сторінки і які аспекти дизайну негативно впливають на досягнення мети. Так само є функціональна можливість генерувати звіти, які включають дані про джерела трафіка, поведінку відвідувачів, їх цілі, зміст та електронну комерцію. Інтерфейс даного сервісу зображено на рисунку 2.1.

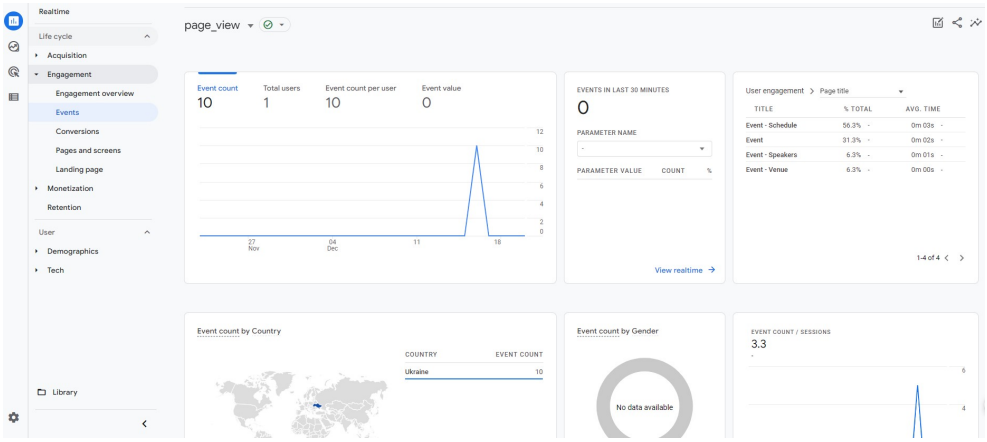


Рисунок 2.1 – Інтерфейс веб-сервісу Google Analytics

На рисунку 2.2 відображено дії, які потрібно зробити, щоб підключити сайт до Google Analytics. Перш за все, після входження в акаунт потрібно відкрити меню і натиснути на “Admin” далі вибрати “Data Streams”, далі додати URL свого сайту, перейти по “View tag Instructions” і побачити там JS код який потрібно вставити на сайт.

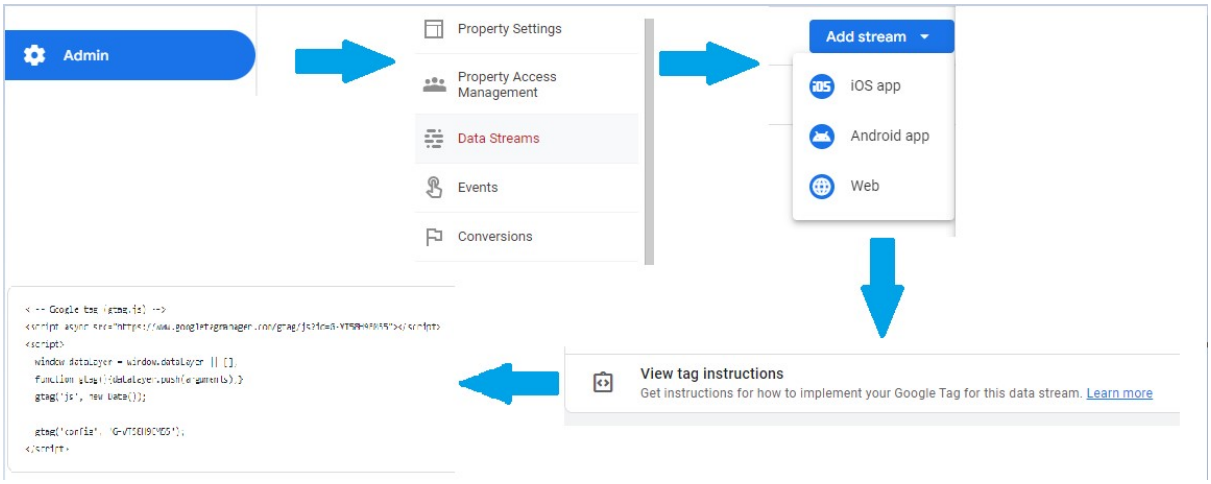


Рисунок 2.2 – Інструкція додавання сайту для Google Analytics

Структура лог файлу, який генерує Google Analytics представлена на рисунку 2.3.

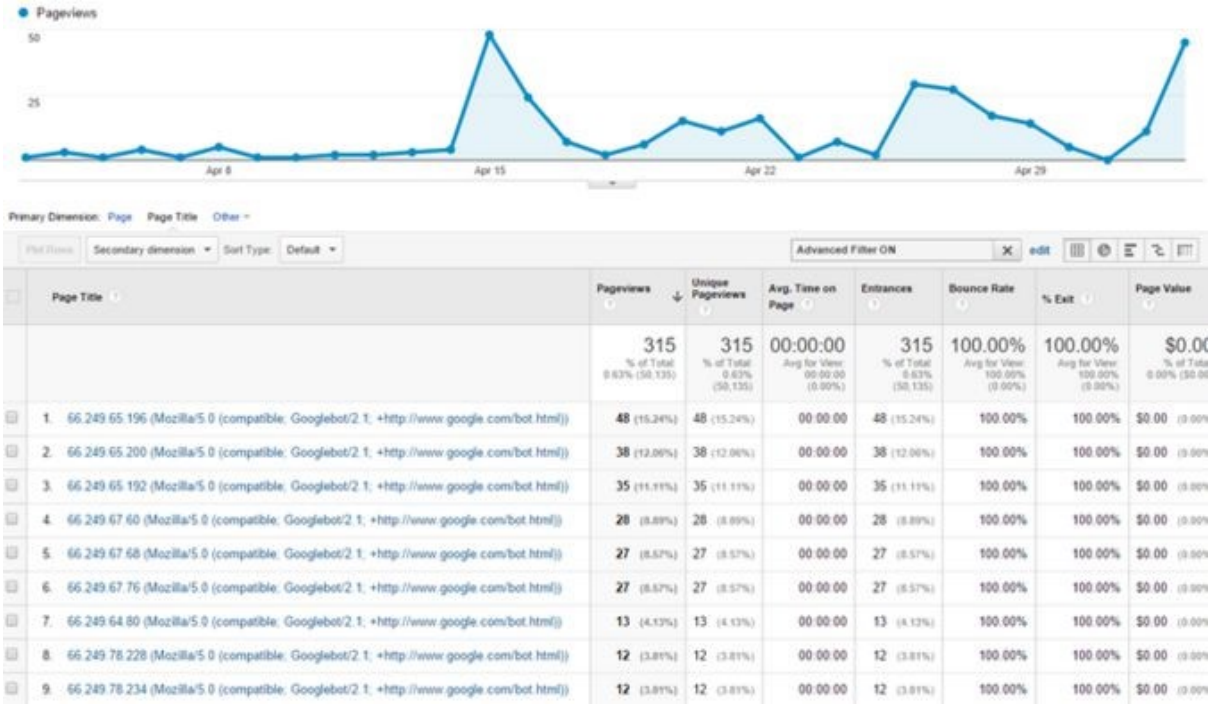


Рисунок 2.3 – Структура лог файлу Google Analytics

Лог файл, згенерований за допомогою Google Analytics, включає інформацію про саму сторінку яку використоує користувач (IP, браузер, силка) сторінки, кількість переглядів, середня кількість переглядів, середній час перебування на сторінці, процент людей які покинули сайт.

2.1.2 Mixpanel

Mixpanel – це інструмент аналітики продукту, який дає змогу отримувати дані про те, як користувачі взаємодіють із цифровим продуктом. Крім того, Mixpanel дає змогу аналізувати ці дані про продукт за допомогою простих інтерактивних звітів, які дозволяють запитувати та візуалізувати дані лише кількома кліками миші. Це рішення для відстеження на основі подій дає уявлення про те, як найкраще залучати, конвертувати та утримувати користувачів на веб-сайтах і мобільних платформах. Mixpanel дозволяє командам вчитися на даних користувачів і швидко впроваджувати інновації у створенні виграшних продуктів. Інтерфейс даного сервісу зображено на рисунку 2.4.

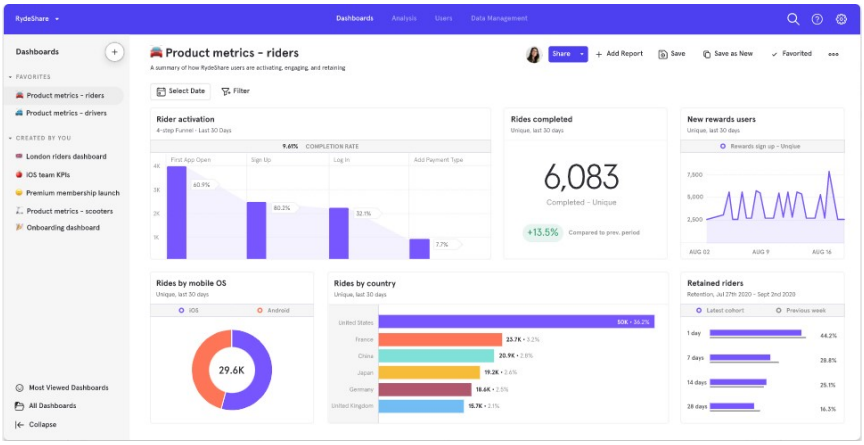


Рисунок 2.4 – Інтерфейс веб-сервісу Mixpanel

Структура лог файлу, який генерує Mixpanel, представлено на рисунку 2.5.

Лог файл, згенерований за допомогою Mixpanel, включає інформацію про подію яку виконав користувач на сайті, а також час затрачений на цю подію, спеціальний ідентифікатор, назва міста де перебуває користувач, код країни, платформа з якої заходили.

```
{
  "event": "Tutorial Complete",
  "properties": {
    "time": 1595055569,
    "distinct_id": "cbbe017b-cb88-5413-9785-29b9a00619c1",
    "$city": "Pittsburg",
    "mp_country_code": "United States",
    "$os": "OSX"
  }
},
{
  "event": "Download Page",
  "properties": {
    "time": 1595055560,
    "distinct_id": "56cf822c-7944-5095-9540-94b389cd0982",
    "$city": "Salvador",
    "mp_country_code": "Brazil",
    "$os": "Android"
  }
},
```


Рисунок 2.5 – Структура лог файлу Mixpanel

2.1.3 Fullstory

Fullstory – це інструмент, який легко можна налаштувати та за допомогою якого можна детально відстежувати кожен рух відвідувача на веб-сайті. FullStory забезпечує детальні записи сеансів із піксельною точністю та інші інструменти аналітики, які допоможуть краще зрозуміти клієнтів сайту. Результати включають карту кліків, перевірки елементів за допомогою «наведіть і кікніть» та інтеграцію з іншими програмами, які можна використовувати, як-от Google Analytics, Trello та Snap. Page Insights надає дані в реальному часі про те, що приваблює користувачів, а що ні. Існує безкоштовна обмежена версія. Також можна вибрати статус Pro або Enterprise. Інтерфейс сервісу Fullstory зображено на рисунку 2.6.

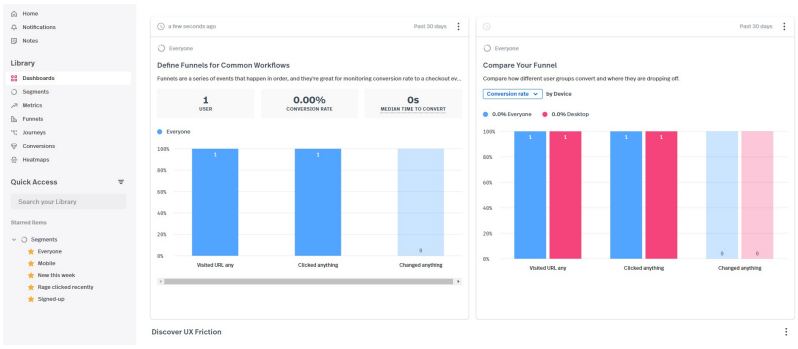


Рисунок 2.6 – Інтерфейс веб-сервісу Fullstory

На рисунку 2.7 можна побачити дії, які потрібно зробити щоб підключити сайт до Fullstory. Доступ до засобів аналітики є тільки в зареєстрованих користувачів. Після авторизації стають доступні інструменти для аналізу веб-ресурсів. Для збору аналітики необхідно зареєструвати свій сайт. Меню “Settings” дозволяє встановити відповідні налаштування для подальшого аналізу, в тому числі, згенерувати за допомогою опції “Fullstory Setup” код, який інтегрується безпосередньо в сайт, що необхідно проаналізувати. Після цього починається автоматичний збір і аналіз даних по всьому підключеному сайту.

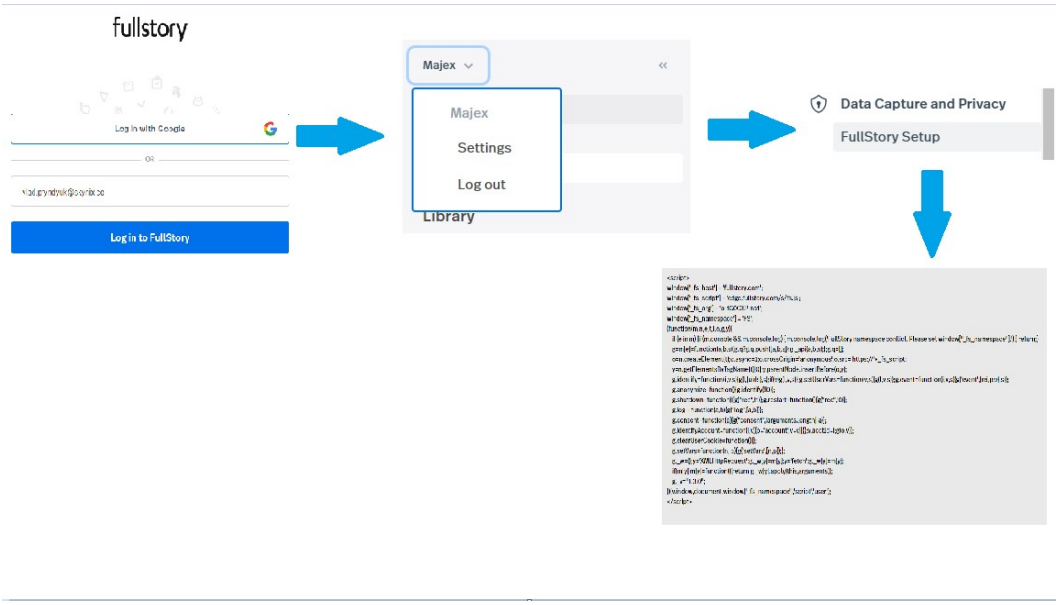


Рисунок 2.7 – Інструкція додавання сайту для Fullstory

Структрура лог файлу, який генерується Fullstory, представлена на рисунку 2.8.

	A	B	C	D	E	F	G	H	I	J	K
1	Event Name,Time,Distinct ID,City,Country,Operating System										
2	Tutorial Complete,1595055569,cbbe017b-cb88-5413-9785-29b9a00619c1,Pittsburg,United States,OSX										
3	Download Page,1595055560,56cf822c-7944-5095-9540-94b389cd0982,Salvador,Brazil,Android										
4	Test Integration,1595055550,582d8824-8eda-567c-9835-5e7d443a7cf4,Merlo,Argentina,OSX										
5	App Install,1595055525,4f32ee08-c2c2-5247-9b59-674165ff979f,Modena,Italy,Windows										
6	Sign Up,1595055524,13b70403-4ddb-511a-85f7-5bfdf18ff711,The Bronx,United States,Android										
7	Tutorial Complete,1595055523,2e23e652-4c9d-591f-ad09-f64eef14707e,Manchester,United Kingdom,iOS										
8	App Install,1595055499,cdd41c8c-a1e5-5782-8269-8ea2fb8b2eda,Seongnam-si,South Korea,iOS										
9	Tutorial Complete,1595055497,b0bbbceb-7836-5aa1-851c-5a9210eacfd6,Botucatu,Brazil,OSX										
10	Test Integration,1595055474,5d35573c-4727-5c9f-8eda-7c101bbbcb52,Islington,United Kingdom,Android										
11	Help Docs,1595055473,65d479b1-024c-5fba-9247-42acef7f7410,Dubai,United Arab Emirates,iOS										
12	Tutorial Complete,1595055461,c0aae1c1-5b4e-5192-b0a3-7a25c6f9232a,Kakuda,Japan,Windows										
13	Send Message,1595055455,e293ea25-09d1-5a46-8565-1f29d40e343b,Orenburg,Russia,Android										
14	App Install,1595055453,328a96da-03c4-5ba9-b391-8455a106e9aa,Tashkent,Uzbekistan,iOS										
15	App Open,1595055452,c43ba20b-7fe5-5d3b-ab2e-7e9bcda30957,Rochester,United States,Android										
16	Tutorial Start,1595055452,0e061bbb-4483-5826-a417-c25d9f402aad,Harrison,United States,Android										
17	Download Page,1595055449,2e5a5b81-2d5c-5e50-a90e-700077231c1a,Tianjin,China,Linux										
18	Tutorial Start,1595055444,f871c82c-2214-5964-b58b-992e2f86c85b,Birmingham,United States,Android										
19	Sign Up,1595055437,635a5b16-c645-5b98-9a3e-e0479d59f25d,Salvador,Brazil,OSX										
20	Send Message,1595055430,644bd1b7-3bbb-5b94-833c-ff97082ed81b,Qingdao,China,Windows										
21	Sign Up,1595055430,49cc0381-e1fc-57c5-84a0-c497949c1220,Liverpool,United Kingdom,iOS										

Рисунок 2.8 – Структура лог файлу Fullstory

Лог файл, згенерований за допомогою Fullstory, включає інформацію про подію яку виконав користувач на сайті, а також час затрачений на цю подію, спеціальний ідентифікатор, назва міста де перебуває користувач, код країни, платформа з якої заходили.

2.1.4 Mouseflow

Mouseflow - це платформа аналітики поведінки, яка використовується для оптимізації взаємодії з веб-сайтами з метою покращення конверсій, ідеально підходить для професіоналів UX, менеджерів із продуктів, цифрових маркетологів, стартапів, малих підприємств і підприємств. За допомогою Mouseflow можна знайти відповіді, яких традиційні інструменти аналітики не можуть знайти, дозволяючи спостерігати за сеансами відвідувачів вашого веб-сайту, автоматично створювати 6 типів теплових карт для всіх сторінок,

налаштовувати воронки, щоб спостерігати, куди та чому переходять відвідувачі, використовувати форму аналітику, щоб покращити залучення потенційних клієнтів, і запустити кампанії зворотного зв’язку, щоб дізнатися більше про своїх відвідувачів. Інтерфейс даного сервісу зображено на рисунку 2.9.

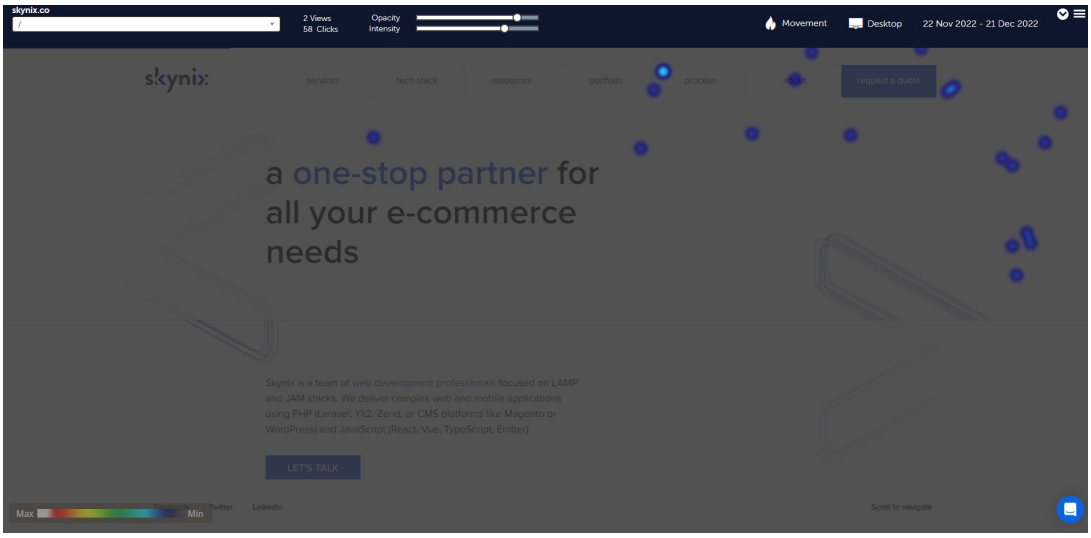


Рисунок 2.9 – Інтерфейс веб-сервісу Mouseflow

На рисунку 2.10 відображено послідовність дій, які потрібно зробити, щоб підключити сайт до Mouseflow. Так само, як і в більшості ресурсів, підключення сайтів до сервісу доступне лише після проходження реєстрації та подальшої авторизації. Додавання сайту для аналізу відбувається через меню налаштувань. Опція “Integration Flow” дозволяє вибрати платформу куди буде встановлюватися додаток, WordPress або HTML5. Після виконання необхідних налаштувань формується код, який необхідно вставити на сторінку, що планується для аналізу.

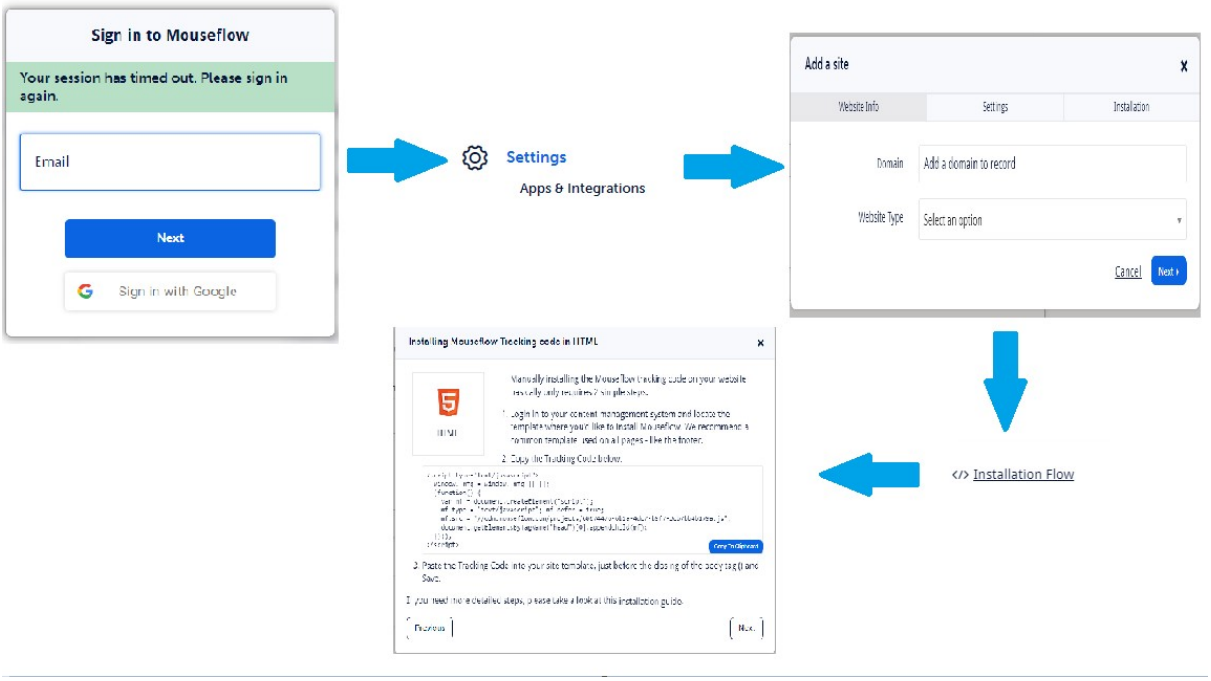


Рисунок 2.10 – Інструкція додавання сайту для Mouseflow

Структура лог файлу, який генерує сервіс Mouseflow, представлена на рисунку 2.11.


```
{
  "IndvId": 4910267242205184,
  "UserId": 4910267242205184,
  "SessionId": 6139895392849920,
  "PageId": 6237301188939776,
  "EventStart": "2022-12-15T09:38:14.564Z",
  "EventType": "navigate",
  "PageUrl": "https://skynix.co/portfolio",
  "PageRefererUrl": "",
  "PageIp": "",
  "PageAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36",
  "PageBrowser": "Chrome",
  "PageDevice": "Desktop",
  "PageOperatingSystem": "Windows",
  "PageNumInfos": 0,
  "PageNumWarnings": 0,
  "PageNumErrors": 0,
  "UserAppKey": "",
  "UserDisplayName": "User 1",
  "UserEmail": "",
  "PageLatLong": "49.8008,30.0983",
  "PageDuration": 1014166,
  "PageActiveDuration": 42685
},
```


Рисунок 2.11 – Структура лог файлу Mouseflow

Лог файл, згенерований за допомогою Mouseflow, включає інформацію про назву події, час початку події, URL сторінки, назва платформи яку

використовував користувач, кількість помилок на сторінці, локація користувача за допомогою довготи і широти, середній час на сторінці, активний час на сторінці.

2.2 Розробка структури лог файлу для збору інформації про дії користувачів сайту

Для аналізу usability сайту було обрано ступні показники:

– responseTime – потрібен для відслідковування з якою швидкістю завантажуються файли, може залежати від переходу між сторінками.

– loadingTime – потрібен для відслідковування з якою швидкістю завантажуються сайт, може залежати від швидкості інтернету користувача, а також переходами по сторінкам.

– windowHeight – потрібен для отримання висоти екрану користувача, може залежати чи буде користувач перевертати свій екран горизонтально.

– windowWidth – потрібен для отримання ширини екрану користувача, може залежати чи буде користувач перевертати свій екран горизонтально.

– countEvents – потрібен для отримання кількості подій на сайті, може залежати скільки раз користувач клікнув на певну кнопку, заповнив певне поле, скролив сайт і т.д..

– countClick – потрібен для відслідковування кількості кліків на сайті, може залежати від кількості натискань на будь яку часину сайту.

– countHover – потрібен для відслідковування кількості наводження, може залежати від наводження курсора миші на будь яку частину сайту.

– pageErrors – потрібен для відслідковування кількості помилок на сайті, може залежати від переходу між сторінками, кліків на елементи чи відправки форми.

– pageWarn – потрібен для відслідковування кількості попереджень, може залежати від переходу між сторінками, кліків на елементи чи відправки форми.

– countUsers – потрібен для отримання кількості користувачів на сайті, може залежати коли користувач переходить на сайт.

– countViews – потрібен для отримання кількості переглядів сайту, може залежати від переходу і перегляду сторінок.

– countReturn – потрібен для відслідковування кількості натискань на кнопку “назад”, може залежати від кліків на кнопку “назад”.

– countPermanentUsers – потрібен для збирання даних постійних користувачів, може залежати наскільки часто користувач переходить на сайт.

– timeSession – потрібен для відслідковування середнього часу сесії, може залежати від часу перебування на сайті.

– averageCountUsers – потрібен для відслідковування середньої кількості користувачів на сайті, може залежати від кількості користувачів які перейшли на сайт.

– averageCountUsersOnline – потрібен для збирання даних про середню кількість онлайн користувачів, може залежати від кількості користувачі які користуються сайтом прямо зараз.

– countPutTheBasketProduct – потрібен для збирання даних про кількість доданих товарів на сайті, може залежати від того скільки товарів додасть в кошик користувач.

– countCompleteThePurchase – потрібен для збирання даних про кількість оформлених товарів на сайті, залежить скільки разів користувачі підтвердили покупку товару.

Нижче зазначені показники які не є критичними для визначення якості usability і в кінцеву модель не включені:

- countEvents
- countPermanentUsers
- averageCountUsers
- averageCountUsersOnline

Зазначені показники повинні фіксуватись з певною періодичністю для того, щоб отримати картину поведінки користувачів сайту в динаміці.

Відповідність показників usability, які обрано для аналізу, та параметрів лог файлу, наведено в таблиці 2.1.

Таблиця 2.1- Зв’язок показників usability та даних лог файлу

Показники usability	**Показники користувацьких логів**	**Вплив на usability**
Швидкість	**responseTime** – час відповіді від сервера	Чим швидше завантажуться бібліотеки, тим швидше завантажиться сайт
Підключення	**loadingTime** – час завантаження сайту	Чим швидше завантажиться сайт, тим швидше користувач зможе приступити до дій
Розмір екрану	**windowHeight, windowWidth** – висота і ширина монітора або екрану	Можливість адаптуватися під різні екрани
Події	**countClick, countHover** – кількість подій, кліків або наводжень на елемент сторінки	Чим менше подій тим швидше працює сайт
Помилки	**pageErrors, pageWarn** – кількість помилок та попереджень на сторінці	Чим менше помилок тим краща швидкість сайту
Простота	**countViews, countReturn, timeSession** – кількість переглядів, повернень, а також час сесії на сайті	Чим простіше зроблений сайт тим більше користувачів він буде цікавити
Воронка конверсій	**countUsers, countPutTheBasketProduct, countCompleteThePurchase** – кількість користувачів, кількість доданих товарів у кошик на сайті, кількість завершених покупок	Чим зрозуміліший і читабельний сайт, тим користувач захоче повернутися до нього і придбати певний товар

2.2.1 Розробка математичної моделі та методу оцінки параметрів usability

Показник usability “Швидкість” позначимо як Speed. Будемо обчислювати Speed на основі змінної з логу responseTime, яка визначає час відповіді від серверу. Відповідно до вимог сервісу PageSpeed Insights час відповіді серверу повинен займати не більше 200 мс. Таким чином Speed може бути визначена виразом (2.1):

$$\text{Speed} = \text{E}[\overline{responseTime}] - 200 \quad (2.1)$$

де E[$\overline{responseTime}$] – математичне очікування набору N даних з лог-файлу за деякий період часу $\text{E}[\overline{responseTime}] = \frac{\sum_i^N responseTime}{N}$.

Показник usability “Підключення” позначимо як Connect. Будемо обчислювати Connect на основі змінної з логу loadingTime, яка визначає час завантаження сайту. Відповідно до вимог сервісу PageSpeed Insights час завантаження першого блоку інформації не повинен займати більше 2500 мс. Таким чином Connect може бути визначена виразом (2.2):

$$\text{Connect} = \text{T}[\overline{loadingTime}] - 2500 \quad (2.2)$$

де T[$\overline{loadingTime}$] – математичне очікування набору P даних з лог-файлу за деякий період часу $\text{T}[\overline{loadingTime}] = \frac{\sum_i^P loadingTime}{P}$.

Показник usability “Розмір екрану” позначимо як windowSize. Величина windowSize формується зі змінних з логу windowHeight і windowWidth, які визначають розмір екрану користувача, а представляє собою пару <windowHeight, windowWidth>. При заходженні на сайт, браузер визначає розмір екрану і підбирає відповідну таблицю стилів для адаптації сайту під розмір windowSize=<windowHeight, windowWidth>. Це робиться для того щоб

користувач міг зайти на сайт з будь якого куточку світу. Велика кількість доступних розмірів позитивно впливає на usability сайту, оскільки дозволяє використовувати різні типи пристроїв, телефон, планшет чи десктоп. Показник usability "Розмір екрану" може бути сформований на основі топ-N розширень екранів $countView_i$ серед усіх використаних (2.3):

$$countView_i = count(windowSize_i). \quad (2.3)$$

Показник usability "Події" позначимо як Events. Будемо обчислювати events на основі змінних з логу countClick і countHover які визначають кількість кліків і наводжень на елемент. Будемо використовувати правило 3-ох кліків. Правило 3-х кліків голосить, що користувачі не повинні здійснювати більше 3-х натискань на сторінці, щоб отримати доступ до потрібної інформації, інакше вони будуть розчаровані. Якщо задача вирішується більш, ніж за 5 кліків, така задача вважається складною і погіршує юзабіліті. Чим менше кліків робить користувач, тим краще юзабіліті це характеризує. При цьому, розташування та призначення клікабельних елементів повинно бути очевидним для користувача. Тобто гарне юзабіліті передбачає мінімальну потребу користувача в операції наводження на елементи, щоб дізнатись їх призначення. Вважатимемо, що в середньому недосвідченому користувачеві потрібно навести на елементи не більше, ніж 1,5*countClick разів. Будемо вимірювати usability за показником "Події" по шкалі від 1 до 3. Значення 1 відповідає низькому рівню юзабіліті, 2 – середньому рівню, 3 – високому рівню юзабіліті. Таким чином, показник usability "Події" будемо обчислювати за виразом (2.4):

$$Events = \begin{cases} 3, \textit{якщо}\ \text{countClick} < 3\ AND\ \text{countHover} < \text{countClick} * 1.5 \\ 2, \textit{якщо}\ 4 < \text{countClick} < 5\ AND\ \text{countHover} < \text{countClick} * 1.5, \\ 1, \textit{якщо}\ \text{countClick} > 5\ AND\ \text{countHover} > \text{countClick} * 1.5 \end{cases} \quad (2.4)$$

Показник usability “Помилки” позначимо як errors. Будемо обчислювати errors на основі змінних з логу pageErrors і pageWarns, які визначають кількість помилок і попереджень на сторінці. Якщо на сторінці буде більше 3 помилок і попереджень pageErrors * 2, це буде погано впливати на usability. Користувач має користуватися сайтом без всяких проблем і попереджень, щоб знову повертатися сюди. Будемо вимірювати usability за показником “Помилки” по шкалі від 1 до 3. Значення 1 відповідає низькому рівню юзабіліті, 2 – середньому рівню, 3 – високому рівню юзабіліті. Таким чином, показник usability “ Помилки” будемо обчислювати за виразом (2.5):

$$Errors = \begin{cases} 3, якщо\ pageErrors = 0\ \ AND\ \text{pageWarns} = 0 \\ 2, якщо\ 3\ \leq\ \text{pageErrors}\ \leq\ 1\ AND\ \text{pageWarns} \leq \text{pageErrors}\ *\ 2 \\ 1, якщо\ \text{pageErrors}\ >\ 3\ AND\ \text{pageWarns} > \text{pageErrors}\ *\ 2 \end{cases} \quad (2.5)$$

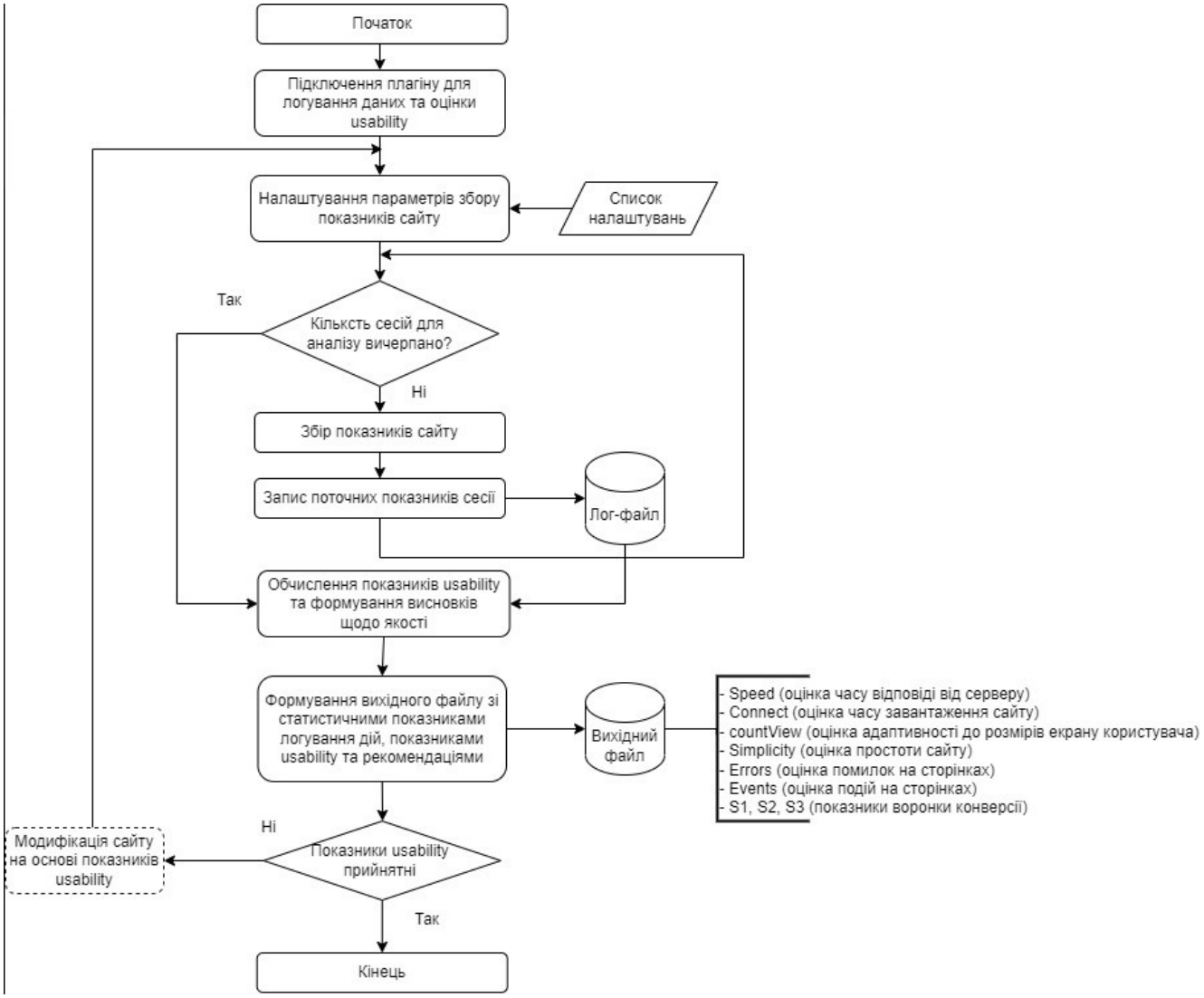
Показник usability “Простота” позначимо як simplicity. Будемо обчислювати simplicity на основі змінних з логу countViews, countReturn і timeSession які визначають кількість переглядів, повернень, а також час сесії на сайті. Простота сайту полягає в тому щоб на сторінці були тільки ті елементи які можуть зацікавити користувача. Вважатимемо, що якщо користувач переглянув сайт більше 2 разів затримався на сайті більше 10 секунд і повернувся назад менш ніж за 10 хвилин то це означає що інтерфейс сайту прости і гарно впливає на usability. Будемо вимірювати usability за показником “Простота” по шкалі від 1 до 3. Значення 1 відповідає низькому рівню юзабіліті, 2 – середньому рівню, 3 – високому рівню юзабіліті. Тому показник usability “Простота” можна вирахувати за виразом (2.6):

$$Simplicity = \begin{cases} 3, якщо\ counViews >\ 2\ AND\ \text{timeSession} >\ 10s\ AND\ countReturn < 10min \\ 2, якщо\ counViews >\ 1\ AND\ \text{timeSession} >\ 5s\ AND\ countReturn < 15min \\ 1, якщо\ counViews <\ 1\ AND\ \text{timeSession} <\ 3s\ AND\ countReturn > 20min \end{cases}. \quad (2.6)$$

Показник usability “Воронка конверсій” позначимо як conversionFunnel. Будемо обчислювати conversionFunnel на основі змінних з логу countUsers, countPutTheBasketProduct і countCompleteThePurchase які визначають кількість користувачів, додавання продукту в корзину і завершених покупок на сайті. Будемо вираховувати conversionFunnel за воронкою конверсії. Воронка конверсії відноситься до подорожі відвідувача на веб-сайті, який проходить кілька послідовних етапів, таких як інтерес, бажання та дії, таким чином перетворюючись із простого відвідувача на покупця/передплатника/лідера тощо. Наприклад, за певний період, сайт відвідала певна кількість користувачів, позначимо через countUser, потім певна кількість людей додати товар до кошика, позначимо через countPutTheBasketProduct, далі певна кількість користувачів завершила покупку товара на сайті, позначимо через countCompleteThePurchase, а також позначимо результати через S_1, S_2, S_3:

$$\begin{aligned} S_1 &= countPutTheBasketProduct/countUser * 100, \\ S_2 &= countCompleteThePurchase / countPutTheBasketProduct * 100, \\ S_3 &= countCompleteThePurchase / countUser * 100. \end{aligned} \quad (2.7)$$

Показник usability “Воронка конверсій” (conversionFunnel) визначається пропорціями відношення показників S_1 , S_2 , S_3 на основі думок експертів, оскільки для кожної предметної галузі сайту можуть бути свої нюанси визначення якості та обмежень цього показника.


Початок
Підключення плагіну для логування даних та оцінки usability
Налаштування параметрів збору показників сайту
Список налаштувань
Так
Кількість сесій для аналізу вичерпано?
Ні
Збір показників сайту
Запис поточних показників сесії
Лог-файл
Обчислення показників usability та формування висновків щодо якості
Формування вихідного файлу зі статистичними показниками логування дій, показниками usability та рекомендаціями
Вихідний файл
- Speed (оцінка часу відповіді від серверу)
- Connect (оцінка часу завантаження сайту)
- countView (оцінка адаптивності до розмірів екрану користувача)
- Simplicity (оцінка простоти сайту)
- Errors (оцінка помилок на сторінках)
- Events (оцінка подій на сторінках)
- S1, S2, S3 (показники воронки конверсії)
Ні
Показники usability прийнятні
Модифікація сайту на основі показників usability
Так
Кінець

3. ОПИС ТА РЕЗУЛЬТАТИ ОЦІНКИ USABILITY ЗА ДОПОМОГОЮ ДАНИХ КОРИСТУВАЦЬКИХ ЛОГІВ

3.1 Особливості використання інструменту «Консоль»

Інструмент Console є допоміжним інструментом для використання з іншими інструментами DevTools. Консоль надає потужний спосіб створення функціональних сценаріїв, перевірки поточної веб-сторінки та керування поточною веб-сторінкою за допомогою JavaScript. Інструмент Console допомагає з кількома завданнями, а саме:

- Відстеження проблем, щоб дізнатися, чому щось не працює в поточному проекті.
- Отримати інформацію про веб-проект у браузері у вигляді повідомлень журналу.
- Реєстрація інформації в сценаріях для цілей налагодження.
- Запуск JavaScript у консолі. Взаємодія з веб-проектом у браузері за допомогою JavaScript.

Консоль показана на рисунку 3.1. Вона містить панель DevTools та відкритий інструмент Elements.

Консоль є місцем за замовчуванням, де повідомляються про помилки JavaScript і підключення. Якщо виникають будь-які помилки, лічильник проблем відображається поруч із піктограмою налаштувань у DevTools, яка відображає кількість помилок і попереджень. Для отримання додаткових відомостей використовується розділ Виправлення помилок JavaScript, про які повідомляється в Консолі. DevTools надає детальну інформацію про помилку в консолі (рис.3.2).

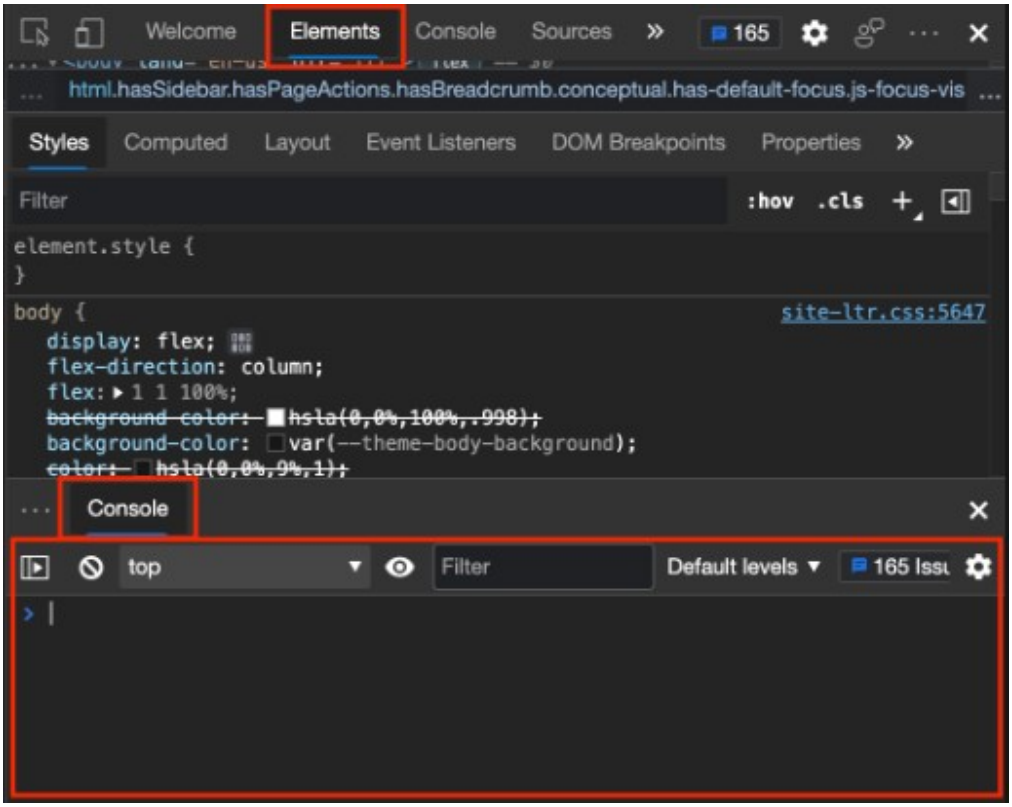


Рисунок 3.1 – Вигляд консолі

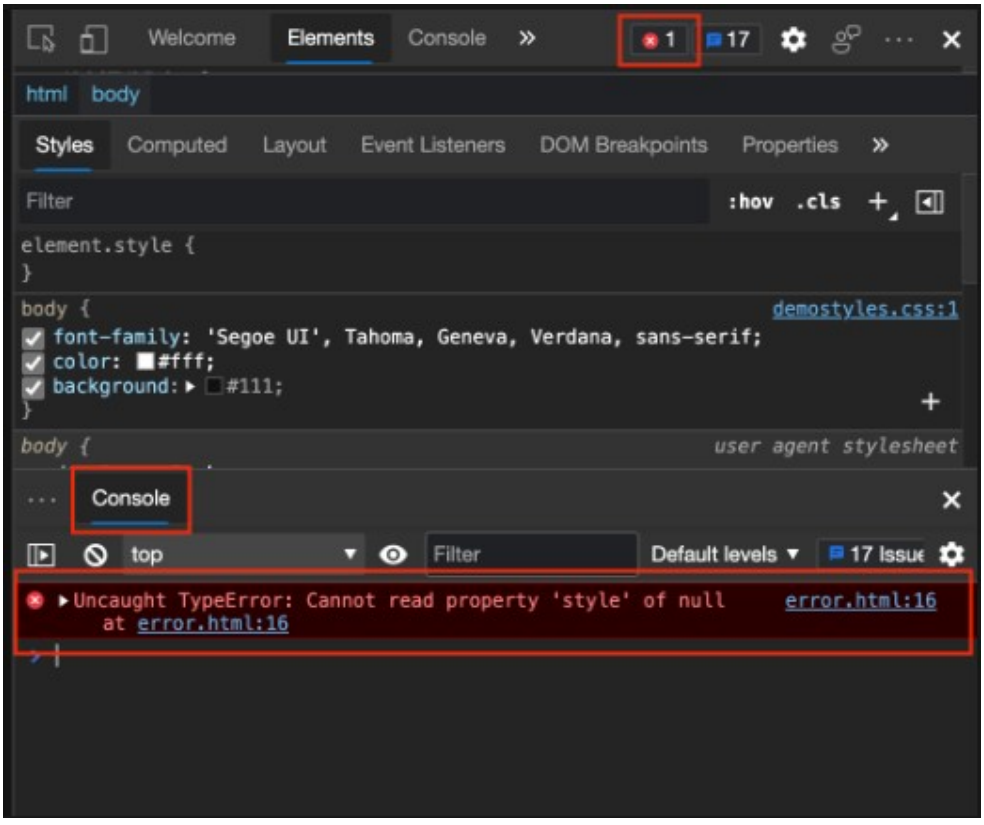


Рисунок 3.2 – Відображення помилки у консолі

При відкритті DevTools на веб-сторінці консолі може містити велику кількість інформації. Обсяг інформації стає проблемою, коли потрібно визначити важливу інформацію. Щоб переглянути важливу інформацію, яка потребує дії, використовується інструмент Issues у DevTools. Це дозволяє поступово перемістити проблеми з консолі в інструмент проблем. Для цієї ж задачі ожуть бути використані параметри автоматичного журналу та фільтри (рис. 3.3).

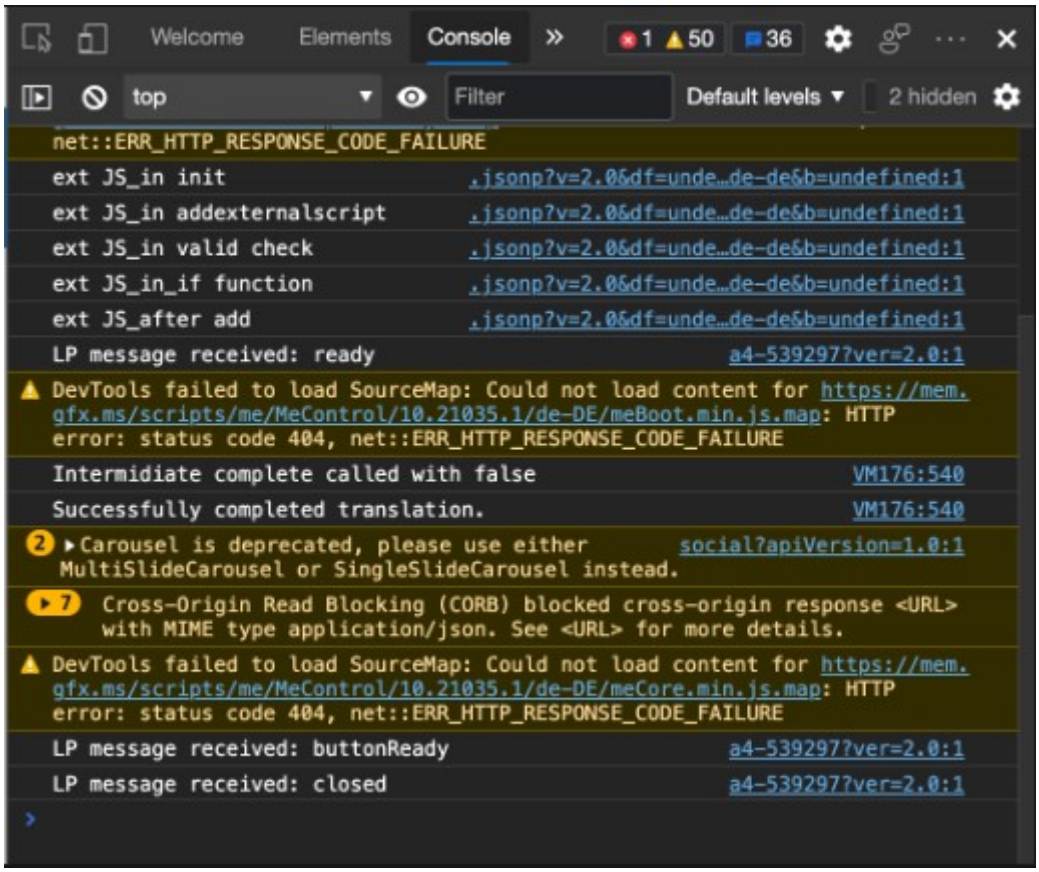


Рисунок 3.3 – Відображення інформації у консолі

Консоль — це не лише місце для реєстрації інформації. Консоль є середовищем де можна писати будь-який JavaScript. На рисунку 3.4 можна побачити виконання простої арифметичної дії 2 + 2, а також за допомогою інструменту “Mouseflow” встановлення коду який буде відстежувати поведінку миші користувача і передавати інформацію.

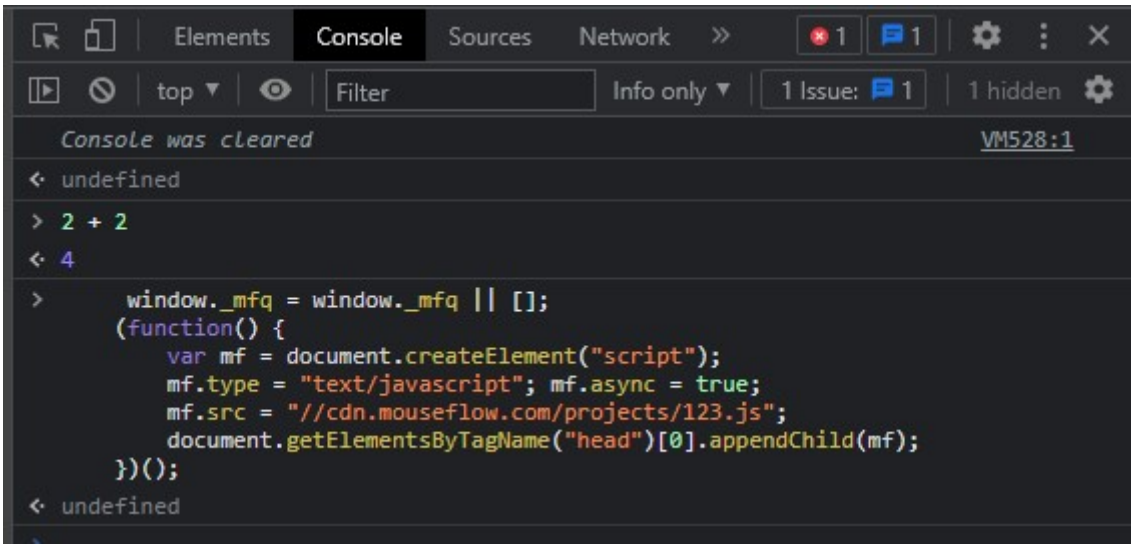


Рисунок 3.4 – Додавання JavaScript коду до консолі

У результаті консоль розробника це інтегроване середовище розробки з набором інструментів, які можна використовувати для створення, налагодження та тестування програм на сайті.

3.2 Опис та особливості використання програми

Порядок застосування інструменту для оцінки даних користувацьких логів включає наступні кроки:

1. Готовий код можна вставити в консоль або підключити до сайту.
2. Коли протрібно будуть дані, натиснути на кнопочку у правому верхньому углу сайта.
3. Після того як дані завантажаться, потрібно відкрити файл із результатами.

Програма написана на мові програмування JavaScript і влючає наступні ключові методи:

- Extension() – повертає об’єкт даних (рис. 3.5);
- OnLoad() – викликається при заходженні на сайт (рис.3.6);

- downloadCsvFile() – дозволяє завантажувати лог файл у форматі csv, результати роботи наведено на рисунку 3.7.

```
function Extension() {
  1 usage
  const getResponseTime = () => getNumberInRange(0, 400);
  1 usage
  const getLoadingTime = () => getNumberInRange(0, 3000);
  1 usage
  const getCountOfClicks = () => getNumberInRange(0, 5);
  1 usage
  const getCountOfHover = () => getNumberInRange(0, 10);
  1 usage
  const getPageErrors = () => getNumberInRange(1, 3);
  1 usage
  const getPageWarn = () => getNumberInRange(1, 6);
  1 usage
  const getCountOfUsers = () => getNumberInRange(10, 25);
  1 usage
  const getCountOfView = () => getNumberInRange(0, 3);
  1 usage
  const getCountOfReturn = () => getNumberInRange(10, 25);
  1 usage
  const getCountPutTheBasketProduct = () => getNumberInRange(0, 15);
  1 usage
  const getCountCompleteThePurchase = () => getNumberInRange(0, 5);

  return {
    responseTime: getResponseTime(),
    loadingTime: getLoadingTime(),
    windowHeight: window.innerHeight,
    windowWidth:  window.innerWidth,
    countClick: getCountOfClicks(),
    countHover: getCountOfHover(),
    pageErrors: getPageErrors(),
    pageWarn: getPageWarn(),
    countViews: getCountOfView(),
    countReturn: getCountOfReturn(),
    timeSession: timeSession,
    countUsers: getCountOfUsers(),
    countPutTheBasketProduct: getCountPutTheBasketProduct(),
    countCompleteThePurchase: getCountCompleteThePurchase()
  }
}
```

Рисунок 3.5 – Функція Extension

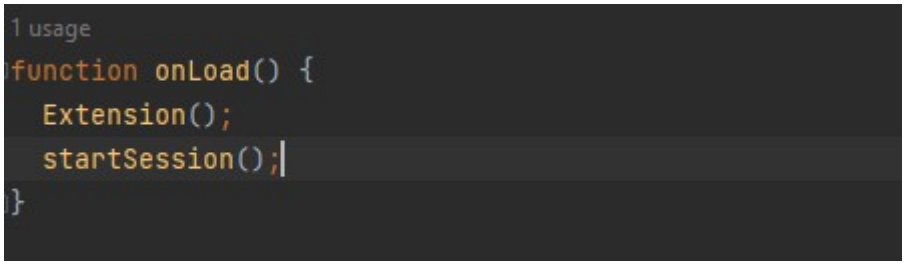


Рисунок 3.6 – Функція onLoad

Рисунок 3.7 – Функція downloadCsvFile

Використовувати програму можна двома шляхами, перший це можна підключити файл в якому буде знаходитися даний код, в головний файл сайту, а саме в index.html звідки запускається будь-який сайт за допомогою тега <script>, який потрібно використовувати для підключення js файлів на сайті, як показано на рисунку 3.8.

Рисунок 3.8 – Підключення js файлу

Зазначену операцію можна викликати іншим шляхом, це відразу підключати даний код в консоль, але ми зможемо зібрати дані тільки з одної сорінки, де можна використовувати код JavaScript, як це показано на рисунку 3.9.

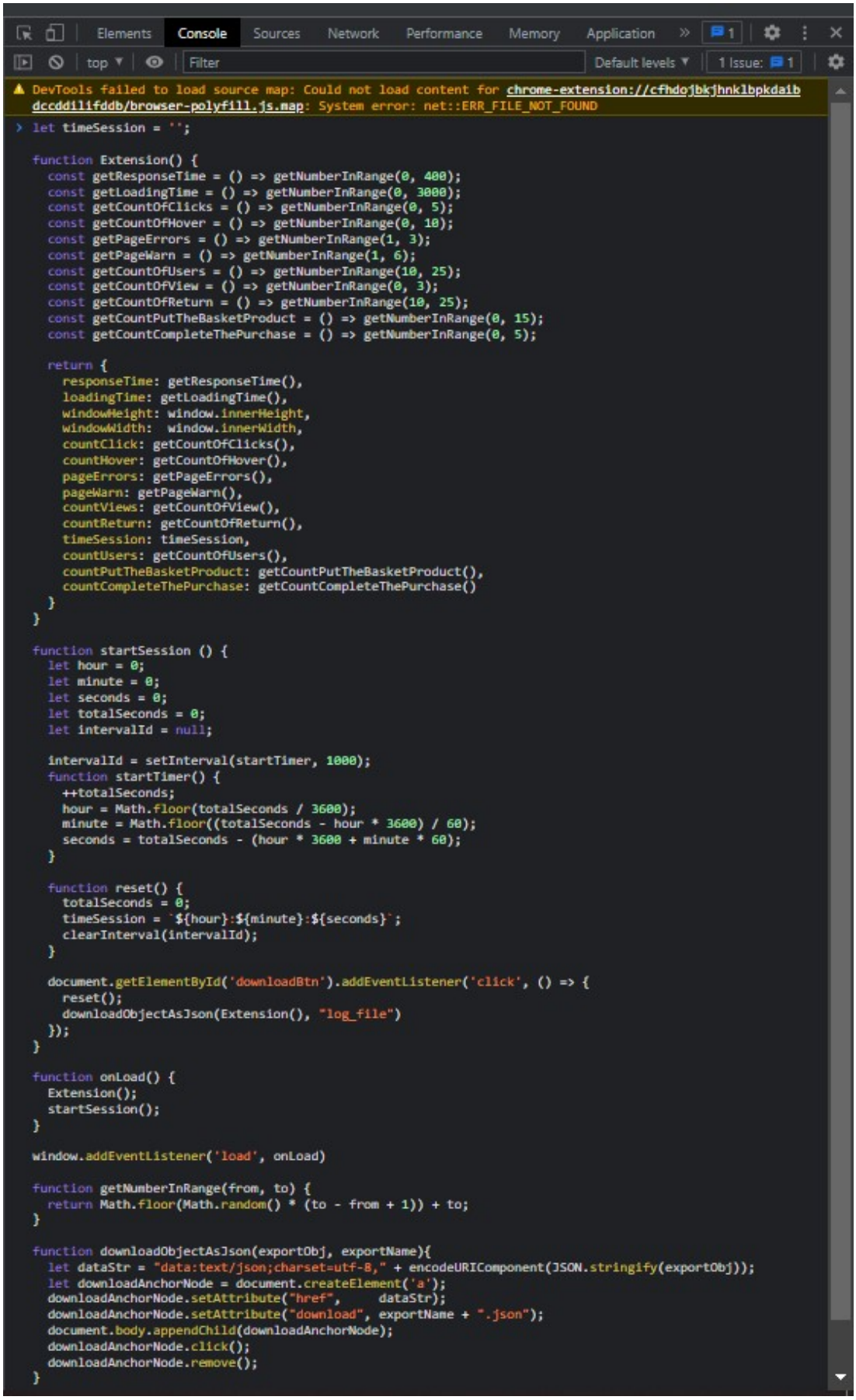


Рисунок 3.9 – Використання програми у консолі

Дизайн у програми досить простий для використання, як показана на рисунку 3.10:

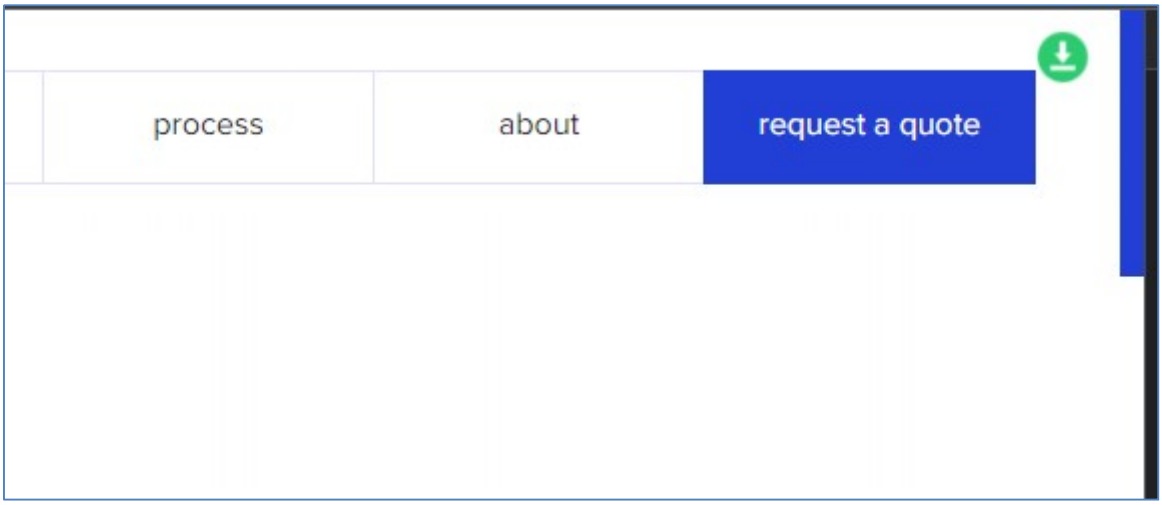


Рисунок 3.10 – Дизайн програми

Як показано на малюнку 3.10, це іконка, яка розташована в правому кутку сайта і буде слідкувати за діями користувача, щоб в будь-який момент можна було натиснути на цю кнопку і отримати лог файл з даними, як показано на рисунку 3.11.

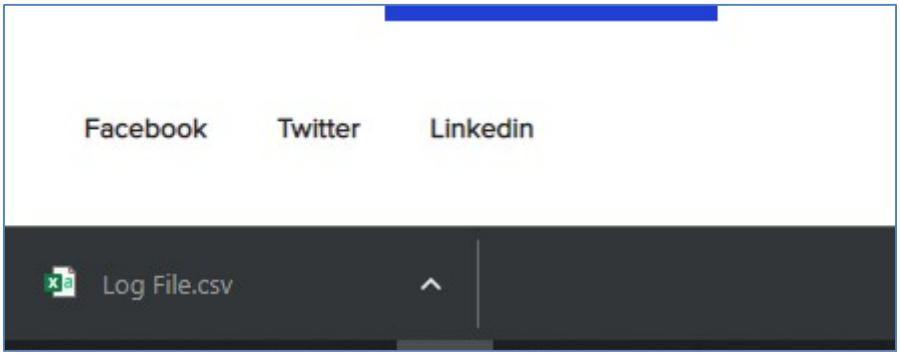


Рисунок 3.11 – Скачування лог файлу

Після скачування файлу користувач може переглянути даний файл і знайти інформацію яка йому потрібна для покращення usability сайту. Дані лог файлу показано на рисунку 3.12.

	A	B	C	D
1		responseTime	217.49	
2		loadingTime	1434.51	
3		windowSize	1: 1920x1080 2: 1440x1024 3: 992x72(	
4		countClick	2.11	
5		countHover	10.44	
6		pageErrors	1.01	
7		pageWarn	1.96	
8		countViews	51.81	
9		countReturn	52.99	
10		timeSession	1:13:34	
11		countUsers	46.58	
12		countPutTheBasketF	7.67	
13		countCompleteTheF	5.18	
14				
15		Speed	200 < 217.49 < 300	середнє
16		Connect	1434.51 < 2500	гарне
17		CountView	count(1920x1080, 1	гарне
18		Events	2.11 < 3 AND 10.44	середнє
19		Errors	1.01 < 3 AND 1.96 <	гарне
20		Simplicity	51.81 > 2 AND 1:13:	гарне
21		S1	7.67 / 46.58 * 100 = 16.46%	
22		S2	5.18 / 7.67 * 100 = 67.53%	
23		S3	5.18 / 46.58 * 100 = 11.12%	

Рисунок 3.12 – Розрахункові дані лог файлу

Зведені дані по лог файлу демонструють, що usability оцінюваного сайту доволі не погане, оскільки середня відповідь від сервера менше 200ms, середнє завантаження файлів менше 2500ms, кількість кліків не більше 3, а за правилом 3-ох кліків це гарний показник usability, а значить користувачі знайшли те що потрібно при цьому зробивши трішки більше 9 наводжень, кількість помилок на сайті не більше 1 і попереджень не більше 3, середній час сесії тривав 30 хвилин і 15 секунд, що означає що користувачам сподобалось простота usasbility, також на сайті побувало 10 користувачів, всі вони додали до корзини речі які обрали в загальній кількості 15 штук, і лиш на 5 речей пройшли форму оформленя.

ВИСНОВКИ

При виконанні роботи було вирішено наступні задачі.

1. Проаналізовано існуючі підходи до оцінки usability. Традиційні методи оцінки здебільшого передбачають залучення експертів до оцінювання показників usability і грунтуються на суб'єктивних даних. Крім того, оцінка людиною потребує багато часу, а тривалий час проведення заходів по оцінці usability може привести до ситуацій, коли людина не помічає певних факторів, що впливають на usability або допускає помилки в оцінці в силу фактору втомленості. В той же час, для веб-ресурсів можна виділити групи показників функціонування самих сайтів та показників поведінки користувачів, які є об'єктивними та вимірюваними в кількісних, а не в якісних одиницях, і збір яких можна автоматизувати, що робить оцінку usability з використанням лог файлів дій користувачів одним із зручних інструментів.

2. Проаналізовано інструменти для збору даних з сайту. Визначено, що кожен інструмент має певну спрямованість та не дозволяє отримати комплексну оцінку.

3. Визначено показники usability для оцінки та перелік факторів, які впливають на цю оцінку і фожуть бути сформовані на основі даних викристання сайту. Розроблено математичну модель, що визначає порядок обчислення якості usability сайту на основі даних лог файлу. Розроблено метод автоматизованої оцінки usability на основі даних користувацьких логів.

4. Розроблено програмне забезпечення, яке реалізує створений метод та проведено тестування методу на реальному сайті. Екпериментальні дослідження охоплюють дані 100 сесій користувачів, на основі яких сформовані показники usability, що можуть бути використані для подальшої оптимізації сайту. Використання для розрахунку показників usability лог файлів дозволяє уникнути впливу людського фактору, підвищити швидкість отримання та об'єктивність кінцевих результатів оцінки usability.

ПЕРЕЛІК ПОСИЛАНЬ

1. 10 good deeds in web design. Nielsen Norman Group. URL: https://www.nngroup.com/articles/ten-good-deeds-in-web-design/ (date of access: 29.12.2022).
2. Anderson N. S., Norman D. A., Draper S. W. User centered system design: new perspectives on human-computer interaction. The american journal of psychology. 1988. Vol. 101, no. 1. P. 148. URL: https://doi.org/10.2307/1422802 (date of access: 29.12.2022).
3. A survey on automated log analysis for reliability engineering / S. He et al. ACM computing surveys. 2021. Vol. 54, no. 6. P. 1–37. URL: https://doi.org/10.1145/3460345 (date of access: 29.12.2022).
4. Automation Anywhere. What is RPA? Robotic process automation | automation anywhere. Automation Anywhere. URL: https://www.automationanywhere.com/rpa/robotic-process-automation (date of access: 29.12.2022).
5. Bastien J. M. C. Usability testing: a review of some methodological and technical aspects of the method. International journal of medical informatics. 2010. Vol. 79, no. 4. P. e18-e23. URL: https://doi.org/10.1016/j.ijmedinf.2008.12.004 (date of access: 29.12.2022).
6. Comprehensive log compression with frequent patterns / H. Kimmo et al. SpringerLink. URL: https://link.springer.com/chapter/10.1007/978-3-540-45228-7_36 (date of access: 29.12.2022).
7. DEEPCASE: semi-supervised contextual analysis of security events. IEEE Xplore. URL: https://ieeexplore.ieee.org/document/9833671/ (date of access: 29.12.2022).
8. Dominici G., Palumbo F. How to build an e-learning product: factors for student/customer satisfaction. Business horizons. 2013. Vol. 56, no. 1. P. 87–96. URL: https://doi.org/10.1016/j.bushor.2012.09.011 (date of access: 29.12.2022).

9. Elliott M., Kling R. Organizational usability of digital libraries: case study of legal research in civil and criminal courts. Journal of the american society for information science. 1997. Vol. 48, no. 11. P. 1023–1035. URL: https://doi.org/10.1002/(sici)1097-4571(199711)48:11%3C1023::aid-asi5%3E3.0.co;2-y (date of access: 29.12.2022).
10. Extension of pacmad model for usability evaluation metrics using goal question metrics (gqm) approach | semantic scholar. Semantic Scholar. URL: https://www.semanticscholar.org/paper/EXTENSION-OF-PACMAD-MODEL-FOR-USABILITY-EVALUATION-Fabil/132e14e65bbc59d04de69d06dc1330a15a06ebe8 (date of access: 29.12.2022).
11. Hertzum M. Images of usability. International journal of human-computer interaction. 2010. Vol. 26, no. 6. P. 567–600. URL: https://doi.org/10.1080/10447311003781300 (date of access: 29.12.2022).
12. Hoashi K. Usability evaluation of visualization interfaces for content-based music retrieval systems. Academia.edu. URL: https://www.academia.edu/2577623/Usability_evaluation_of_visualization_interfaces_for_content-based_music_retrieval_systems (date of access: 29.12.2022).
13. ISO 9241-11:2018(en) ergonomics of human-system interaction – part 11: usability: definitions and concepts. Online Browsing Platform (OBP). URL: https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en (date of access: 29.12.2022).
14. ISO 9241-210:2010. ISO. URL: https://www.iso.org/standard/52075.html (date of access: 29.12.2022).
15. ISO/IEC 25010:2011. ISO. URL: https://www.iso.org/standard/35733.html (date of access: 29.12.2022).
16. ISO/IEC 9126-1:2001. ISO. URL: https://www.iso.org/standard/22749.html (date of access: 29.12.2022).

17. ISO/IEC 9126:1991 software enginnering – product quality. iTeh Standards. URL: https://standards.iteh.ai/catalog/standards/iso/0c496d92-bb02-451c-93cc-26cd504a6987/iso-iec-9126-1991 (date of access: 29.12.2022).
18. List of salient user interface events with number of occurrences and a brief description. SMARTech. URL: https://smartech.gatech.edu/bitstream/handle/1853/3558/95-13.pdf?sequence=1&isAllowed=y (date of access: 29.12.2022).
19. LogTree: a framework for generating system events from raw textual logs. IEEE Xplore. URL: https://ieeexplore.ieee.org/document/5694003/ (date of access: 29.12.2022).
20. Logzip: extracting hidden structures via iterative clustering for log compression. arXiv.org. URL: https://arxiv.org/abs/1910.00409 (date of access: 29.12.2022).
21. Remote usability testing: study guide. Nielsen Norman Group. URL: https://www.nngroup.com/articles/remote-usability-testing-study-guide/ (date of access: 29.12.2022).
22. Shackel B. Usability – Context, framework, definition, design and evaluation. Interacting with Computers. 2009. Vol. 21, no. 5-6. P. 339–346. URL: https://doi.org/10.1016/j.intcom.2009.04.007 (date of access: 29.12.2022).
23. Top 10 mistakes in web design. Nielsen Norman Group. URL: https://www.nngroup.com/articles/top-10-mistakes-web-design/ (date of access: 29.12.2022).
24. Usability evaluation methods | usability.gov. Home | Usability.gov. URL: https://www.usability.gov/how-to-and-tools/methods/usability-evaluation/index.html (date of access: 29.12.2022).
25. Google Analytics URL: https://analytics.google.com/.
26. Mixpanel URL: https://mixpanel.com/.
27. Fullstory URL: https://app.fullstory.com.
28. Mouseflow URL: https://app.mouseflow.com.