

Міністерство освіти і науки України

Український державний університет науки і технологій

Факультет Комп'ютерні технології та системи
Кафедра Комп'ютерні інформаційні технології


Пояснювальна записка

до кваліфікаційної роботи
освітнього ступеня магістра

на тему: «Дослідження якості інтерфейсів у веб-сайтах з використанням системи для А/В тестування»

за освітньою програмою **Інженерія програмного забезпечення**
зі спеціальності: **121 Інженерія програмного забезпечення**

Виконав: студент групи «ПЗ2221»



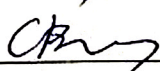
/Олексій БЕЗРЕБРИЙ/

Керівник:




/ Іван КЛИМЕНКО/

Нормоконтролер:



/Світлана ВОЛКОВА/

Засвідчую, що у цій роботі немає запозичень з
праць інших авторів без відповідних посилань
Студент



(підпис)

Дніпро – 2025 рік

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: Комп'ютерних технологій і систем
Кафедра: Комп'ютерні інформаційні технології
Рівень вищої освіти: магістр
Освітня програма: Інженерія програмного забезпечення
Спеціальність: Інженерія програмного забезпечення



ЗАТВЕРДЖУЮ
Завідувач кафедри _____ КІТ
_____ Вадим ГОРЯЧКІН
_____ грудня 2024 р. ~~X~~

змінити шріфт

ЗАВДАННЯ

На кваліфікаційну роботу _____ Магістр _____
студенту Безребрий Олексій Олександрович.

1. Тема дипломної роботи: «Дослідження якості інтерфейсів у веб-сайтах з використанням системи для А/В тестування».

Керівник роботи: Клименко Іван Вікторович
затверджені наказом 1196 ст від 05.12.2023 року

2. Строк подання студентом роботи заповнити 01.2025 року

3. Вихідні дані до дипломної роботи:

заповнити.

4. Зміст пояснювальної записки (перелік питань до розробки):

4.1. Аналітична частина: Аналіз сучасного стану дослідження за науковими літературними джерелами;

4.2. Основна частина: Дослідження ККД у різних дизайнів за допомогою використання А/В тестування;

5. Перелік демонстраційного матеріалу:

5.1. презентація;

5.2. демонстраційне відео.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз сучасного стану дослідження проблеми за науковими літературними джерелами	01.09.23 – 01.10.23	10%
2	Аналіз сучасного стану програмно-апаратного забезпечення, яке потребує вдосконалення для вирішення проблем дослідження	01.10.23 – 10.10.23	
3	Постановка задачі, технічне завдання	04.11.24 – 10.11.24	30%
4	Розробка інструментальних засобів дослідження	15.11.24 – 01.12.24	
5	Виконання досліджень	09.12.24 – 15.12.24	60%
6	Оформлення пояснювальної записки	01.12.24 – 01.01.25	
7	Розробка демонстраційних матеріалів	06.01.25 – 12.01.25	100%
8	Подання кваліфікаційної роботи до кафедри	18.01.25	
9	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	<u>24.01.25</u>	



Студент:

Олексій БЕЗРЕБРИЙ

Керівник роботи:

Іван КЛИМЕНКО

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи магістра складається з 14 рис. , 6 табл., 3 додатків, 22 джерел.

Об'єкт дослідження: функціональні можливості інтерфейсів веб-сайтів у контексті їхньої оптимізації за допомогою систем А/В тестування.

Мета роботи: дослідити якість інтерфейсів веб-сайтів та ефективність їх удосконалення за допомогою А/В тестування. Провести аналіз методів А/В тестування, визначити їхній вплив на зручність користувачів, конверсію та бізнес-показники. Надати рекомендації щодо використання А/В тестування для поліпшення користувацького досвіду.

Методика дослідження: теоретичний аналіз наукових і літературних джерел з UX-дизайну та А/В тестування, практичне тестування інтерфейсів, статистичний аналіз результатів А/В тестування, порівняння ефективності різних підходів до оптимізації інтерфейсів.

Перелік ключових слів: веб-сайти, інтерфейс користувача, А/В тестування, UX-дизайн, оптимізація.

ЗМІСТ

Вступ.....	8
Розділ 1 Аналіз сучасного стану дослідження за науковими літературними джерелами	9
1.1 Загальні поняття веб-інтерфейсів.....	9
1.1.1 Аналіз якості інтерфейсів веб-сайтів	13
1.1.2 Важливість якості інтерфейсів у веб-середовищі.....	16
1.1.3 Засоби оцінки якості інтерфейсів	16
1.2 А/В тестування як метод дослідження.....	22
1.2.1 Визначення А/В тестування	22
1.2.2 Цілі та переваги використання А/В тестування в інтерфейсах веб-сайтів	23
1.3 Моделі якості інтерфейсів.....	35
1.4 Висновки до розділу	36
Розділ 2 Обґрунтування методів дослідження.....	38
2.1 Основні функціональні вимоги.....	38
2.2 Вхідні данні.....	38
2.3 Вихідні дані.....	39
Висновки до розділу 2	40
Розділ 3 ДОСЛІДЖЕННЯ ЯКОСТІ ВЕБ-ІНТЕРФЕЙСА ЗА ДОПОМОГОЮ СИСТЕМИ ДЛЯ А\В ТЕСТУВАННЯ НА ОСНОВІ ОБРАНОГО САЙТУ	41
3.1 Методологія дослідження	41
3.2 Вибір сайту для дослідження.....	41
3.3 Опис системи А/В тестування, що використовується.....	42
3.3.1 Визначення мети тестування	42
3.3.2 Вибір змін для тестування.....	42
3.4 Розробка тестових варіантів.....	43
3.5 Аналіз отриманих результатів	45
3.6 Рекомендації для поліпшення якості інтерфейсу веб-сайту.....	47
Висновки до розділу 3	49
Розділ 4 ПОБУДОВА БІЗНЕС МОДЕЛІ	50
4.1 Оцінка проблеми	50
4.2 Зацікавлені сторони	51
4.3 Комерційне рішення	52
4.4 Клієнти, сегменти ринку споживання.....	53

	7
4.4 Бізнес модель	53
4.5 Висновки за розділом 4	55
Загальний висновок.....	56
Бібліографічний список	57
ДОДАТОК А.....	65
ДОДАТОК Б	81
ДОДАТОК В.....	99

ВСТУП

У сучасному суспільстві, інтернет є важливим засобом комунікації та обміну інформацією. Останніми роками багато уваги приділяється розробці та вдосконаленню веб-інтерфейсів, які впливають на користувацький досвід. Однак, незважаючи на позитивні досягнення в цій галузі, виникли проблеми, пов'язані з якістю інтерфейсу та його впливом на користувачів.

Існуючі дослідження в галузі якості веб-інтерфейсів вказують на необхідність більш ефективних способів оцінки та покращення їх функціональності. Деякі аспекти ще недостатньо вивчені і потребують подальших дослідницьких зусиль.

У сучасному цифровому світі, з надто великою конкуренцією в інтернет-просторі, важливо надавати користувачам максимально зручні та ефективні інтерфейси.

Актуальність даного дослідження полягає в необхідності забезпечення конкурентоспроможності веб-сайтів шляхом покращення їх дизайну та взаємодії.

Основною метою даної кваліфікаційної роботи є розробка та впровадження ефективного методу оцінки якості веб-інтерфейсів з використанням системи А/В тестування.

Дослідження спрямоване на надання веб-сайтам інструментів для покращення користувацького досвіду та оптимізації продуктивності.

Для досягнення цієї мети перед дослідженням поставлені наступні завдання:

- проаналізувати існуючі методи оцінки якості веб-інтерфейсів.
- розробити та впровадити методологію використання системи А/В тестування для оцінки та порівняння ефективності веб-інтерфейсів.
- оцінити результати дослідження та надати рекомендації щодо покращення якості веб-сайту.

Результати дослідження матимуть практичну користь для веб-розробників та сприятимуть підвищенню якості та конкурентоспроможності веб-сайтів.

РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО СТАНУ ДОСЛІДЖЕННЯ ЗА НАУКОВИМИ ЛІТЕРАТУРНИМИ ДЖЕРЕЛАМИ

1.1 Загальні поняття веб-інтерфейсів

Розвиток веб-технологій сьогодні має величезний вплив на життя людей. Наука ніколи не стоїть на місці, і веб-інфраструктури все більше розвиваються. Інтернет-технології замінили телебачення і газети, завоювавши довіру користувачів і дозволивши їм користуватися онлайн-програмами і читати новини в будь-якій точці світу.

Варто зазначити, що зміст і наповнення веб-сайтів також впливають на споживачів. Інтернет-культура відображає цінності сучасного суспільства. Веб-дизайн відіграє важливу роль у формуванні естетичної культури різних категорій інтернет-користувачів, впливаючи на їхнє візуальне мислення, світогляд та вподобання. Оскільки метою багатьох дизайн-проектів є перетворення продукту або послуги, завдання дизайнера - розробити веб-сайт, який позитивно впливає на психіку користувача, є естетично привабливим, прийнятним, адаптивним, зручним і комфортним для роботи на будь-якому пристрої.

Протягом історії Інтернету його контент еволюціонував від простого джерела інформації до витвору мистецтва. Щоб привернути увагу користувачів, веб-дизайнери відображають інформацію все більш витонченими способами. Користь цих інноваційних методів не завжди очевидна.

У будь-якому випадку, тенденції веб-дизайну продовжують змінюватися і розвиватися, так само як мода і музика. Більше того, якщо на початку розвитку інтернету багато дизайнерських рішень були обмежені рівнем розвитку технологій, то сьогодні дизайнери мають майже повну свободу. Більше того, якщо дизайнерські інновації правильно застосовувати поряд з технологічними, веб-сайти будуть привабливими для користувачів з різними смаками та вподобаннями.[11]

У вузькому сенсі веб-інтерфейс - це не лише графічний дизайн і компоненти, але й архітектура, що лежить в основі веб-сайту. Важливо, щоб він враховував досвід користувачів, відповідав їхнім потребам і забезпечував зручність використання.

Концепція веб-інтерфейсу також включає в себе принципи взаємодії, такі як визначення завдань, простота навігації та доступність. Враховуючи сучасні тенденції, такі як використання мобільних пристроїв, адаптивний дизайн і мінімалістичний підхід також стають важливими елементами веб-інтерфейсу.

Серед найважливіших складових частин веб-інтерфейсу можна виділити у табл. 1: назву оформити згідно положенню

Таблиця 1 Частини веб-інтерфейсу

Складова	Поняття
Дизайн веб-інтерфейсів	Елементи дизайну включають візуальні аспекти веб-інтерфейсу, такі як кольорова палітра, шрифти, графіка та макет. Добре продуманий веб-сайт сприяє простоті взаємодії та створює приємні враження у користувачів.
Навігація веб-інтерфейсів	Ефективна навігація включає в себе логічну структуру сайту, легкий пошук інформації та інтуїтивно зрозумілу навігацію. Добре організована навігація полегшує взаємодію та забезпечує швидкий доступ до різних розділів.
Відображення інформації	Якісне подання інформації означає не лише якісний дизайн, але й зрозумілість інформації на веб-сайті. Це включає в себе елементи, які

	полегшують розпізнавання інформації, такі як чітка ієрархічна структура, зрозумілий текст і легкі для сприйняття графіки.
Взаємодія веб-інтерфейсів	Взаємодія включає в себе всі можливі способи взаємодії користувача з веб-сайтом, такі як введення даних, вибір опцій і взаємодія з елементами керування. Добре продумані взаємодії роблять користувацький досвід зручним та ефективним.

Також розрізняють декілька видів веб-інтерфейсів наведених у табл. 2:

Таблиця 2 Види веб-інтерфейсів

Вид інтерфейсу	Поняття
Текстовий (зокрема, інтерфейс командного рядка)	тип інтерфейсу користувача, який використовує лише алфавітно-цифрові символи та псевдографічні символи для введення, виведення та відображення інформації. Характеризується низькими апаратними ресурсами (особливо пам'яттю), необхідними для введення і виведення, і високою швидкістю відображення інформації.
Тактильний	сенсорні комп'ютерні технології. Ці технології можна використовувати для створення віртуальних об'єктів у комп'ютерних симуляціях, керування віртуальними об'єктами та

	покращення дистанційного керування машинами та обладнанням.
Графічний	інструментальна система для взаємодії користувача з електронними пристроями, заснована на відображенні всіх доступних користувачеві об'єктів і функцій системи у вигляді графічних компонентів екрану (вікон, піктограм, меню, кнопок, списків тощо). Однією з вимог до хорошого графічного інтерфейсу користувача програмної системи є концепція «роби те, що я маю на увазі».
Жестовий	тип графічного інтерфейсу для пристроїв з сенсорними екранами або спеціальними пристроями введення, які можуть емулювати клавіатурні команди і комбінації клавіш за допомогою жестів.
Матеріальний(дотик)	це різновид інтерфейсу користувача, у якому взаємодія людини з електронними пристроями відбувається з допомогою матеріальних предметів і конструкцій[1].
Нейрокомп'ютерний інтерфейс	Системи, призначені для обміну інформацією між мозком та електронними пристроями (наприклад, комп'ютерами) [2]. В

	<p>односпрямованому інтерфейсі зовнішні пристрої можуть отримувати сигнали від мозку або надсилати сигнали до мозку (наприклад, електронні імплантати імітують сітківку при відновленні зору). У двонаправленому інтерфейсі мозок і зовнішній пристрій можуть обмінюватися інформацією в обох напрямках. Нейрокомп'ютерні інтерфейси часто базуються на методах біологічного зворотного зв'язку.</p>
Голосовий	<p>використання голосових або мовних платформ уможливорює людсько-комп'ютерну взаємодію для запуску автоматизованих сервісів і процесів. Визначення цільової аудиторії має вирішальне значення. Наприклад, якщо ПІ створюється для широкої споживчої аудиторії, особливу увагу слід приділити простоті використання та надати низку інструкцій і порад для початківців.</p>

1.1.1 Аналіз якості інтерфейсів веб-сайтів

Подібно до того, як стрімкий розвиток веб-технологій відображається на функціональності сучасних веб-сайтів, технології дизайну також не відстають і з кожним роком стають все більш важливими для користувачів.

Користувачі, які відвідують веб-сайт, відразу ж звертають увагу на його графічне оформлення і зручність навігації. У більшості випадків враження, яке справляє зовнішній вигляд сайту, визначає, чи залишаться відвідувачі на цій сторінці, чи продовжать пошук на інших ресурсах.

Аналіз якості інтерфейсу веб-сайту - це комплексний підхід до вивчення й оцінювання чинників, що визначають взаємодію користувачів із веб-ресурсом. Здатність веб-сайту задовольняти потреби та очікування користувачів визначається якістю інтерфейсу, який поєднує в собі дизайн, навігацію, функціональність і взаємодію. Аналіз якості інтерфейсу сайту визначає важливий теоретичний аспект, що лежить в основі процесу розроблення та вдосконалення інтерактивних платформ в Інтернеті.

Сьогодні для аналізу дизайну веб-сайтів та поведінки споживачів можна використовувати широкий спектр інструментів і методів[14].

Нижче перераховані найпопулярніші.

1. Eye tracking - це метод маркетингових досліджень, який дає змогу отримати об'єктивні дані про те, куди і чому дивляться респонденти, вивчаючи інтерфейс веб-сайту компанії. Ця інформація дає змогу розробникам сайтів контролювати увагу відвідувачів. Основними недоліками цього інструменту є його висока вартість і те, що він може бути проведений тільки на відносно невеликій вибірці респондентів.

2. Юзабіліті-тестування - це метод оцінки зручності та ефективності інтерфейсу сайту. Цей вид тестування надає власнику сайту набір даних про поведінку відвідувачів сайту та допомагає оцінити сценарії поведінки користувачів і знайти "вузькі місця" на сайті, що характеризуються незручними для користувачів функціями.

3. Sidebyside- респондентам пропонуються два варіанти дизайну. Респонденти голосують за той, який їм більше подобається. Цей метод підходить для тестування іконок, тексту та логотипів, але не для складніших інтерфейсів, оскільки респонденти не можуть натиснути на рішення.

4. А/В-тестування - дає об'єктивну інформацію про зручність використання різних версій сайту. Часто маркетологи тестують верстку і дизайн сторінок сайту, зміст сайту і розташування окремих елементів на сторінках сайту.

5. Методи аналізу кліків і конверсій дають змогу виявити елементи сайту, що привертають увагу користувачів, і на основі цієї інформації створити сценарії поведінки користувачів.

6. Методи онлайн-досліджень можна використовувати на різних етапах розроблення та розвитку сайту, щоб допомогти розв'язати локальні проблеми, поліпшити інтерфейс і визначити напрямки подальшого розвитку ресурсів

7. Тестування привабливості дає змогу оцінити перші враження користувачів від сайту та його прототипу, знайти слабкі місця в дизайні інтерфейсу і створити додаткові тригери для розкриття пропозицій щодо угод на сайті.

8. Сортування карток. Цей метод використовується для розробки системи навігації на сайті. Пропонуючи респондентам розташувати картки в логічному порядку, розробники генерують ідеї щодо розташування блоків і розділів на майбутніх сайтах.

9. Евристична оцінка - це експертний аналіз для визначення схильності системи до певних ризиків. Евристичний аналіз у UX-дизайні використовується UX-фахівцями для виявлення загальних проблем під час використання продукту. процедура.

10. Експертна оцінка схожа на евристичну оцінку, але не спирається на конкретну евристику. Як правило, в оцінці системи беруть участь два або три аналітики, які перевіряють дизайн системи на відповідність правилам і принципам. Вони записують свої спостереження і класифікують їх за ступенем важливості.

11. Фокус-групи - це тип опитування, під час якого проводяться групові обговорення для визначення ставлення членів групи до дизайну інтерфейсу веб-сайту. Основний недолік цього методу полягає в тому, що учасники фокус-груп схильні до сильного впливу і можуть неохоче висловлювати свою думку, якщо вони не згодні з думкою більшості.

12. Інтерв'ю - проводяться в прямому ефірі, коли респондентам ставлять запитання про продукт, а їхні відповіді записуються.

1.1.2 Важливість якості інтерфейсів у веб-середовищі

В умовах сучасної конкуренції на ринку важливо розуміти, що якість веб-інтерфейсу виходить за рамки естетики. Користувацький досвід і взаємодія з веб-сайтом - ключовий фактор успіху онлайн-проекту.

Успішні веб-сайти не тільки надають продукти та інформацію, а й створюють позитивний емоційний досвід для користувача. Це охоплює не тільки зручний і зрозумілий дизайн, а й ефективну взаємодію та розуміння потреб цільової аудиторії.

Важливі аспекти[13]:

- користувацький досвід (UX): Якість інтерфейсу безпосередньо впливає на користувацький досвід. Зручний та логічний інтерфейс робить взаємодію із веб-сайтом приємною та ефективною.
- ефективність взаємодії: Якість інтерфейсу визначає, наскільки легко користувач може використовувати функціонал веб-сайту. Ефективна взаємодія забезпечує швидкість та точність використання ресурсу.

Розуміння потреб користувачів: вдало спроектований інтерфейс враховує потреби та очікування цільової аудиторії, що є важливим аспектом для забезпечення їхнього задоволення від взаємодії з веб-сайтом.

1.1.3 Засоби оцінки якості інтерфейсів

Оцінка якості інтерфейсів включає в себе ряд засобів, спрямованих на удосконалення користувацького досвіду та забезпечення ефективної взаємодії з веб-сайтом, зокрема:

Усунення помилок включає в себе активний моніторинг взаємодії користувача з веб-сайтом та оперативне виправлення виявлених недоліків. Це може бути здійснене через регулярні тестування та залучення користувачів до процесу виправлення.

Панель користувача — це елемент, який визначає зручність та легкість використання веб-сайту. Це включає в себе як структурні елементи (меню, кнопки, тощо), так і взаємодію з контентом.

Аналітика відіграє ключову роль у визначенні ефективності та взаємодії користувача з веб-сайтом. Це може бути здійснене через використання інструментів аналітики, які дозволяють збирати та аналізувати дані про поведінку користувачів.

Незважаючи на низку досліджень, присвячених дизайну веб-сайтів для загального пошуку інформації та електронної комерції, досі не існує стандартних рамок для визначення ефективності веб-сайту з візуальної точки зору.

Тому, науковці Т.О. Говорущенко, В.С. Гончарук та К.О. Кобель запропонували деякі загальні категорії (рис. 1.1), які можуть чітко визначити термін «зручність використання» [3].

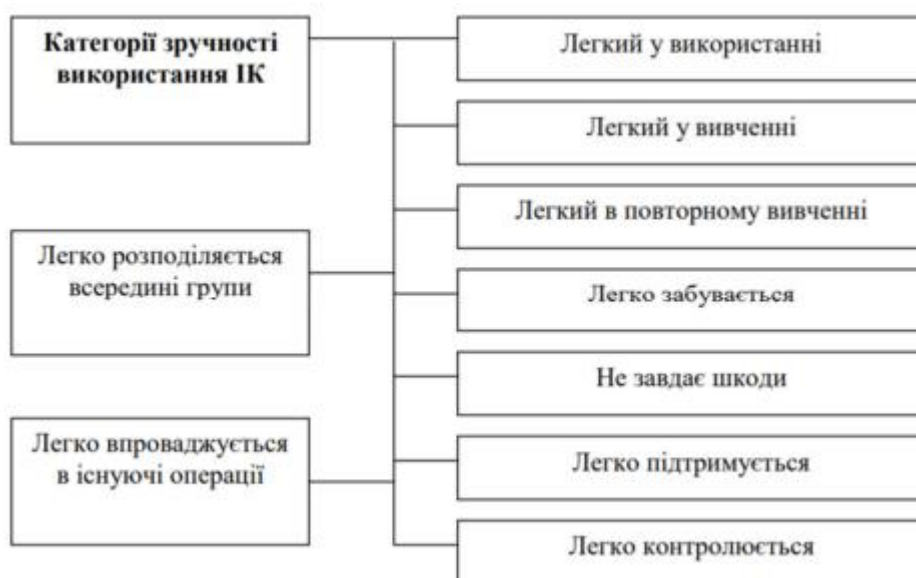


Рис. 1.1. Категорії зручності використання інтерфейсу користувача [3]

Загалом, для користувачів якість дизайну означає візуальні особливості веб-сайту, які приваблюють користувачів, утримують їх на сайті довше і спонукають повернутися. Найбільш важливими аспектами якості дизайну є такі показники і пункти контрольного списку

- колір

- Зображення/звук/відео

- текст.

Розглянемо ці критерії більш детально.

Колір Під час розроблення веб-сайту зручність використання та естетика, безсумнівно, є важливими чинниками. Однак одного цього недостатньо для досягнення високих показників конверсії. Насправді, колірна гамма веб-сторінки також є важливим фактором її успіху [4, 7].

Колір - це перше, що привертає увагу користувача, і його відповідність темі дуже важлива. Вибір кольору - це не питання особистих уподобань. Кожен колір має своє значення і може чинити певний вплив на людей. Саме тому кольори так важливі у веб-дизайні і можуть миттєво передати потрібну інформацію про ваш бізнес. Згідно з соціологічними дослідженнями, 85 % людей стверджують, що кольори мають значний вплив на те, що вони купують [4, 14].

Щоб вибрати найбільш підходящу колірну схему, веб-дизайнери керуються психологією кольору, яка розкриває емоційні та смислові елементи кольорів, і колірним колом, яке показує взаємозв'язок між різними кольорами. Вони також враховують наявні тенденції в конкретних сферах діяльності. Вирішальним фактором є бізнес і цілі читача.

Усім відомо, що кольори впливають на настрій людей. Теплі кольори розслаблюють нас і змушують забути про проблеми. І навпаки, холодні кольори змушують людей почуватися пригніченими. Також слід враховувати аудиторію вашого сайту. Якщо сайт орієнтований на дітей, то кольори та картинки мають бути яскравими та життєрадісними, а якщо на літніх людей, то слід використовувати теплі, спокійні кольори. Під час вибору кольорів для сайту слід дотримуватися певних правил:

- основна кольорова гама сайту повинна характеризувати ідею або тему сайту.

Вона повинна відповідати фірмовому стилю;

- обрані кольори повинні відповідати змісту;

- оптимальна кількість кольорів для сайту - до чотирьох;

- кольори слід підбирати відповідно до вікової групи відвідувачів;

- не рекомендується використовувати кольори, що засліплюють очі. Краще використовувати спокійні та заспокійливі відтінки;

- рекомендується використовувати однакову кольорову гаму на кожній сторінці сайту.

Зображення/звук/відео. Зображення - це більше, ніж просто елемент краси. У веб-дизайні зображення відіграють важливу роль у юзабіліті. Більшість веб-користувачів розпізнають зображення швидше, ніж текст. Тому зображення часто є першим елементом макету, який відвідувачі бачать, сканують і розшифровують [7,9]. Крім того, зображення є інформативними, емоційними та передають повідомлення, а також викликають певні естетичні відчуття. Правильно використані та оптимізовані зображення також позитивно впливають на SEO-показники сайту, тобто на здатність користувачів знаходити сайт або конкретну сторінку в пошукових системах.

Для веб-сайтів підходять зображення (фотографії), які

- насичені інформацією
- високоякісні
- унікальні
- емоційно-залучаючі.

Правильно підібрані ілюстрації можуть оживити сайт. Головна роль ілюстрацій - не просто заповнювати порожній простір, а викликати емоційний або брендовий зв'язок. Авторські ілюстрації допоможуть зробити сайт впізнаваним. Їх можна використовувати для створення заголовків, іконок та анімації.

Під час наповнення сайту ілюстраціями важливо дотримуватися одного стилю і не перестаратися. Навіть найкрасивіші ілюстрації можуть бути відкинуті, якщо їх буде занадто багато.

Відео на сторінці миттєво привертає увагу користувача. Основні переваги відео - динаміка, розвиток сюжету та більший емоційний відгук.

Відео на сайтах підходять для таких цілей

- показати щось, що складно пояснити (навчальні відео)

- розповісти історію
- демонструвати продукти

Фонове відео не слід використовувати виключно з естетичних міркувань. Це пов'язано з тим, що воно погіршує продуктивність веб-сайту і не передає ніякої інформації користувачеві. Зараз існують різні способи аналізу цифрових зображень[9]. Такий аналіз дозволяє використовувати цифрові зображення як джерело додаткової інформації [4].

Текст є основним джерелом інформації для будь-якого ресурсу. Тексти створюються для певної цільової аудиторії і покликані вирішити конкретне завдання. Тому тут важливим є написаний зміст, унікальність контенту, спосіб його подачі та візуальне оформлення [8]. Інформація повинна легко сприйматися користувачем сайту з будь-якого ракурсу.

Найкращі шрифти для веб-сайту - це ті, які дають змогу швидше читати текст і краще розпізнавати та засвоювати інформацію. Відповідний розмір шрифту не викликає втоми очей. Особливо під час роботи з великими обсягами інформації.

Більшість користувачів не звертають особливої уваги на шрифти і розпізнають їх інтуїтивно. Однак вони закривають сторінку, якщо з нею щось не так. Шрифти також можуть надати тексту переконливості, зробити його більш діловим або, навпаки, створити розважальну атмосферу.

Найбільша складність у виборі шрифту полягає у великій кількості існуючих варіантів. При виборі слід дотримуватися таких основних правил

Орієнтуйтеся на мету й аудиторію. Обирайте діловий шрифт, якщо ви пишете серйозну новинну статтю, а цільова сторінка не викликатиме роздратування. Також визначьтеся з читацькою аудиторією. Великі шрифти, що легко зчитуються, підходять для дітей, а контрастні, декоративні - для дорослих.

Зберігайте простоту Під час вибору шрифтів для веб-сайту перевагу слід віддавати простим шрифтам, які легко зчитуються. Рукописні або химерні шрифти можуть ускладнити читання тексту, тому їх слід використовувати в шапці сайту, щоб привернути до нього увагу.

Мінімалізм. Не використовуйте занадто багато шрифтів. Залежно від типу ресурсу, достатньо 2-3. В іншому випадку не тільки дизайн буде порушений, але і продуктивність сайту буде негативно впливати на роботу. Це пов'язано з тим, що кожен раз, коли ви завантажуєте зайвий шрифт, проект сповільнюється [10].

Комбінації. Рідко коли використовується лише один шрифт. Тому важливо підбирати пари, які гармонійно поєднуються і витримані в одному тоні.

Розмір. Обираючи шрифт для сайту, слід також враховувати розмір тексту. Оптимальний розмір тексту - від 16 до 24 пунктів, але рукописні шрифти не підходять, оскільки більші шрифти виглядають краще.

Відображення. Важливо, щоб контент відображався належним чином у всіх браузерах та операційних системах. Особливо це стосується мобільних версій веб-сайтів. Розглянувши та проаналізувавши основні показники перевірки дизайну веб-сайтів, ми сформулювали перелік критеріїв для оцінки елементів дизайну інтерфейсу веб-сайту, який наведено в Табл 3.

Таблиця 3 Основні показники перевірки дизайну

Ознака	Критерій
Якість	Актуальність дизайну
	Єдиний стиль
Відповідність	Відповідність інформації
	Баланс
	Зручність навігації
Колір	Гармонійна кольорова гама
	Читабельність тексту за кольором
Зображення/звук/відео	Якість зображення/звуку/відео
Текст	Поєднання шрифтів
	Читабельність шрифтів

Загалом, на думку М.О.Компанєєтс, ефективність (корисність, зручність та візуальна привабливість) веб-сайту залежить від виконання наступних умов

- всі елементи на веб-сторінці повинні бути логічно структуровані та структурно впорядковані відповідно до їхньої важливості;

- розробка інтерфейсу веб-сайту таким чином, щоб користувачі розуміли, де вони знаходяться на сайті і як вони можуть перейти туди, де їм потрібно; і

- створення візуального образу веб-сайту з урахуванням змісту, характеру та призначення веб-сайту з відповідною структурою;

- дотримання загального стилю та тематики всіх сторінок веб-сайту, щоб він сприймався цілісно та гармонійно [12].

Таким чином, у цьому дослідженні були сформульовані та обґрунтовані основні критерії оцінки елементів інтерфейсу веб-сайту. Ці критерії, з одного боку, створюють більш ергономічні умови для користувачів і підвищують привабливість дизайну, а з іншого боку, правильно реалізовані на етапі проектування заходи допомагають збільшити дохід від цього ресурсу.

1.2 А/В тестування як метод дослідження

А/В-тестування є важливим інструментом для дослідження та оптимізації веб-інтерфейсів. Воно дає змогу проводити експерименти, порівнюючи ефективність різних варіантів дизайну, і визначати зміни, які приводять до кращих результатів. За статистикою, 77% компаній у світі використовують цей метод для розвитку своїх веб-ресурсів [5]. Поліпшення комфорту користувачів сприяє залученню нових клієнтів і збільшенню прибутку. Далі розглянемо, як проводиться А/В-тестування і чому воно настільки ефективне.

1.2.1 Визначення А/В тестування

А/В-тестування (спліт-тестування) — це метод порівняння двох версій дизайну, який допомагає обрати найефективніше рішення для досягнення конкретної мети. Можна тестувати веб-сторінки, окремі елементи (наприклад, кнопки), мобільні програми чи шаблони електронних листів.

Наприклад, припустимо, що червона кнопка приверне більше клієнтів, ніж зелена. Відвідувачів сайту можна розділити на дві групи: одна бачить сторінку із зеленою кнопкою, інша — з червоною. Порівнюючи результати, легко визначити, який варіант працює краще.

Таким чином, А/В-тестування є контрольованим експериментом, що базується на об'єктивних даних. Це дозволяє ефективніше досягати поставлених цілей і мінімізувати ризики. Метод зазвичай застосовується для підвищення конверсії, розширення аудиторії та покращення бізнес-показників.

1.2.2 Цілі та переваги використання А/В тестування в інтерфейсах веб-сайтів

Правильно організувавши процес спліт-тестування, можна проаналізувати всі елементи веб-сайту, рекламних оголошень, цільових сторінок та електронних листів:

- назва та опис публікації (ефективність)
- дизайн сторінки (кольори, стиль, розміщення ключових елементів)
- форми зворотного зв'язку (дизайн, розміщення, заклик до дії);
- зображення товарів на сторінках інтернет-магазину.

Всі продукти, як і веб-сайти, відображають бачення обмеженої кількості людей: вас (замовника) і команди розробників. Але чи сподобається це вашим клієнтам? Чи правильні припущення експертів щодо комерційної привабливості продукту? Відповіді на ці питання дає спліт-тестування.

А/В-тестування вирішує щонайменше чотири завдання:

1. допомагає перевіряти гіпотези.
2. захищає від ризикованих експериментів, непередбачуваних витрат і неефективних витрат часу. Провести спліт-тест і отримати рішення за два тижні набагато простіше, ніж витратити гроші на випадково впроваджені зміни.
3. рішення можна приймати на основі перевірених даних, а не інтуїції. Часто думки та досвід маркетологів та інших експертів не збігаються зі світоглядом більшості реальних бізнес-клієнтів. Чи так це насправді, можна з'ясувати за допомогою тестування.

4. сприяти дослідженню цільових аудиторій. Експерименти можуть показати, що подобається клієнтам, що їх заохочує і як збільшити продажі. Ця інформація може бути дуже корисною при роботі над новими проектами.

5. А/В-тестування дозволяє експериментувати по-науковому, використовуючи статистичні дані для отримання об'єктивних результатів.

Основними цілями А/В-тестування є:

Оптимізація конверсії: визначити та впровадити зміни для збільшення кількості користувачів, які виконують бажану дію (покупку, реєстрацію тощо).

Покращення користувацького досвіду: покращення користувацького досвіду: впровадження та тестування нових елементів дизайну, навігації та функціональності для покращення користувацького досвіду.

Максимізація часу перебування на сайті: розгляд та впровадження змін для збільшення часу, який користувачі проводять на сайті.

Глибокий аналіз: дозволяє проаналізувати вплив конкретних змін на результати, щоб з'ясувати, які елементи впливають на користувацький досвід і продуктивність.

Які ж переваги дає А/В-тестування? Вирішення проблем відвідувачів. Користувачі приходять на сайт з певною метою: вони хочуть дізнатися більше про продукт, купити щось або просто подивитися. При цьому відвідувачі сайту стикаються з типовими труднощами. Наприклад, кнопка "замовити" знаходиться в незручному місці або її важко знайти. Такі моменти негативно впливають на користувацький досвід і позначаються на конверсії.

Підвищення рентабельності інвестицій. Тестування та аналіз сайту дозволяє ефективніше використовувати поточний трафік і підвищити конверсію без витрат на залучення нового трафіку. Навіть невеликі коригування можуть позитивно вплинути на показники.

Зменшити кількість прямих повернень користувачів. Для оцінки ефективності сайту важливо аналізувати цей показник. Люди закривають сторінки ресурсів з різних причин: невідповідність очікуванням, незручні форми замовлення, складність пошуку товарів тощо. За допомогою тестування можна

створити кілька варіацій розташування елементів і перевірити, яка з них працює краще. Якщо спостерігаються позитивні зрушення, можна вносити глобальні зміни.

Мінімізуйте ризик коригувань. Експерти радять вносити зміни дозовано і послідовно, щоб не знизити показники конверсії. Наприклад, припустимо, вам потрібно впровадити нову функцію на вашому сайті. Проведіть тести, щоб зрозуміти, який варіант буде більш вигідним.

A/B аналіз ґрунтується на конкретній інформації, а не на кавовій гущі. На основі статистично значущих поліпшень, таких як час перебування на сторінці, кількість покинутих візків тощо, можна легко визначити виграшний варіант змін.

Це може бути корисно для покращення дизайну. Редизайн може бути різним за обсягом: від заміни кількох кольорів до повної зміни ресурсу. Вибирати рішення слід виключно на основі результатів спліт-тесту. Спочатку зробіть невеликі зміни, протестуйте їх і спробуйте зберегти найкращу версію.

A/B-тестування, без сумніву, є необхідним інструментом. Воно виводить роботу сайту на бажаний рівень і гарантує, що витрати і час, витрачені на зміни, виправдані[16].

1.2.3 Сервіси для A/B тестування

Маркетологи використовують спеціальні інструменти для спліт-тестування. Найпопулярніший з них - Google Optimise. Він допомагає тестувати окремі елементи сторінки, такі як заголовки, шрифти та зображення. Цей інструмент має безкоштовну версію і користується популярністю серед конкурентів.

Інші сервіси для тестування сторінок

- Optimizely: сервіс, який дозволяє користувачам тестувати різні варіанти веб-сайту, мобільної програми або інших елементів. Надає аналіз та інструменти для легкого впровадження змін.

- VWO (Visual Website Optimiser): цей сервіс дозволяє проводити A/B-тестування, багатоваріантне тестування, макетування сайту та інші експерименти для оптимізації конверсії.

- Split.io: надає можливість проводити функціональні тести та керувати функціональними властивостями разом з аналізом результатів.

- Unbounce: фокусується на тестуванні цільових сторінок, оптимізації конверсії та підвищенні ефективності рекламних кампаній.

- Crazy Egg: надає теплові карти та інші інструменти візуалізації, які допомагають зрозуміти, як користувачі взаємодіють з різними елементами сторінки.

- Convert: сервіс, що дозволяє запускати різні типи тестів, включаючи A/B-тестування, багатовимірне тестування та спліт-тестування.

- AB Tasty: надає інструменти для тестування різних елементів, таких як заголовки, зображення та кнопки, щоб визначити їхній вплив на конверсію.

Існує також низка програм, які збирають статистику користувачів для встановлених веб-сайтів. Серед них виділяється Google Analytics. У своїй реалізації Google Analytics значною мірою покладається на вже створені продукти: пошукову систему Google і Google API. З цієї причини програма має найширший спектр користувацьких метрик, які можна відстежувати.

Всі інші програми, що існують сьогодні у світі, базуються на інших принципах і не мають такої ж функціональності, як Google Analytics. Недоліком створення такої програми є те, що його потрібно створювати окремо для кожної платформи. Програми, які збирають інформацію про користувачів, більш детально розглядаються нижче:

Google Analytics

Google Analytics може збирати дані різними способами, найпоширеніший з яких - за допомогою коду відстеження Google Analytics, який найчастіше реалізується за допомогою Google Tag Manager. Дані також можна збирати з HTML-сторінок, прискорених мобільних сторінок і мобільних програм. Наступний крок - обробка даних. Це налаштування, що використовуються програмою для відстеження інформації, яку хоче отримати користувач. Програма також має функцію звітування. Доступ до цих даних можна отримати через веб-інтерфейс або API Google Analytics, а також використовувати інші

інструменти, такі як таблиці Google Data Studio. Якщо коротко, то архітектуру програми можна описати чотирма основними компонентами (рис. 1.2)

- модуль збору даних
- модуль обробки даних
- модуль конфігурації пошуку даних
- модуль для відображення статистики користувачів. Програма має можливість налаштовувати спосіб відображення звітів, використовуючи вбудовані в Google Analytics опції, такі як кастомні звіти та інформаційні панелі.

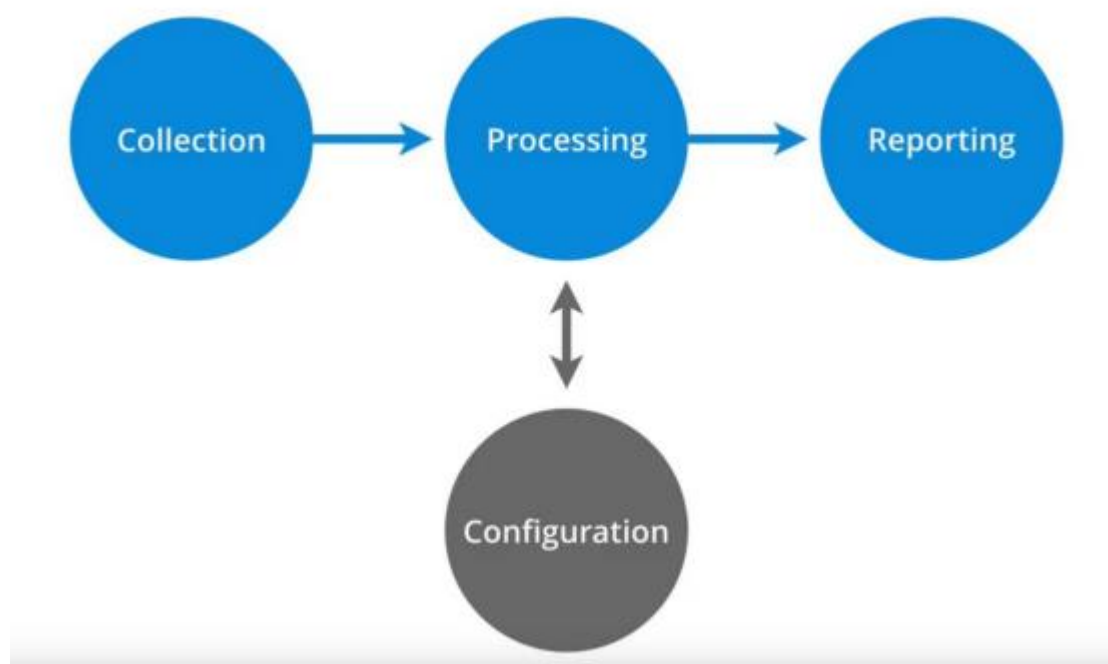


Рис. 1.2 Основні функції Google Analytics

Більшість функцій, пов'язаних зі збором даних, включені в Google Tag Manager. Він повинен бути встановлений на кожній сторінці вашого сайту.

Google Analytics відстежує низку різних метрик користувачів, які можна класифікувати по-різному. На рисунку 1.1.3 показано меню основних метрик, пов'язаних з активністю користувачів на сайті.

Однією з основних метрик програми є кількість відвідувань веб-сайту. Це кількість користувачів, які відвідали сайт. Наприклад, тут ви можете побачити "органічний пошук". Це користувачі, які прийшли на сайт через пошукову

систему Google, і ви можете побачити кількість сесій, тобто кількість відвідувань.

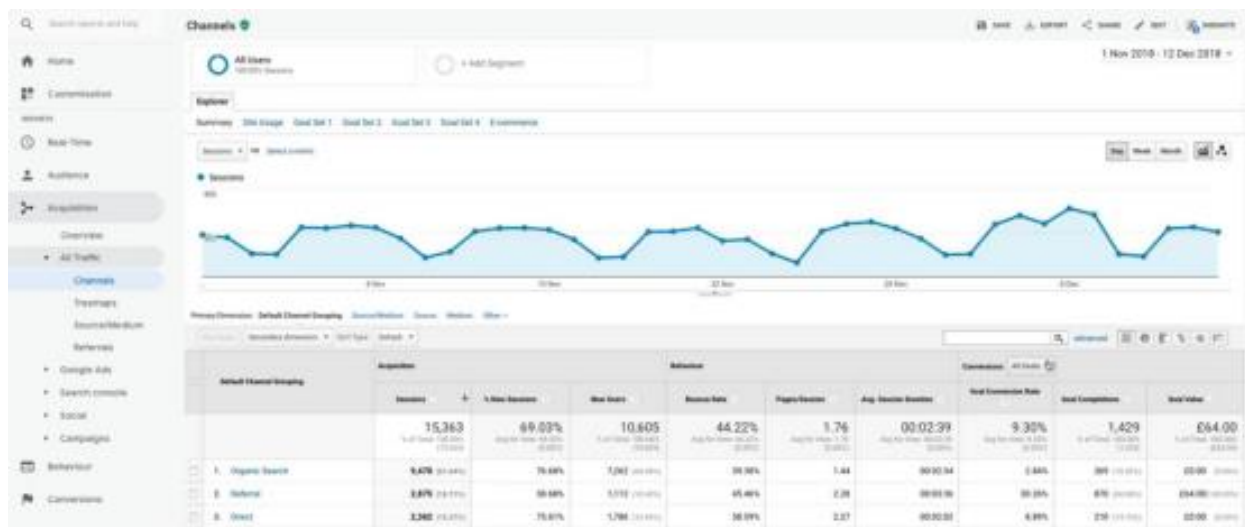


Рис. 1.3 Головне меню Google Analytics

Програма дозволяє побачити відсоток повторних сеансів.

Ви також можете побачити, скільки користувачів прийшло з веб-сайтів і соціальних мереж, які співпрацюють з вашою реферальною програмою.

Загалом, Google Analytics - це, безумовно, найбільша програма у цій галузі. Його успіх полягає в тому, що велика кількість функцій реалізована завдяки використанню Google API і пошукової системи Google, найбільшого продукту компанії, яка створила ця програма.

Перевагою цієї програми є те, що він надає різноманітну статистику щодо дій користувачів.

Недоліком цієї програми є те, що зібрана статистика подається лише у зручній формі і не містить висновків. Це робить використання цієї програми досить складним і вимагає певних людських ресурсів для роботи з ним.

Hotjar

Ця програма має іншу мету. Основна увага приділяється візуальному відображенню поведінки користувача на сайті. Основна відмінність цієї програми від інших полягає в тому, що вся інформація представлена у вигляді карти кліків. Багато підприємців використовують ця програма; Hotjar наразі

перебуває на стадії бета-розробки, але вже зарекомендував себе як популярний комерційний веб-сервіс.

Основними показниками, які надає програма, є

- карти кліків користувачів (рис. 1.4);
- карта положення курсору користувача
- карта прокрутки користувача (Рис 1.5)

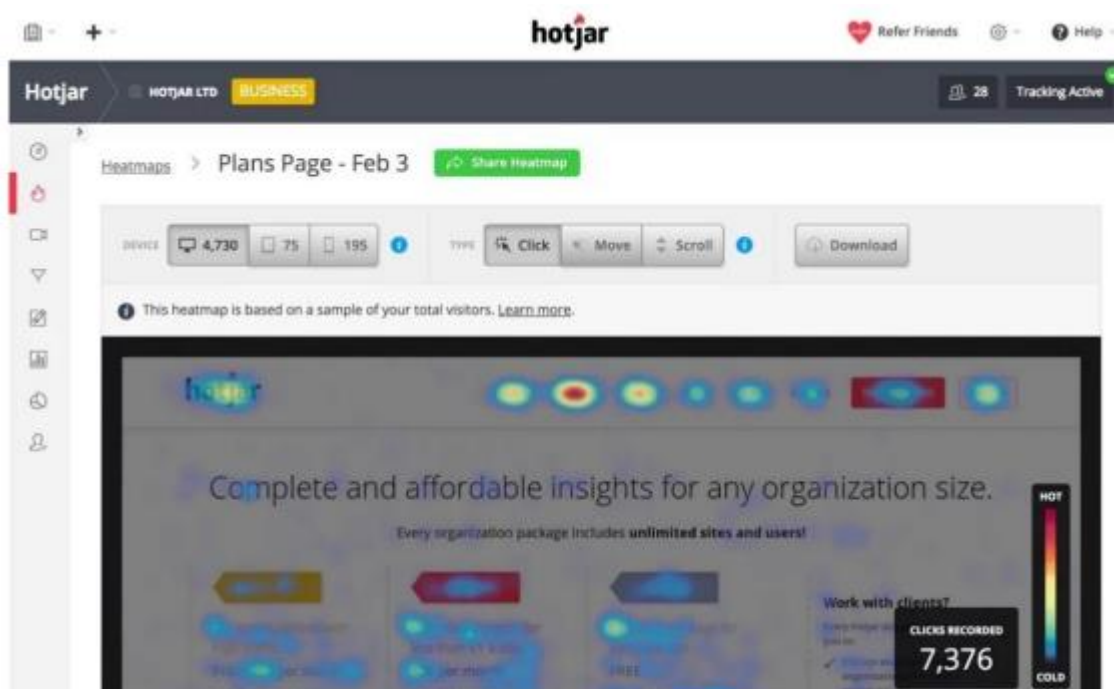


Рис. 1.4 Мапа кліків користувачів



Рис. 1.5 Мапа скролів користувачів

SmartLook

Цей сервіс дозволяє привілейованим користувачам (адміністраторам, консультантам) відстежувати поведінку будь-якого користувача на сайті.

SmartLook надає консультантам можливість

- бачити онлайн, що користувачі пишуть у текстових полях на сайті;
- відстежувати переміщення курсору на сайті
- автоматично створювати відеозаписи того, що користувачі робили на веб-сайті; та

- створювати карту кліків на сайті.

До недоліків можна віднести:

- сервіс не збирає і не генерує статистику відвідувань, здійснених самими користувачами.

Сервіс не збирає і не генерує статистику відвідувань, здійснених самими користувачами;

- інтереси.

Не існує алгоритму для розрізнення звичайних користувачів;

- сервіс не надає можливості спілкування в чаті.

Сервіс є джерелом статистики активності користувачів. Однак, через недостатню автоматизацію роботи, для успішної реалізації цієї програми необхідна низка консультантів, які б відстежували, чи змогли користувачі знайти необхідну інформацію та функції веб-сервісу; за допомогою AngularJS, React та Ember, AJAX підходу, було створено створюється запис активності користувача в режимі онлайн. Веб-сторінки користувача зберігаються віддалено [6]. Особливості цієї програми полягають у наступному:

- можливість записувати відеозаписи дій користувача;
- можливість підтримувати до 100 000 запитів на секунду;
- оскільки програма є веб-сервісом, операційна система, яку використовує консультант, не впливає на його роботу.

SmartLook був обраний для великомасштабних комерційних послуг. Його головною перевагою є здатність підтримувати високонавантажені системи, що стало можливим завдяки використанню бази даних PostgreSQL. Сервіс встановлюється на більшість сучасних CMS на базі PHP.

AWstats

Дана програма є open-source застосунком, що означає, що він є безкоштовним і його програмний текст можна змінювати. Дана програма є найпопулярнішим вибором ІТ-фахівців у світі. Оскільки ця програма є Open Source, окрім розробників, велику кількість функціональних можливостей вже дописали інші користувачі. Обсяг можливостей даної програми постійно зростає. Дана програма є найбільшим конкурентом Google Analytics оскільки надає повну статистичну інформацію користувацьких дій.



Рис. 1.6 Головна сторінка адміністраторського модуля AWStats

Основні метрики AWStats:

- кількість відвідувачів за обраний проміжок часу;
- кількість унікальних відвідувачів за обраний проміжок часу;
- кількість afk-сеансів;
- основні канали приходу користувачів на даний веб-сайт;
- drop rate по кожній сторінці веб-сайту;
- середній час сесії користувача на веб-сайті;
- кількість кліків за годину або інший обраний проміжок часу.

Перевагою даної програми є те, що він безкоштовний і Open Source. Через це нові функціональні можливості створюються великою кількістю розробників у швидкому темпі.

Недоліком даної програми є те, що не пропонується жодних висновків у зібранні користувацькі статистиці. Оскільки дана програми є open-source, він має найкращий дизайн і подає інформацію не в читабельному вигляді.

1.2.4 Етапи проведення А/В тестування

Етап 1. Висунути гіпотезу

Щоб провести тестування, на цьому етапі треба чітко сформулювати гіпотезу та визначити, як ми вимірюватимемо результат.

Формулюємо своє бачення про одну, максимум дві зміни, які покращать коефіцієнт конверсії сторінки та її продуктивність.

Припустимо, зміна кольору призведе до того, що кнопка більше впадатиме в очі і по ній почнуть частіше клікати. За збільшенням кількості кліків ми зрозуміємо, що зміна пішла на користь.

Етап 2. Створити варіанти сторінки

У нас є контрольна сторінка А із червоною кнопкою та перевірна сторінка Б — із зеленою. Та ж логіка, якщо тестуєте розміщення політики повернень: сторінка А в незмінному вигляді, з посиланням у нижньому меню, сторінка Б з новим розташуванням посилання, збоку.

Якщо вам потрібно протестувати кілька змін, не застосовуйте їх усі на одній-єдиній сторінці Б. Це призведе до плутанини — ви не зможете визначити, які зміни виявилися ефективними. Краще розбити процес: спочатку тестуйте одну зміну, потім другу.

Етап 3. Розділити вхідний трафік

Для спліт-тестування ділимо трафік порівну між варіантом Б та вихідною сторінкою А. Рис1.7

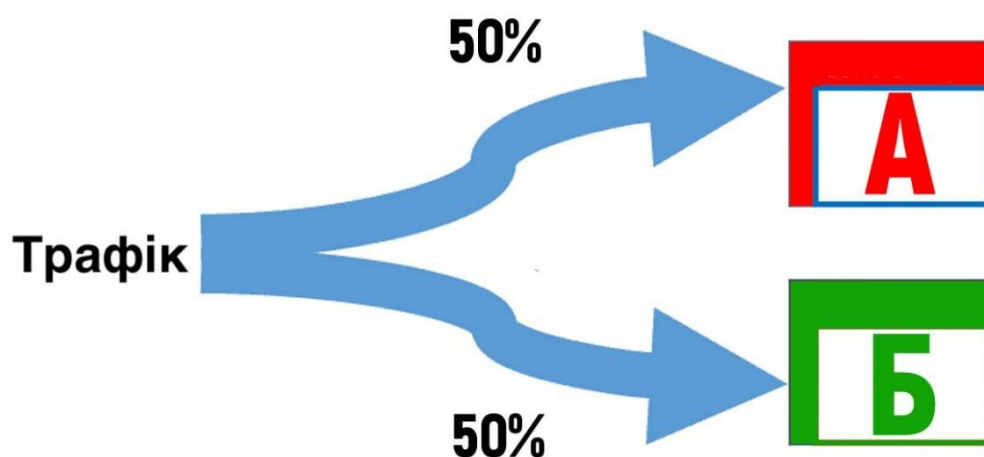


Рис. 1.7 Ділимо трафік між сторінками на рівні частини

Щоб керувати трафіком в ході А/В-тесту, використовують різні інструменти спліт-тестування. Якщо трафік йде до вас через платну рекламу або пошукові системи, можна скористатися веб аналітикою. Наприклад, проведіть А/В-тестування з Google Analytics, якщо їх лічильники встановлені у вас на сайті. У Google Analytics є окремий функціонал, повністю присвячений спліт-тестуванню сторінок — Google Optimize.

Етап 4. Впровадження та виконання тесту:

Запуск тестування, де контрольна група отримує оригінальний варіант, а експериментальна група отримує нові зміни. Під час проведення тесту корисно визначити, яким чином ви будете збирати дані. Основні метрики для аналізу ефективності тесту включають конверсію (наприклад, кількість кліків на певну кнопку), час, який користувач проводить на сторінці, та інші параметри, які безпосередньо пов'язані із цілями тесту.

Етап 5. Аналіз результатів

Статистичний аналіз є ключовим етапом у визначенні ефективності проведеного А/В тестування. Цей етап включає в себе використання статистичних методів для оцінки різниці між контрольною та експериментальною групами з метою визначення, чи є ця різниця статистично значущою.

Важливо враховувати, що статистична значущість не завжди означає практичну значущість. Отже, аналіз має бути комплексним, з урахуванням не тільки числових даних, але й контексту веб-сайту та його цілей.

Етап 6. Висновки та впровадження

Засновані на аналізі висновки визначаються з урахуванням результатів та їх відповідності меті тестування. Цей етап включає в себе:

Оцінку впливу: Визначення того, як зміни вплинули на користувацький досвід та досягнення цілей.

Прийняття рішення: Визначення того, чи слід впроваджувати зміни на постійній основі на основі результатів тестування.

Оптимізація стратегій: Врахування вивчених уроків для подальшої оптимізації стратегій розвитку веб-інтерфейсу.

Висновки та рішення, прийняті на цьому етапі, формують основу для подальших кроків в удосконаленні веб-сайту та його користувацького досвіду.

1.3 Моделі якості інтерфейсів

Модель якості інтерфейсу користувача є важливим інструментом для оцінки та покращення користувацького досвіду веб-сайту. Тут ми розглянемо два важливі аспекти: оцінку користувацького досвіду (UX) та параметри взаємодії, які впливають на якість UX.[15]

Користувацький досвід (UX) - це сукупність почуттів, вражень і взаємодій, які користувачі мають з веб-сайтом, маючи на увазі, що UX не обмежується дизайном інтерфейсу, але також включає в себе ефективність, доступність і загальну задоволеність від використання веб-ресурсів. Важливо пам'ятати, що UX також включає в себе ефективність, доступність і загальну задоволеність від використання веб-ресурсів.

UX метрики:

Час взаємодії: середній час, який користувач витрачає на взаємодію з веб-сайтом. Коротший час може свідчити про ефективність інтерфейсу.

Кількість кліків для досягнення мети: мінімізація кількості кліків для досягнення певної мети свідчить про зручність навігації.

Коефіцієнт прямого повернення: визначте, який відсоток користувачів відмовляється взаємодіяти з сайтом. Нижчий відсоток свідчить про вищий рівень задоволеності користувачів.

Аспекти, що впливають на UX

Дизайн і ергономіка: привабливий, зрозумілий дизайн і увага до робочого процесу користувача покращують UX.

Взаємодія та навігація: зручні системи навігації та інтуїтивно зрозуміла взаємодія допомагають користувачам ефективно використовувати ресурси.

Продуктивність: швидкість завантаження сторінок і реакція інтерактивних елементів впливають на досвід використання веб-сайту.

Опції взаємодії включають елементи, які визначають, як користувачі взаємодіють з веб-сайтом. До них відносяться кнопки, форми, меню, гіперпосилання та інші елементи, які полегшують взаємодію.

Вплив параметрів взаємодії на якість інтерфейсу

Зручність використання: простота і зрозумілість взаємодії, включаючи розташування і дизайн елементів, сприяє приємній взаємодії.

Ефективність: вимірює, наскільки швидко та ефективно користувач може виконати певне завдання завдяки взаємодії з інтерфейсом.

Помилки взаємодії: мінімізація ймовірності виникнення помилок під час взаємодії, щоб їх можна було швидко виправити.

Аспекти оптимізації параметрів взаємодії

Адаптивний дизайн: спрощення та оптимізація взаємодії на різних пристроях для покращення загального користувацького досвіду.

Тестування користувацького досвіду: проведення тестів, спрямованих на виявлення найефективніших параметрів взаємодії.

1.4 Висновки до розділу

У цьому розділі описано основні подібності веб-сервісів, що надають статистику активності користувачів на сайті, на якому встановлено веб-сервіс.

Основним недоліком є те, що такі програми необхідно створювати окремо для кожної платформи. На особливу увагу заслуговує програма Google Analytics, у функціоналі якого використовуються продукти, створені компаніями-розробниками. Перевагою розглянутих програм є велика кількість статистичних даних, що відображаються у відстежуваному і зручному графічному форматі.

Головний недолік усіх розглянутих вище застосунків полягає в тому, що жоден із них не надає можливості аналізу та виведення зібраної статистики. У бізнесі це означає, що власнику сайту доведеться наймати ще одного

співробітника для аналізу статистики, що надається цими програмами. Робота таких співробітників несистематична, неорганізована і неефективно затратна.

РОЗДІЛ 2 ОБГРУНТУВАННЯ МЕТОДІВ ДОСЛІДЖЕННЯ

Для дослідження проблемної області буде створено віконну програму.

2.1 Основні функціональні вимоги

- збирання даних про активність користувачів: дослідження повинно охоплювати здатність системи фіксувати та записувати активності користувачів на фронтенді веб-сайту, зокрема події кліків, переходів, заповнення форм тощо;
- випадкова варіативність версій сайту: необхідно дослідити функціонал, що дозволяє направляти користувачів на випадкові версії інтерфейсу сайту (варіанти А і В), забезпечуючи рівномірний розподіл трафіку між версіями;
- обробка даних у базі даних: потрібно проаналізувати ефективність запису даних у базу, включаючи структурування даних про користувачів, варіативність версій та результати взаємодій;
- обчислення конверсій: дослідження має охоплювати механізм обробки даних у Python-скрипті для аналізу конверсій різних версій сайту на основі даних, отриманих з бази даних;
- захист даних: має бути досліджена система захисту даних користувачів, включаючи безпечний запис, зберігання та доступ до інформації про активності.

Ці вимоги до функціональних характеристик дослідження якості інтерфейсів за допомогою А/В тестування допоможуть зрозуміти ефективність підходу, вплив різних версій інтерфейсу на користувацький досвід і бізнес-показники та сформулювати рекомендації для їх оптимізації.

2.2 Вхідні данні

Вхідні дані для веб-програми, що використовує А/В тестування:

- дані про активності користувачів: структура даних, що зберігаються в базі, включаючи події користувачів (кліки, перегляди сторінок), час подій,

- унікальні ідентифікатори користувачів, а також інформацію про версію інтерфейсу (А або В);
- обсяг даних: кількість подій користувачів, які планується записати, обсяг інформації в контексті тестування (наприклад, тисячі чи мільйони подій за певний період часу);
 - методи доступу до даних: опис способів запису, зчитування та аналізу даних, включаючи автоматизований запис подій з фронтенду, вибіркові запити до бази даних для аналізу;
 - сценарії використання: визначення сценаріїв, які будуть протестовані, наприклад, додавання нових варіантів інтерфейсу, відстеження поведінки користувачів у кожному варіанті, аналіз взаємодії з ключовими елементами сторінки;
 - налаштування серверів та баз даних: конфігурація серверів, що використовуються для збереження даних, включаючи обсяг доступної пам'яті, швидкість обробки запитів та інші технічні параметри.

2.3 Вихідні дані

Вихідні дані для аналізу якості інтерфейсів:

- результати А/В тестування: звіт із результатами порівняння варіантів А і В, включаючи метрики конверсії, час взаємодії користувачів з елементами інтерфейсу, показники відмов та інші ключові показники ефективності;
- рекомендації щодо оптимізації: детальний аналіз, який інтерфейс є більш ефективним, із поясненнями, які елементи покращують користувацький досвід, та вказівкою на можливі напрямки для подальшого тестування;
- план впровадження: практичні рекомендації для інтеграції результатів А/В тестування в робочий процес, включаючи можливості автоматизації тестування, оновлення інтерфейсу та безперервного аналізу.

Висновки до розділу 2

Другий розділ охоплює основні функціональні вимоги до системи А/В тестування, а також вхідні та вихідні дані веб-програми. Основні функціональні вимоги включають запис та обробку даних про активності користувачів, що надходять із фронтенду веб-програми, забезпечення рандомної варіативності версій інтерфейсу для рівномірного розподілу трафіку, структурування даних у базі для ефективного аналізу, обчислення метрик конверсії за допомогою Python-скриптів і забезпечення захисту даних користувачів.

До вхідних даних належать опис структури даних, що зберігаються у базі, обсяг інформації для аналізу, методи доступу до бази, сценарії використання системи, а також конфігурації серверного середовища, необхідні для забезпечення коректної роботи програми.

До вихідних даних відносяться результати тестування з аналізом ефективності кожного варіанту інтерфейсу, порівняльний аналіз варіантів А та В, рекомендації щодо оптимізації інтерфейсу та план інтеграції отриманих результатів у подальший розвиток веб-програми.

Розділ формує цілісне уявлення про ключові аспекти дослідження, необхідні для забезпечення якості інтерфейсів та ефективності їх тестування.

РОЗДІЛ 3 ДОСЛІДЖЕННЯ ЯКОСТІ ВЕБ-ІНТЕРФЕЙСА ЗА ДОПОМОГОЮ СИСТЕМИ ДЛЯ А\В ТЕСТУВАННЯ НА ОСНОВІ ОБРАНОГО САЙТУ

3.1 Методологія дослідження

Незалежно від предметної області, процес ухвалення інтелектуальних рішень можна представити як послідовність таких основних етапів [7, 8, 22]:

- формулювання й аналіз цілей
- вибір можливих рішень (альтернатив), що забезпечують досягнення цілей
- вибір заходів (оцінка), які дають змогу якісно або кількісно оцінити перевагу або неповноцінність (ефективність) альтернатив Система критеріїв) (етап оцінки)
- вибір крайніх рішень відповідно до заданих критеріїв оцінки (етап оптимізації).

Перші два етапи можуть бути виконані з використанням різних методів системного аналізу. Етап оцінки в ланцюжку, навпаки, є одним із найважливіших етапів, але його складно формалізувати. Це пов'язано з тим, що в найпростіших випадках часто можна обґрунтувати єдиний скалярний критерій ефективності, який адекватно характеризує альтернативи [17,20]. Формально це означає, що експертам зазвичай доводиться оцінювати ефективність альтернатив на основі аналізу деякого набору суперечливих критеріїв, кожен з яких характеризує часткову характеристику.

Крім того, ці часткові критерії мають різні розмірності, значущість, шкали виміру, інтервали можливих значень і домінуючі орієнтації. Такі задачі називаються задачами багатокритеріального оцінювання.

3.2 Вибір сайту для дослідження

У дослідженні брали участь інтернет-магазини, що спеціалізуються на продажу одягу. Загальна відвідуваність сайту становить 100-170 відвідувачів на місяць. Сайт був розроблений на Nextjs, а як бекенд використовувався Nestjs.

Серед особливостей інтерфейсу - зручна взаємодія з каталогом, інтуїтивно зрозумілий інтерфейс і можливість переходу на темну тему.

3.3 Опис системи А/В тестування, що використовується

Розроблена система А/В-тестування використовує базу даних MySQL, а всі записи метрик інтегровані в сайт за допомогою React і TypeScript А/В-тестування - це складний і важливий процес, спрямований на поліпшення різних аспектів роботи інтернет-магазину. Розроблена система тестування включає в себе кілька етапів:

1. визначення цілей тесту
2. вибір змін для тесту
3. створення варіантів
4. розподіл трафіку
5. запуск тесту
6. аналіз результатів
7. ухвалення рішення і впровадження

3.3.1 Визначення мети тестування

Оскільки у цього інтернет-магазину невелика база постійних клієнтів, головна мета - збільшити їхню кількість.

По-друге, відстежували кількість кліків покупців, унаслідок чого з'ясувалося, що багато клієнтів кладуть товари в кошик, але не здійснюють покупку. Таким чином, ще однією метою тесту було підвищення коефіцієнта конверсії магазину.

3.3.2 Вибір змін для тестування

Якщо судити з практики інших популярних магазинів і подивитися на карту кліків покупців, слайдер із зображенням виробників мав змінитися, а покупці або не помітили його, або відволіклися на слайдер «Бестселери».

Після видалення слайдера з виробниками було вирішено замінити слайдери «Бестселери» і «Нові товари». Це пов'язано з тим, що більшість покупців

переходять у каталог через верхнє навігаційне меню, а не через слайдери. Досвідчені маркетологи також радять додати на головну сторінку «нові продукти» та «кампанії», щоб підтримувати інтерес користувачів.

3.4 Розробка тестових варіантів

Після вибору змін і розробки тестової сторінки було додано можливість розподіляти клієнтів між двома версіями клієнта. Вибір версії сайту призначається на сторінці реєстрації. Користувач не може знати, яку версію сайту він використовує, поки не перейде в меню «Вивчити елемент», де він побачить призначену йому версію сайту (рис. 2.1).

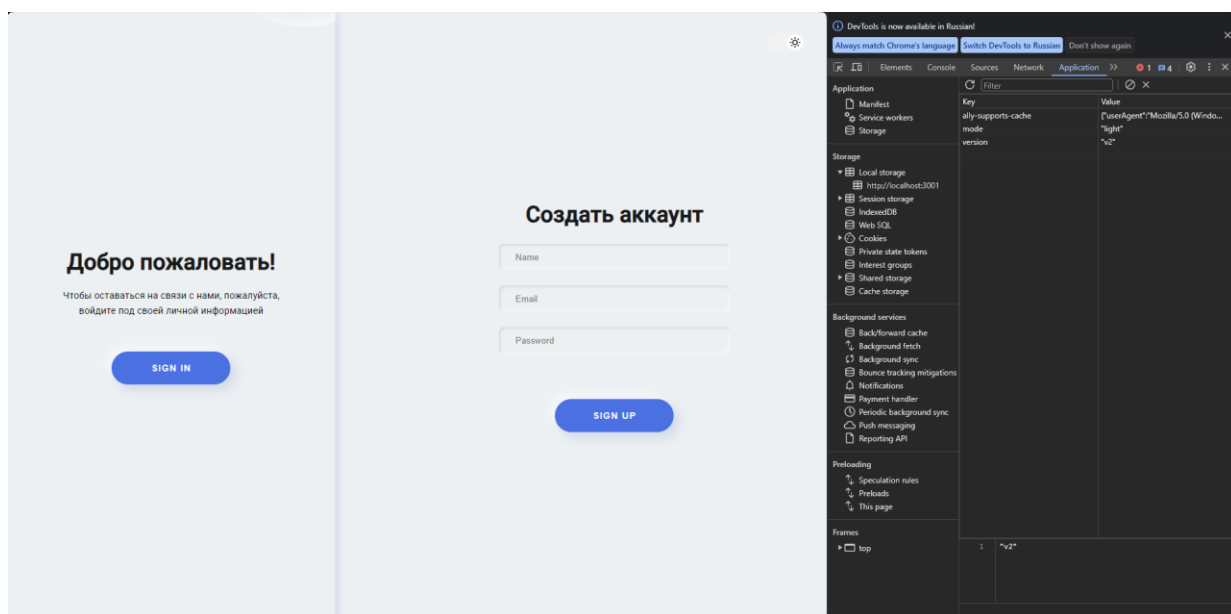


Рис. 2.1 Сторінка реєстрації та «дослідити елемент»

По-перше, було додано перевірку кнопки зміни теми. Це означає, що коли користувач обирає темну або світлу тему, ця дія записується в базу даних і можна порівняти, скільки людей використовують ту чи іншу тему. Коли користувач додає товар у кошик, у більшості випадків він не переходить до оплати і продовжує користуватися сайтом, тому ми додали повідомлення на головній сторінці про те, що товар у кошику доступний (Рис 2.2).

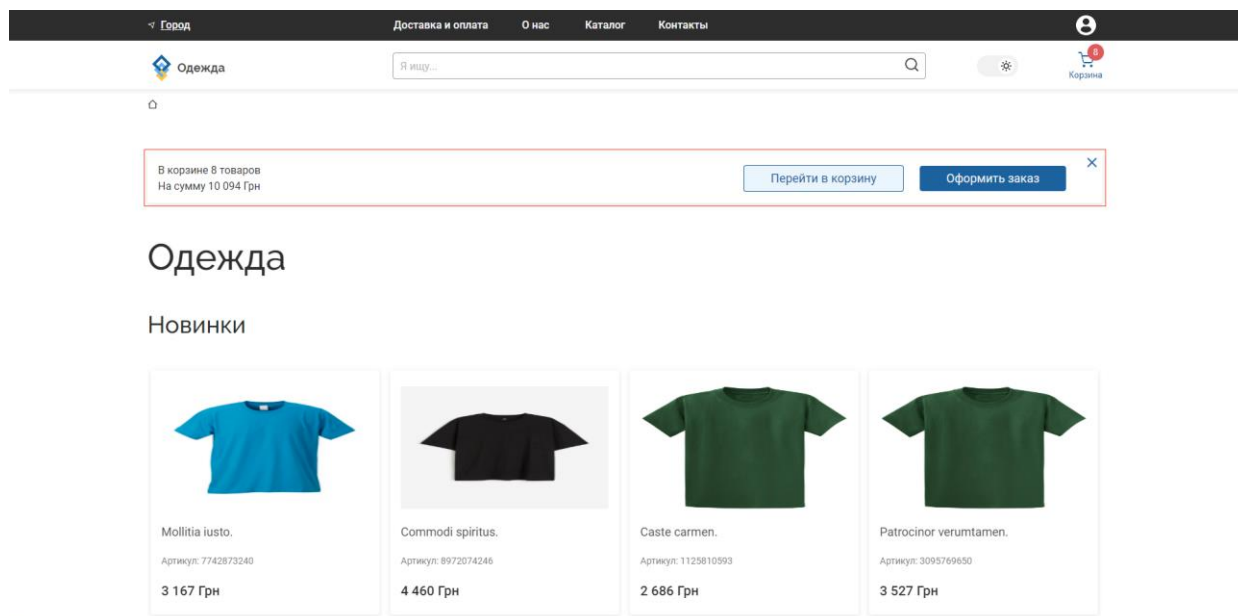


Рис 2.2 Сповіднення про наявність товару в корзині

Деяким покупцям доводилося робити додаткове клацання, щоб видалити деякі категорії після вибору деяких категорій у фільтрі. Наприклад, один із покупців вибрав категорію «Одяг», «Стать» і «Товари», і щоб видалити товар із цього фільтра, йому довелося натиснути на кнопку, щоб розгорнути одну з категорій, а потім натиснути на прапорець у фільтрі, щоб видалити його. Тому, щоб полегшити і спростити навігацію, на вкладку каталогу додали більш зручну навігацію за фільтрами.

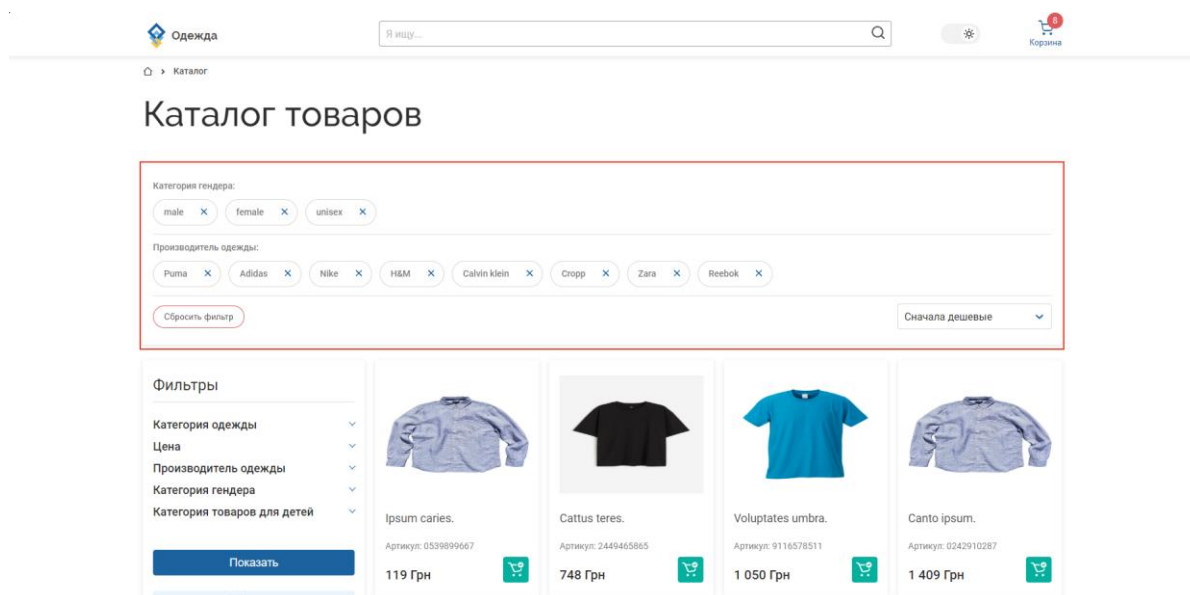


Рис 2.3 Зручна навігація по фільтрам

Змінено місцями слайдери з новинками та лідери продаж та видалено вікно з марками виробників(Рис 2.4)(Рис 2.5)

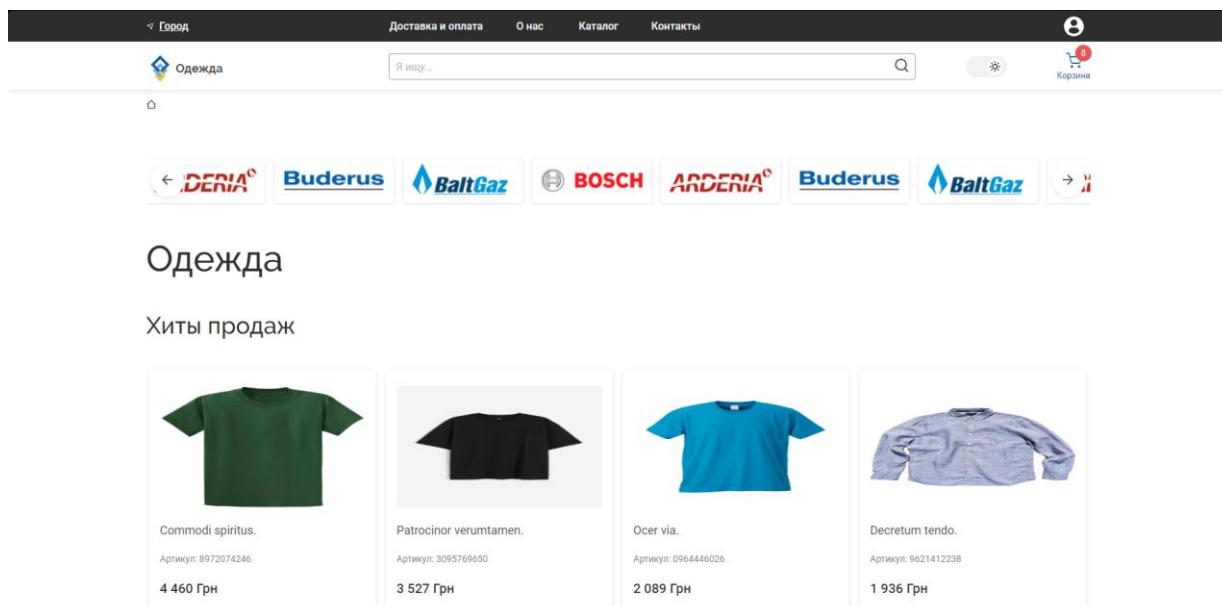


Рис 2.4 Головна сторінка оригіналу сайту

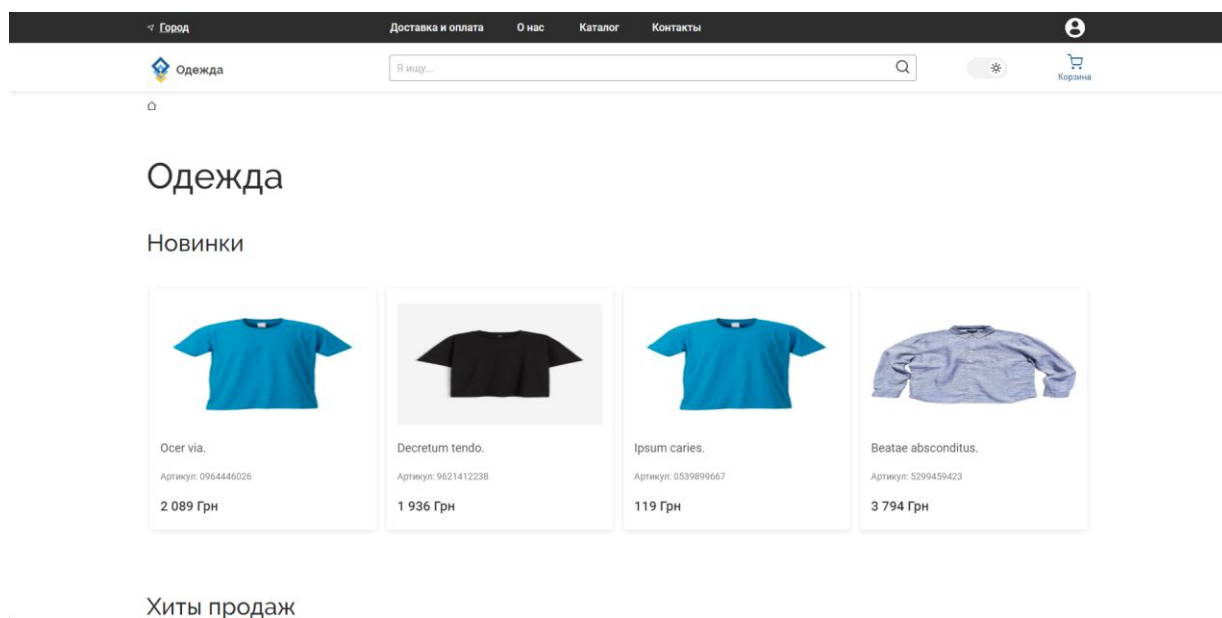


Рис 2.5 Головна сторінка тестового варіанту

3.5 Аналіз отриманих результатів

Для перевірки ефективності програмного забезпечення та аналізу результатів було зроблено таке: 107 користувачам було надано два тестових стенди: перший - основна програма без змін, друга – програма зі змінами. Фронтенд-частини обох програм були написані з використанням React.JS. Тестування проводилося у два етапи.

На першому етапі тестувався інтерфейс без будь-яких змін на сайті. На першому етапі було зібрано банк даних для аналізу. Після отримання результатів

було внесено зміни в інтерфейс сайту, щоб зробити його доступним для користувачів.

Для полегшення аналізу даних було розроблено застосунок, що допомагає користувачам зрозуміти, як порівнюються результати протестованих варіантів, який варіант кращий і чи вважаються результати значущими (рис. 2.6).

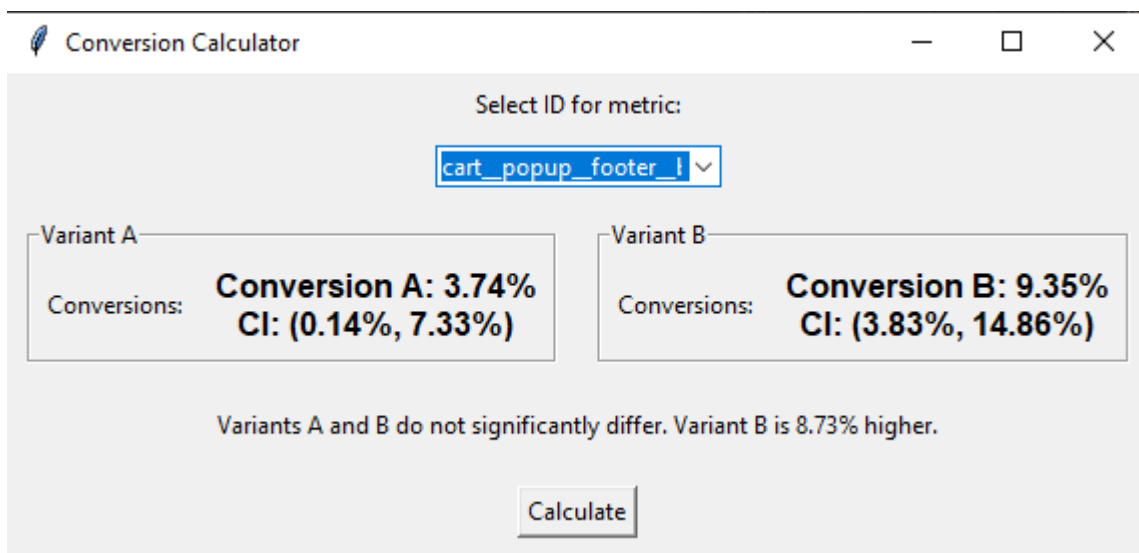


Рис 2.6 Приклад роботи програми

Підрахувавши усі дані, їх можна занести у Табл. 4:

Таблиця 4 Результат розрахунку даних

Метрики	Конверсія %(до /після)	Конверсійна ставка %
Оформили замовлення	3.74 / 9.35	Збільшилась на 8.73
Видалили товар з корзини	3.74/ 1.87	Зменшилась на 3.49
Перейшли до каталогу через новинки	17.76/ 32.71	Збільшилась на 30.57
Перейшли за посиланням	5.61/7.48	Збільшилась на 6.99
Скинули фільтр(старий)	0.93/2.80	Збільшилась на 2.62
Скинули фільтр(новий)	0.00/3.74	Збільшилась на 3.49
Видалили налаштування фільтра(новий)	0.00/8.41	Збільшилась на 7.86

Категорії речей	25.23/17.76	Зменшилась на 23.58
-----------------	-------------	---------------------

3.6 Рекомендації для поліпшення якості інтерфейсу веб-сайту

Підвищення якості користувацького інтерфейсу вашого сайту може значно поліпшити враження користувачів і забезпечити більш ефективну взаємодію з відвідувачами. Нижче наведено загальні рекомендації:

- зберігайте простоту і зрозумілість: зробіть навігацію простою та інтуїтивно зрозумілою для користувача.

- зберігайте просту структуру сторінок і меню.

- використовуйте чіткі та лаконічні назви для меню та елементів інтерфейсу.

Адаптивність:

- використовуйте адаптивний дизайн, щоб забезпечити оптимальне відображення сайту на різних пристроях.

- переконайтеся, що сайт ефективно працює на мобільних пристроях.

Швидкість завантаження:

- оптимізуйте зображення та інші ресурси для прискорення завантаження сторінок.

- використовуйте кешування, щоб скоротити час завантаження під час повторних відвідувань.

- зробіть важливу інформацію легко читабельною.

- розміщуйте важливу інформацію на видному місці.

- використовуйте великі заголовки, що вказують на основний зміст сторінки.

Високоякісні зображення та графіка

- використовуйте високоякісні зображення та графіку.

- виберіть колірну палітру, яка відповідає вашому бренду і приємна для очей.

Використовуйте новітні технології:

- використовуйте анімацію і переходи, щоб зробити ваш сайт більш захоплюючим.

- використовуйте новітні технології, такі як AJAX, для поліпшення взаємодії без перезавантаження сторінки.

Тестування юзабіліті:

- проводьте тести юзабіліті та отримуйте відгуки від реальних користувачів.
- проаналізуйте, як користувачі взаємодіють з вашим сайтом, і поліпшіть його відповідно до цього.

Інтерактивність:

- додайте форми, кнопки, анімацію та інші інтерактивні елементи, щоб залучити відвідувачів.
- розгляньте можливість використання чат-ботів та інших інтерактивних засобів спілкування.

SEO-оптимізація:

- встановіть релевантні мета теги та ключові слова, щоб поліпшити індексацію в пошукових системах.
- створіть зручні URL-адреси, щоб полегшити пошукову оптимізацію.

Функціональність і надійність:

- переконайтеся, що всі функції сайту працюють правильно і без збоїв.
- регулярно оновлюйте контент і перевіряйте всі внутрішні посилання.

Рекомендації щодо поліпшення протестованого сайту:

1. змінити форму і колір нового меню фільтрів, оскільки, згідно зі статистикою, коефіцієнт конверсії не дуже високий;

2. видалити старий інтерфейс для скасування категорій фільтрів;

3. створити нове меню фільтрів, більш зручне і просте у використанні. Таким чином, ви зможете побачити, скільки людей користуються фільтрами, і, можливо, вам знадобиться поліпшити його.

4. Поліпшення роботи з фільтрами Важливо не тільки враховувати результати тестування, а й постійно вдосконалювати інтерфейс на основі поведінки користувачів та їхніх відгуків. Під час внесення змін слід також враховувати галузеві тенденції.

Висновки до розділу 3

Дослідження якості веб-інтерфейсу за допомогою системи А/В-тестування на основі відібраних веб-сайтів виявило низку важливих аспектів. Застосування системи А/В-тестування на базі обраних сайтів дало змогу провести глибокий аналіз і оцінку різних варіантів дизайну та функціональності. Дослідження підтвердило, що система А/В-тестування є ефективним інструментом для виявлення оптимальних рішень, спрямованих на поліпшення користувацького досвіду. Аналіз взаємодії користувачів із різними варіантами інтерфейсу допоміг зрозуміти, які зміни позитивно вплинуть на показники продуктивності та задоволеності користувачів.

Використання А/В-тестування також виявилось корисним під час визначення оптимальних стратегій для підвищення залученості аудиторії та конверсії. Аналіз зібраних даних і результатів тестування дав змогу зробити обґрунтовані висновки про те, які зміни слід впровадити для досягнення поліпшень.

Отримані результати свідчать про ефективність використання системи А/В-тестування під час вивчення та оптимізації веб-інтерфейсу на основі обраного сайту. Такий підхід знижує ризик невдачі, надаючи фактичну основу для ухвалення рішень про те, як поліпшити веб-сайт і забезпечити задоволеність користувачів.

РОЗДІЛ 4 ПОБУДОВА БІЗНЕС МОДЕЛІ

4.1 Оцінка проблеми

Сьогодні багато комерційних операцій проводяться з використанням електронних джерел, як у всьому світі, так і в Україні. Електронні джерела також використовуються державними органами та податковими інспекціями.

Під час створення нового комерційного веб-сервісу підприємці стикаються з низкою проблем, як-от створення інтуїтивно зрозумілого інтерфейсу, цілісність функціоналу та підвищення загальної якості роботи застосунку і власного персоналу.

Вирішити ці проблеми можна за допомогою аналізу статистичних даних. З цієї причини у світі почали з'являтися різноманітні мікросервіси та доповнення до веб-застосунків для збору різної статистичної інформації та поліпшення взаємодії з клієнтами. Такі сервіси пропонують такі функції, як вбудовування живого чату на сайт, моніторинг поведінки відвідувачів, картування кліків відвідувачів і відстеження того, як користувачі знаходять сайт. Існує безліч подібних програм, створених виключно для підвищення продуктивності комерційних веб-програми. Однак проблема цих сервісів полягає в їхній розрізненості: не існує єдиного рішення, яке втілювало б у собі всі функції, що розв'язують перелічені вище завдання. Саме тому варто звернути увагу на розробку подібних програм.

Ще одним фактором, що істотно впливає на комерційний успіх веб-сервісу, є його зовнішній вигляд, тобто інтерфейс і дизайн. Найпоширеніші шаблони інтерфейсів створюються з метою наслідування зовнішнього вигляду успішних високобюджетних програм. Наприклад, одним із найбільших інтернет-магазинів в Україні є Rozetka, і більшість інтернет-магазинів в Україні намагаються скопіювати його зовнішній вигляд, змінюючи лише шрифти, колірні схеми та інші деталі дизайну. Це означає, що інтерфейс веб-програм може бути формалізовано, а формалізовану модель можна проаналізувати щодо різних

характеристик і метрик, які можуть поліпшити зовнішній вигляд веб-програм шляхом відстеження статистики користувачів [21].

Крім відстеження часу між кліками користувачів на сайті, карти кліків користувачів можуть надати інформацію про те, які елементи користувацького інтерфейсу розташовані невдало. Наприклад, якщо елемент призначеного для користувача інтерфейсу, який найчастіше шукають, має найбільший час пошуку порівняно з іншими елементами, це означає, що цей елемент має бути розміщений на найвиднішому місці веб-сторінки.

4.2 Зацікавлені сторони

У вирішенні вищезазначених проблем беруть участь кілька зацікавлених сторін. Найбільшими зацікавленими сторонами є малі та середні підприємства, які використовують веб-сервіси у своєму бізнесі, тобто всі ті, у кого є веб-сайт для зв'язку зі своїми клієнтами. Це пов'язано з тим, що в міру підвищення якості інтерфейсів веб-застосунків у всьому світі користувачі витратять менше часу на пошук потрібної послуги, продукту тощо. ІТ-компанії також є зацікавленими сторонами, оскільки більшість веб-сервісів SME створюються ІТ-компаніями. Це пов'язано з тим, що застосунок підвищує швидкість і чіткість їхньої роботи, що, своєю чергою, збільшує продуктивність ІТ-компаній. У таблиці 5 наведено всі групи зацікавлених сторін, їхні інтереси та вплив (ступінь зацікавленості у розв'язанні наявних проблем).

Таблиця 5 Зацікавлені сторони

Зацікавлена сторона	Зацікавленість	Вплив зацікавленої сторони	Стратегії приваблення
Бізнес	Збільшення прибутку за допомогою метрики власного	Високий	Участь у спеціалізованих виставках, форумах,

	веб-сервісу. Підвищення автоматизації роботи відділів, що взаємодіють з клієнтами		конференціях, проведення презентацій для зацікавлених компаній
Клієнти	Скорочення часу пошуку бажаної послуги, товару тощо	Низький	
ІТ компанії	Збільшення час на виконання проектів	Низький	

4.3 Комерційне рішення

Відповідно до перерахованих завдань можна описати кінцевий продукт, який їх вирішує. Продукт не тільки реалізує інтерфейс для підтримання контакту між клієнтами й адміністраторами, а й використовує алгоритми для збору й аналізу метрик користувацької статистики для кожного елемента інтерфейсу на кожній сторінці веб-сервісу відповідно до вимог, описаних у попередньому розділі. Результати роботи цієї програми можна побачити через деякий час після встановлення. По-перше, зібрану статистику користувацької активності подано в зручному для читання форматі, що дає змогу заощадити час на аналіз метрик відповідними співробітниками. Це означає, що процес аналізу активності користувачів стає більш автоматизованим, що знижує витрати співробітників. По-друге, сервіс містить додаткові функції для підтримки контакту з клієнтами, такі як автоматичні чат-боти. Це означає, що власники комерційних веб-сервісів є клієнтами цього сервісу, а співпраця будується за моделлю B2B. Ця програма

має бути проста у використанні, легко встановлюватися і надавати всю необхідну інформацію співробітникам встановленого веб-сервісу.

4.4 Клієнти, сегменти ринку споживання

Для того щоб максимально підвищити ефективність роботи проектної команди, рекомендується сегментувати ринок споживачів. Тому сегментація споживчого ринку застосунку шляхом аналізу показників інтерфейсу веб-сервісу ґрунтується на статистиці користувачів і здійснюється за такими параметрами

- обсяг очікуваних відключень для конкретної веб-сторінки веб-програми;
- кількість щоденних відвідувань сайту.

Обидва критерії підходять для груп клієнтів, що представляють малий і середній бізнес, зацікавлених у відстеженні та поліпшенні вищевказаних показників.

4.4 Бізнес модель

Споживачі: підприємства, що продають товари/послуги через свої веб-сайти.

Проблема: відсутність автоматизованих рішень, висока вартість людських послуг, низька якість послуг модераторів і неточність наявних метрик для подібних послуг.

Рішення: програмне забезпечення, що надає напівавтоматизовані модулі спілкування з клієнтами і демонструє недоліки інтерфейсів, розроблених на основі аналізу активності користувачів.

Структура витрат:

- штат технічної підтримки (зарплата, соціальні виплати); - штат технічної підтримки (зарплата, соціальні виплати);
- витрати на утримання робочих місць співробітників (оренда офісу, комунальні послуги);
- податки;
- оплата послуг юристів, бухгалтерів і прибиральниць.

Бізнес-модель також включає структурні елементи:

- приховані вигоди (вигоди, які не можна скопіювати або купити);
- ключові показники (ключові показники для вимірювання);
- канали (як досягаються користувачі).
- канали: через відділ співпраці/інтеграції кожного мікросервісу.
- ключовий показник: кількість проданих ліцензій.

Приховані вигоди: значимість веб-сторінок на конкретних сайтах. Таким чином, дивлячись на таблицю 5, можна зробити висновок, що запропонований у цій роботі проєкт із впровадження методу аналізу ефективності інтерфейсу на основі відстеження поведінки користувачів має хороші перспективи для подальшої реалізації.

Звісно, цей аналіз не враховує всіх ризиків і чинників, як-от особливості системи оподаткування в країні присутності, але одних лише наявних досліджень достатньо, щоб спрогнозувати комерційний успіх продукту та його окупність.



додати назву та номер таблиці

Проблема	Рішення	Споживачі
Відсутність цілісного рішення; Висока вартість людських послуг; Недоскональність сучасних алгоритмів.	Програмне забезпечення що містить модуль зв'язку з клієнтами, а також модуль для аналізу користувацької статистики.	Компанії, що реалізують свої послуги за допомогою веб-сайтів.
	Ключові метрики	Канали збуту
Структура витрат		Потоки доходів
Утримання персоналу для надання технічної підтримки (виплат заробітних плат, соціальних виплат); утримання робочих місць персоналу.		Реклама, пошук інвесторів

4.5 Висновки за розділом 4

У цьому розділі проаналізовано поточну ситуацію в галузі збору користувацьких метрик на веб-сайтах, виявлено наявні проблеми та узагальнено відповідне дерево проблем. Разом із проблемою було визначено основні і зацікавлені сторони у вирішенні наявних недоліків, а також ступінь впливу цих сторін на розв'язання проблеми. У результаті було запропоновано комерційне рішення з конкурентними перевагами, що задовольняє інтереси зацікавлених сторін і підкреслює унікальну ціннісну пропозицію пропонованого продукту. Було проаналізовано потенційних споживачів і вивчено сегменти споживчого ринку. Таблиця 6 Бізнес модель

ЗАГАЛЬНИЙ ВИСНОВОК

У результаті виконання кваліфікаційної роботи якість інтерфейсу веб-сайту було детально проаналізовано за допомогою системи А/В-тестування. Цей метод продемонстрував свою ефективність у вивченні впливу різних змін інтерфейсу на поведінку користувачів. Збір даних про активності відвідувачів, таких як кліки, заповнення форм і переходи, дозволив глибше зрозуміти, як нові елементи дизайну впливають на взаємодію з сайтом.

Результати проведених експериментів стали основою для обґрунтованого прийняття рішень щодо оптимізації інтерфейсу. Використовуючи дані А/В-тестування, розробники змогли визначити, які зміни підвищують конверсію, сприяють утриманню аудиторії та позитивно впливають на досягнення маркетингових цілей. Це дозволило впроваджувати лише ті рішення, які дійсно відповідають потребам користувачів і приносять відчутну користь бізнесу.

Окрім підвищення конверсії, А/В-тестування виявилось корисним для постійного вдосконалення UX/UI дизайну. Завдяки цьому підходу можна поступово знижувати ризики впровадження невдалих рішень, фокусуючись лише на змінах, які сприяють покращенню користувацького досвіду. Постійний аналіз і впровадження результатів забезпечують стабільний розвиток веб-сайту, орієнтованого на потреби аудиторії.

Загалом, А/В-тестування є одним із ключових інструментів для розробників і маркетологів, який допомагає підвищувати якість веб-інтерфейсу. Цей метод забезпечує можливість приймати обґрунтовані рішення, які базуються на реальних даних, і враховувати побажання користувачів. У майбутньому використання А/В-тестування сприятиме створенню інтерфейсів, які не лише відповідають сучасним тенденціям, а й забезпечують ефективну взаємодію з аудиторією.

Таким чином, виконана робота підтверджує, що А/В-тестування є незамінним інструментом для підвищення ефективності веб-сайтів, оптимізації їхніх функціональних і дизайнерських рішень, а також досягнення поставлених бізнес-цілей.

БІБЛОГРАФІЧНИЙ СПИСОК

1. Осипов, И.В. Технические средства человеко-компьютерного взаимодействия ТUI. Обзор и анализ возможностей использования в игрофикации. / Cloud of science. — 2016. — № 4, Т.3. (дата звернення: 24.11.2023)
2. URL: https://vuzlit.com/1021195/primery_ushpeshnyh_razrabotok (дата звернення: 24.11.2023)
3. Говорущенко Т. О. Метод використання технології eye-tracking у проектуванні інтерфейсів користувача / Т. О. Говорущенко, В. С. Гончарук, К. О. Кобель // Вісник Хмельницького національного університету. Технічні науки. - 2018. - № 2. - С. 167-175. (дата звернення: 26.11.2023)
- ✓ 4. ^{оформити згідно з вимогами} Петрова К., Зелений О.П., Дейнеко Ж.В. Передумови врахування колірного простору при виборі технології для оброки зображень // Поліграфічні, мультимедійні та web-технології : тези доповідей VII Міжнар. наук.-техн. конф., 17-21 травня 2022 р. Харків: Мадрид, 2022. Т. 1. С. 99-100. (дата звернення: 26.11.2023)
- ✓ 5. URL: <https://99firms.com/blog/ab-testing-statistics/#gref> (дата звернення 26.11.2023)
6. Perkins, С.Е. Ad hoc On-demand Distance Vector (AODV) Routing / С.Е. Perkins, С.Е. Beldong-Royer // RFC 3561. – July 2003.(дата звернення: 01.12.2023)
7. Orobinskyi P., Deineko Z., Lyashenko V. Comparative Characteristics of Filtration Methods in the Processing of Medical Images // American Journal of Engineering Research. 2020. № 9(4). P. 20-25. .(дата звернення: 28.11.2023)
8. Петровський А.Б. Теорія прийняття рішень. М.: Видавничий центр «Академія», 2019. 400 с. .(дата звернення: 28.11.2023)
9. Lyashenko V.V., Matarneh R., Baranova V., Deineko Z.V. Hurst Exponent as a Part of Wavelet Decomposition Coefficients to Measure Long-term Memory Time Series Based on Multiresolution Analysis // American Journal of Systems and Software. 2016. № 4(2). P. 51-56. (дата звернення: 30.11.2023)

10. Weinschenk S. 100 things every designer needs to know about people. Pearson Education Limited, 2020. 256 p. .(дата звернення:30.11.2023)
11. Галузинський Г. П. Засади побудови користувацького інтерфейсу інформаційних управляючих систем / Г. П. Галузинський // Моделювання та інформаційні системи в економіці. - 2015. - № 91. - С. 116-123. (дата звернення: 28.11.2023)
12. Компанєєтс М. О. Принципи проектування ефективних веб-сайтів / М. О. Компанєєтс // Молодий вчений. - 2015. - № 9(2). - С. 106-109. (дата звернення: 27.11.2023)
13. Голобородько В. М. Формування дизайн-ергономічних вимог до користувацького інтерфейсу web-сайтів / В. М. Голобородько, М. Л. Опалєв // Вісник Харківської державної академії дизайну і мистецтв. - 2011. - № 3. - С. 16-19. .(дата звернення: 28.11.2023)
14. Гаврилов А.Г., Кубашева Е.С. Методики оцінювання якості сайтів. М., 2011. (дата звернення: 26.11.2023)
15. Боровинська Ю.Д., Колесникова Т.А. Рекомендації щодо розробки ui елементів для e-learning продуктів. // Поліграфічні, мультимедійні та web технології: матеріали Молодіжної школи-семінару V Міжнар. наук.-техн. конф. (18-22 травня 2021 р. м. Харків). 2021. Т2. С. 95-97. (дата звернення: 30.11.2023)
16. Даниленко С. П. Інструменти проектування сайту та їх особливості / С. П. Даниленко, Є. О. Ольховський // Комп'ютер у школі та сім'ї. - 2013. - № 3. - С. 48-51. .(дата звернення: 31.11.2023)
17. Древич А. І. Інформаційні технології розробки веб-сайтів / А. І. Древич, Р. В. Моравецький // Збірник наукових праць Інституту проблем моделювання в енергетиці ім. Г. Є. Пухова. - 2014. - Вип. 70. - С. 79-83. .(дата звернення: 31.11.2023)
18. Мандел Т. Разработка пользовательского интерфейса. М.: ДМК Пресс, 2001. 416 с. (дата звернення: 01.12.2023)

20. Котенко Н. О. Дослідження основних тенденцій сучасної розробки вебсайтів / Н. О. Котенко, С. В. Жирова, В. І. Чубаєвський, А. М. Десятко // Кібербезпека: освіта, наука, техніка. - 2019. - № 1. - С. 6-15. .(дата звернення: 01.12.2023)
21. Chakeres I. D., Klein-Berndt L. AODVjr, AODV simplified //ACM SIGMOBILE Mobile Computing and Communications Review. – 2002. – Т. 6. – №. 3. – С. 100-101. (дата звернення: 28.12.23)
22. Негрей М., Тужик К. Теорія прийняття рішень. К.: Центр навчальної літератури, 2018. 272 с. (дата звернення: 3.01.2024)

ДОДАТОК А

Технічно завдання

у вас додаток А та Б має одну назву. Такого не може бути

ЗАТВЕРДЖУЮ
Проректор Українського державного
університету науки і технологій
Анатолій РАДКЕВИЧ

«ДОСЛІДЖЕННЯ ЯКОСТІ ІНТЕРФЕЙСІВ У ВЕБ-САЙТАХ З
ВИКОРИСТАННЯМ СИСТЕМИ ДЛЯ А/В ТЕСТУВАННЯ»

Технічне завдання
44165850.1426-01

01426

одна лінія

✓ Завідувач кафедри КІТ
_____Вадим ГОРЯЧКІН
Керівник розробки
_____Іван КЛИМЕНКО
Виконавець
✓ _____Олексій БЕЗРЕБРИЙ
Нормоконтролер
✓ _____Світлана ВОЛКОВА

2024

ЗАТВЕРДЖЕНО
44165850.1426-01

01426

«ДОСЛІДЖЕННЯ ЯКОСТІ ІНТЕРФЕЙСІВ У ВЕБ-САЙТАХ З
ВИКОРИСТАННЯМ СИСТЕМИ ДЛЯ А/В ТЕСТУВАННЯ»

Технічне завдання *два додатка з одною назвою*

44165850.1426-01

Листів 15

2024

44165850.1426-01

ЗМІСТ

у додатках номер сторінки-по центру

Вступ.....	3
1 Підстава для розробки	4
2 Призначення розробки.....	5
2.1 Функціональне призначення розробки	5
2.2 Експлуатаційне призначення розробки:	6
3 Вимоги до програми	8
3.1 Вимоги до функціональних характеристик.....	8
3.2 Вимоги до надійності.....	10
3.3 Вимоги експлуатації	10
3.4 Вимоги до складу та параметрів технічних засобів	11
3.5 Вимоги до інформаційної та програмної сумісності.....	12
4 Вимоги до програмної документації.....	13
5 Стадії та етапи розробки.....	14
6 Порядок прийняття	15
7 Технічно-економічні показники	16

44165850.1426-01

ВСТУП

у додатках номер сторінки- по центру

У сучасному світі якість веб-інтерфейсу є запорукою ефективної взаємодії користувача з сайтом, а оптимізація користувацького інтерфейсу має безпосередній вплив на такі ключові бізнес-показники, як користувацький досвід, задоволеність користувачів, конверсії та утримання аудиторії. Одним з останніх підходів до покращення користувацьких інтерфейсів є використання систем А/В-тестування, які дозволяють об'єктивно оцінити ефективність різних варіантів інтерфейсу, порівнюючи їхній вплив на поведінку користувачів.

Це дослідження присвячене аналізу якості інтерфейсів веб-сайтів за допомогою систем А/В-тестування. Цей підхід розділяє трафік користувачів на кілька варіантів інтерфейсу (А і В), а також збирає та аналізує дані про взаємодію користувачів з цими варіантами. У веб-програмах, що реалізують таку систему, користувачі автоматично перенаправляються на різні версії сайту, їхня активність фіксується в базі даних, а коефіцієнти конверсії розраховуються за допомогою скриптів аналізу.

Дослідження має на меті розглянути ключові аспекти процесу А/В тестування, такі як збір та обробка даних, забезпечення надійності та конфіденційності інформації, а також вплив випадкових варіацій на точність отриманих результатів. Особливу увагу буде приділено аналізу ефективності різних версій інтерфейсу та їх впливу на користувацький досвід.

Основною метою дослідження є вироблення практичних рекомендацій для розробників і дизайнерів, які хочуть покращити якість інтерфейсів своїх веб-сайтів. У дослідженні визначено переваги та недоліки підходу А/В-тестування, розглянуто сценарії використання та надано рекомендації щодо ефективної оптимізації веб-інтерфейсу відповідно до останніх вимог та тенденцій..

44165850.1426-01
1 ПІДСТАВА ДЛЯ РОЗРОБКИ

Основою для розробки є наказ ректора Українського державного університету науки і технології Радкевич А.В. «Про затвердження тем та призначення керівників дипломних проєктів» №1196 ст від 05.12. 2022 року.

Тема проєкту: «Дослідження якості інтерфейсів у веб-сайтах з використанням системи для А/В тестування».

Керівник дипломного проєкту: Клименко І. В.

44165850.1426-01
2 ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка спрямована на дослідження якості інтерфейсів веб-сайтів за допомогою системи А/В-тестування.

Основна мета - проаналізувати ефективність різних варіантів інтерфейсу та оцінити їхній вплив на користувацький досвід та бізнес-показники, такі як конверсії, утримання та залучення користувачів. Мета цієї статті - висвітлити переваги та недоліки використання А/В-тестування в процесі розробки інтерфейсу. Зокрема, будуть розглянуті такі аспекти, як збір даних про дії користувачів, обробка інформації, забезпечення випадкових варіацій версій сайту та розрахунок показників ефективності.

Результати дослідження допоможуть розробникам приймати обґрунтовані рішення при виборі оптимальних варіантів дизайну інтерфейсу на основі отриманих даних. Крім того, дослідження має на меті визначити основні сценарії використання А/В-тестування та надати практичні рекомендації для успішного впровадження А/В-тестування в процес розробки сайту.

Результати цього дослідження будуть корисними для організацій та команд розробників, які хочуть покращити якість інтерфейсу та підвищити ефективність взаємодії користувачів з продуктом.

2.1 Функціональне призначення розробки

Основні аспекти функціонального призначення розробки:

- аналіз ефективності інтерфейсу - Важливою функцією розробки є детальний аналіз ефективності різних варіантів інтерфейсу сайту в контексті А/В тестування. Сюди входить збір даних про поведінку користувачів, розрахунок показників конверсії, рівня залученості та інших показників, що впливають на користувацький досвід;
- порівняння результатів А/В-тестування - розробка дозволяє порівняти ефективність різних варіантів інтерфейсу, виявити переваги та недоліки

44165850.1426-01

кожного з них і об'єктивно оцінити їхній вплив на такі ключові показники, як конверсія та утримання користувачів;

- вивчення практичних сценаріїв застосування - метою є аналіз сценаріїв використання А/В тестування в процесі розробки веб-інтерфейсів. Зокрема, будуть розглянуті ситуації, в яких тестування може допомогти підвищити ефективність дизайну, протестувати нові функції та оптимізувати підказки для користувачів;
- надання рекомендацій - результати розробки допомагають надати практичні рекомендації розробникам і дизайнерам щодо того, як проводити А/В-тестування для покращення інтерфейсу. Ці рекомендації базуються на отриманих даних та аналізі конкретних сценаріїв.

Висновки та документація - розробка завершується формулюванням висновків і підготовкою технічної документації, що включає результати дослідження, практичні рекомендації та інструкції щодо впровадження і використання системи А/В тестування в майбутніх проектах.

2.2 Експлуатаційне призначення розробки:

Основні аспекти експлуатаційного призначення розробки:

- оцінка ефективності інтерфейсу - результати опитування допомагають розробникам і організаціям об'єктивно оцінити, які варіанти веб-інтерфейсу найкраще відповідають вимогам користувачів і бізнес-цілям на основі таких показників А/В-тестування, як конверсії, залученість і час взаємодії;
- планування розвитку проекту - інформація з цього дослідження допоможе розробникам планувати та реалізовувати нові проекти з акцентом на оптимізацію інтерфейсу для покращення користувацького досвіду;
- оптимізація інтерфейсу - аналіз результатів А/В-тестування для виявлення слабких місць в існуючих інтерфейсах, покращення дизайну та

44165850.1426-01

функціональності, а також поліпшення ключових показників ефективності;

- підтримка та розвиток існуючих проектів - підтримка та вдосконалення існуючих веб-програм з використанням системного підходу, включаючи тестування нових функцій, аналіз результатів та впровадження рішень, що найкраще відповідають мінливим потребам користувачів;
- підготовка документації - оперативні цілі розробки включають підготовку технічної документації, включаючи кращі практики, рекомендації та результати досліджень для подальшого використання в проекті.

Операційною метою отриманої розробки є застосування на практиці отриманих результатів для розробки, вдосконалення та підтримки веб-проектів, орієнтованих на покращення якості інтерфейсу та покращення користувацького досвіду.

44165850_1426-01
3 ВИМОГИ ДО ПРОГРАМИ

3.1 Вимоги до функціональних характеристик

Онсовні вимоги до функціональних характеристик:

- зчитування та запис даних – дослідження повинно охоплювати здатність системи фіксувати та записувати активності користувачів із фронтенду веб-програми, включаючи операції створення, оновлення та видалення записів у базі даних;
- розподіл трафіку та рандомізація – необхідно оцінити здатність системи рівномірно розподіляти користувачів між варіантами А та В із забезпеченням рандомної варіативності інтерфейсів;
- обробка даних – дослідження має охоплювати можливість системи аналізувати події користувачів, проводити агрегацію даних і фільтрацію для обчислення ключових метрик, таких як конверсія та залученість.
- масштабованість – система повинна забезпечувати стабільну роботу за умов збільшення обсягу даних і кількості користувачів, підтримуючи високий рівень продуктивності та швидкодії;
- безпека даних – дослідження має враховувати наявність механізмів захисту даних користувачів, включаючи автентифікацію, авторизацію та шифрування;
- резервне копіювання та відновлення – система повинна забезпечувати можливість створення резервних копій даних та їх відновлення у випадку пошкодження чи втрати.
- обчислення метрик – дослідження повинно оцінити ефективність Python-скриптів для обчислення ключових метрик А/В тестування на основі даних, збережених у базі;
- інтеграція з іншими технологіями – важливо дослідити можливості інтеграції системи з веб-серверами, мовами програмування та іншими сервісами для забезпечення гнучкості впровадження;

44165850.1426-01

- адміністрування та моніторинг – система повинна підтримувати інструменти для моніторингу активності користувачів, обробки даних та виявлення потенційних проблем у роботі.
- аналіз результатів – вимоги повинні охоплювати можливість генерації звітів із результатами тестування для аналізу ефективності інтерфейсів та подальшої оптимізації;

Ці вимоги допомагають забезпечити ретельну перевірку якості інтерфейсу сайту і полегшують впровадження ефективних рішень на основі А/В-тестування.

Вхідні дані:

- дані структури подій – опис структур даних, що зберігаються в системі, включаючи тип події користувача, ім'я атрибута, формат і зв'язок з варіантом інтерфейсу (А або В);
- обсяг даних – кількість і обсяг подій користувачів, які планується зберігати в базі даних, включаючи текстову інформацію про дії користувачів і метрики тестування;
- методи доступу до даних – опис способів запису, зчитування, оновлення та видалення даних із бази для забезпечення збору й аналізу результатів А/В тестування;
- сценарії використання – визначення сценаріїв тестування, таких як аналіз нових варіантів дизайну, перевірка реакції користувачів на функціональні зміни чи покращення елементів інтерфейсу;
- серверні налаштування – інформація про конфігурацію серверів для забезпечення зберігання та обробки даних, включаючи обсяг пам'яті, продуктивність процесора та пропускну здатність системи.

Вихідні дані:

- результати дослідження ефективності інтерфейсів – звіт, який містить аналіз ефективності різних варіантів інтерфейсів (А і В), метрики конверсії, утримання користувачів та інших показників;

44165850.1426-01

- рекомендації щодо оптимізації інтерфейсів – документ із практичними порадами щодо впровадження найкращого варіанту інтерфейсу на основі результатів тестування, із зазначенням переваг і недоліків кожного варіанту;
- план впровадження А/В тестування – покрокова інструкція щодо інтеграції системи тестування у процес розробки веб-програм, включаючи методи автоматизації, аналітики та обробки даних.

3.2 Вимоги до надійності

Вимоги до надійності наступні:

- інформування про стан роботи системи – під час оновлення даних система повинна забезпечувати виведення повідомлень, які інформують користувачів про поточний стан операцій та результати їх виконання (наприклад, успішне завершення, помилка чи затримка);
- архівна копія програми – текст програми повинен зберігатися у вигляді архівної копії на зовнішньому носії (наприклад, USB-накопичувач чи хмарний сервіс) для забезпечення можливості відновлення у разі втрати основного джерела;
- резервна копія бази даних – база даних, яка використовується для зберігання даних про активності користувачів і результати тестування, повинна мати резервну копію на зовнішньому носії для запобігання втраті інформації через технічні збої чи помилки.

3.3 Вимоги експлуатації

Працювати з програмою може людина, що має навички роботи з десктопними пристроями та ознайоmlена з керівництвом користувача програмного продукту.

44165850.1426-01

3.4 Вимоги до складу та параметрів технічних засобів

Програма, що розробляється, повинен використовуватись на пристроях, які відповідають наступним технічним характеристикам:

- операційна система – пристрій повинен працювати під управлінням операційної системи, яка підтримує розроблений продукт. Рекомендовані ОС: Windows 10 або новіша версія, macOS 10.14 або новіша, або Linux із сучасним ядром (версія 5.0 і вище);
- діагональ монітора – для зручного відображення інтерфейсу та взаємодії з програмою монітор повинен мати діагональ не менше 21 дюйма. Оптимальний діапазон: 21–24 дюйми;
- роздільна здатність монітора – монітор повинен підтримувати роздільну здатність не менше Full HD (1920x1080), що забезпечить чітке та якісне відображення елементів інтерфейсу;
- оперативна пам'ять (RAM) – пристрій повинен мати не менше 8 ГБ оперативної пам'яті для ефективною роботи системи та обробки даних;
- вбудована пам'ять (жорсткий диск або SSD) – для зберігання даних продукту потрібна ємність не менше 256 ГБ. Рекомендовано використання SSD для підвищення швидкості роботи системи;
- частота процесора – для швидкої обробки даних пристрій повинен мати процесор із частотою не менше 2.5 ГГц. Рекомендується багатоядерний процесор із підтримкою сучасних інструкцій (наприклад, Intel Core i5 або AMD Ryzen 5);
- порти та комунікації – пристрій повинен мати USB-порти для підключення периферійних пристроїв, Ethernet-порт для мережевого підключення, а також підтримувати Wi-Fi для бездротового доступу до інтернету.

44165850.1426-01

3.5 Вимоги до інформаційної та програмної сумісності

Програма підтримує роботу на операційних системах Windows (версія 10 та новіші), macOS (версія 10.14 та новіші) і Linux (сучасні дистрибутиви з ядром 5.0 або новішим). Забезпечується коректна робота в зазначених середовищах із збереженням однакової функціональності та продуктивності.

44165850_1426-01

4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

До складу документації мають входити:

- текст програми;
- керівництво користувача для користування веб програмою.

Вся документація програми повинна задовольняти вимоги до програмної документації.

44165850.1426-01
5 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

Таблиця 5.1 – Стадії та етапи розробки

Стадія	Зміст	Строки виконання
Технічне завдання	Постановка задачі, збір інформації, вибір та обґрунтування критеріїв розробки. Попередній вибір методів рішення задач. Визначення вимог до технічних засобів. Узгодження і затвердження технічного завдання.	04.09.24 – 15.09.24
Робочий проект	Програмування та відлагодження програми.	18.09.24 – 22.09.24
	Тестування програми	25.09.24 – 27.09.24
	Розробка, узгодження і затвердження програмної документації.	28.09.24 – 29.09.24

виправити шріфт



44165850.1426-01
6 ПОРЯДОК ПРИЙНЯТТЯ

Контроль за виконанням роботи здійснює керівник розробки доц. Клименко
І. В.

44165850.1426-01

7 ТЕХНІЧНО-ЕКОНОМІЧНІ ПОКАЗНИКИ

Показники та їх розрахунок не були описані у документах бо розробка програми несе в собі навчальний характер, а не комерційний.

ДОДАТОК Б

Технічне завдання

ЗАТВЕРДЖУЮ

Проректор Українського державного
університету науки і технологій
Анатолій РАДКЕВИЧ

«ДОСЛІДЖЕННЯ ЯКОСТІ ІНТЕРФЕЙСІВ У ВЕБ-САЙТАХ З
ВИКОРИСТАННЯМ СИСТЕМИ ДЛЯ А/В ТЕСТУВАННЯ»

Текст програми

44165850.01426–01 12–01

Завідувач кафедри КІТ

_____Вадим ГОРЯЧКІН

Керівник розробки

_____Іван КЛИМЕНКО

✓ Виконавець

✓ _____Олексій БЕЗРЕБРИЙ

Нормоконтролер

✓ _____Світлана ВОЛКОВА

2024

ЗАТВЕРДЖЕНО

44165850.01426-01 12-01

**ДОСЛІДЖЕННЯ ЯКОСТІ ІНТЕРФЕЙСІВ У ВЕБ-САЙТАХ З
ВИКОРИСТАННЯМ СИСТЕМИ ДЛЯ А/В ТЕСТУВАННЯ**

Текст програми
44165850.01426-01 12-01
Листів 10

44165850.01426–01 12

АНОТАЦІЯ

Документ 44165850.01426–01 12 01 «Дослідження якості інтерфейсів у веб-сайтах з використанням системи для А/В тестування» входить до складу програмної документації на програму, що виконує програму для аналізування конверсії між різними видами дизайнів інтерфейсу.

У даному документі представлений текст програми. Програма написана на мові Python з використанням мови програмування TypeScript для реалізації фронтенду та бекенду на фреймворках Nextjs та Nestjs у програмному середовищі Visual Studio Code.

44165850.01426–01 12

ЗМІСТ

1 Текст програми	4
1.1 Текст основної програми TestWithDropBox.py	4
1.2 Текст моделі бази даних для активності activity.model.ts	5
1.3 Текст контролера для активності activity.controller.ts	6
1.4 Текст сервіса для активності activity.service.ts	6
1.5 Текст моделі бази даних для часової активності activity-timer.model.ts	7
1.6 Текст контролера для часової активності activity-timer.controller.ts	7
1.7 Текст сервіса для часової активності activity-timer.service.ts	8
1.8 Текст фронтенд частини index.tsx	8
1.9 Текст фронтенд частини запитів activity-timer.ts	10
1.10 Текст фронтенд частини запитів activity.ts	11

44165850.01426-01 12

1 ТЕКСТ ПРОГРАМИ

1.1 Текст основної програми TestWithDropBox.py

```

import tkinter as tk
from tkinter import ttk

from scipy.stats import norm
import math
import mysql.connector

def calculate_conversion_and_confidence_interval(conversions,
sample_size, confidence_level=0.95):
    conversion_rate = (conversions / sample_size) * 100
    z_value = norm.ppf((1 + confidence_level) / 2)
    standard_error = math.sqrt((conversion_rate * (100 -
conversion_rate)) / sample_size)
    lower_bound = conversion_rate - z_value * standard_error
    upper_bound = conversion_rate + z_value * standard_error
    confidence_interval = (lower_bound, upper_bound)
    return conversion_rate, confidence_interval

class ConversionCalculator(tk.Tk):
    def __init__(self):
        super().__init__()

        self.title("Conversion Calculator")

        self.metric_mapping = {
            "Show Filters": "filters__action__show",
            "Price Range": "filters_price_range",
            "Clothes Categories": "filters_clothes_categories",
            "Clothes Manufacturer": "filters_clothes_manufacturer",
            "Clothes Gender Category":
"filters_clothes_gender_category",
            "Children Clothes Categories":
"filters_clothes_categories_for_children",
            "Dashboard Slide": "dashboard__slide",
            "Dashboard Alert Button Close":
"dashboard__alert__btn_close",
            "Instagram Image": "footer_social_img_instagram",
            "Cart Button": "cart__btn",
            "Order Button in Cart": "cart_popup__footer__btn_order",
            "Delete Button in Cart": "delete_cart_popup_btn",
            "Remove Filter Button(New)":
"manufacturers__list__item__btn_remove_filter",
            "Filter Reset Button(New)": "catalog__top__reset_btn",
            "Filter Reset Button(Old)": "filters__actions__main__reset"
        }

        # Выпадающий список для выбора метрика конверсии
        self.label_metric = tk.Label(self, text="Select metric:")
        self.combobox_metric = ttk.Combobox(self, width=30)
        self.combobox_metric['values'] =
list(self.metric_mapping.keys())
        self.combobox_metric.set("Select metric")

        # Вариант A
        self.variant_a_frame = tk.LabelFrame(self, text="Variant A")
        self.variant_a_frame.grid(row=2, column=0, padx=10, pady=10,
sticky="nsew")

        self.label_a_conversions = tk.Label(self.variant_a_frame,
text="Conversions:")
        self.result_label_a = tk.Label(self.variant_a_frame, text="",
font=("Helvetica", 12, "bold"))

        # Вариант B
        self.variant_b_frame = tk.LabelFrame(self, text="Variant B")

        self.variant_b_frame.grid(row=2, column=1, padx=10,
pady=10, sticky="nsew")

        # Отображение
        self.label_b_conversions = tk.Label(self.variant_b_frame,
text="Conversions:")
        self.result_label_b = tk.Label(self.variant_b_frame, text="",
font=("Helvetica", 12, "bold"))

        # Результаты
        self.result_label = tk.Label(self, text="Comparison Result:")

        # Кнопка расчета
        self.calculate_button = tk.Button(self, text="Calculate",
command=self.calculate_and_compare)

        # Геометрия окна
        self.columnconfigure(0, weight=1)
        self.columnconfigure(1, weight=1)

        # Упаковка элементов
        self.label_a_conversions.grid(row=0, column=0, padx=5,
pady=5, sticky="e")
        self.result_label_a.grid(row=0, column=1, padx=5, pady=5,
sticky="w")

        self.label_b_conversions.grid(row=0, column=0, padx=5,
pady=5, sticky="e")
        self.result_label_b.grid(row=0, column=1, padx=5, pady=5,
sticky="w")

        self.label_metric.grid(row=0, column=0, colspan=6,
pady=5)
        self.combobox_metric.grid(row=1, column=0, colspan=6,
pady=5)

        self.result_label.grid(row=3, column=0, colspan=2,
pady=10)

        self.calculate_button.grid(row=4, column=0, colspan=2,
pady=10)

        # Подключение к базе данных
        self.db_connection = mysql.connector.connect(
            host="localhost",
            user="root",
            password="some_password",
            database="test_online_store"
        )

    def calculate_and_compare(self):
        try:
            # Вариант A
            conversions_a =
self.get_conversion_v1(self.get_selected_metric())
            sample_size = self.get_sample_size()
            conversion_rate_a, ci_a =
calculate_conversion_and_confidence_interval(conversions_a,
sample_size)
            result_text_a = f"Conversion A:
{conversion_rate_a:.2f}%\nCI: ({ci_a[0]:.2f}%, {ci_a[1]:.2f}%)
"
            self.result_label_a.config(text=result_text_a)

            # Вариант B
            conversions_b =
self.get_conversion_v2(self.get_selected_metric())
            sample_size = self.get_sample_size()

```

44165850.01426-01 12

```

        conversion_rate_b, ci_b =
calculate_conversion_and_confidence_interval(conversions_b,
sample_size)
        result_text_b = f"Conversion B:
{conversion_rate_b:.2f}%\nCI: ({ci_b[0]:.2f}%, {ci_b[1]:.2f}%)"
        self.result_label_b.config(text=result_text_b)

        # Сравнение
        result_text = self.compare_variants(conversion_rate_a,
sample_size, conversion_rate_b)
        self.result_label.config(text=result_text)

    except ValueError as e:
        self.result_label.config(text=f"Error: {str(e)}")

def get_selected_metric(self):
    selected_metric_key = self.combobox_metric.get()
    return self.metric_mapping.get(selected_metric_key, None)

def get_sample_size(self):
    # Получение количества уникальных значений из столбца
    "id" таблицы "users"
    cursor = self.db_connection.cursor()
    query = "SELECT COUNT(DISTINCT id) FROM users"
    cursor.execute(query)
    result = cursor.fetchone()

    if result:
        return result[0]
    else:
        raise ValueError("Error fetching sample size from the
database.")

def get_conversion(self, version, metric):
    # Получение количества уникальных значений из столбца
    "id" таблицы "activities"
    cursor = self.db_connection.cursor()
    query = "SELECT COUNT(DISTINCT id) FROM activities
WHERE VERSION = %s AND ITEM_ID = %s"
    cursor.execute(query, (version, metric))
    result = cursor.fetchone()

    if result:
        return result[0]
    else:
        raise ValueError("Error fetching conversion from the
database")

def get_conversion_v1(self, metric):
    return self.get_conversion('v1', metric)

```

```

def get_conversion_v2(self, metric):
    return self.get_conversion('v2', metric)

def compare_variants(self, conversions_a, sample_size,
conversions_b):
    # Расчет конверсии и доверительных интервалов
    conversion_rate_a, ci_a =
calculate_conversion_and_confidence_interval(conversions_a,
sample_size)
    conversion_rate_b, ci_b =
calculate_conversion_and_confidence_interval(conversions_b,
sample_size)

    # Преобразование конверсии в проценты
    conversion_rate_a_percentage = f"{conversion_rate_a:.2f}%"
    conversion_rate_b_percentage = f"{conversion_rate_b:.2f}%"

    # Сравнение конверсий и доверительных интервалов
    if ci_a[0] <= ci_b[1] and ci_b[0] <= ci_a[1]:
        # Доверительные интервалы пересекаются
        if conversion_rate_a > conversion_rate_b:
            result_text = f"Variants A and B do not significantly differ.
Variant A is {conversion_rate_a_percentage} higher."
        elif conversion_rate_b > conversion_rate_a:
            result_text = f"Variants A and B do not significantly differ.
Variant B is {conversion_rate_b_percentage} higher."
        else:
            result_text = "Variants A and B have the same conversion
rate."
    else:
        # Доверительные интервалы не пересекаются
        if conversion_rate_a > conversion_rate_b:
            result_text = f"Variants A is significantly better than
Variant B by {conversion_rate_a_percentage}."
        elif conversion_rate_b > conversion_rate_a:
            result_text = f"Variants B is significantly better than
Variant A by {conversion_rate_b_percentage}."
        else:
            result_text = "Variants A and B have the same conversion
rate."

    self.result_label.config(text=result_text)

    return result_text

if __name__ == "__main__":
    app = ConversionCalculator()
    app.mainloop()

```

1.2 Текст моделі бази даних для активності activity.model.ts

```

import { Column, Model, Table } from 'sequelize-typescript';

@Table

export class Activity extends Model {

    @Column
    username: string;

    @Column
    email: string;

```

```

@Column
item_id: string;

@Column
mode: string;

@Column
version: string;

```

1.3 Текст контролера для активності activity.controller.ts

```

import {
  Body,
  Controller,
  Header,
  HttpStatusCode,
  HttpStatus,
  Post,
  UseGuards,
} from '@nestjs/common';
import { AuthenticatedGuard } from 'src/auth/authenticated.guard';
import { ActivityService } from './activity.service';
import { CreateActivityDto } from './dto/create-activity.dto';

@Controller('activity')
export class ActivityController {
  constructor(private readonly activityService: ActivityService) {}

  @Post('/click')
  @HttpCode(HttpStatus.CREATED)
  @Header('Content-type', 'application/json')
  createActivity(@Body() createActivityDto: CreateActivityDto) {
    return this.activityService.create(createActivityDto);
  }
}

@UseGuards(AuthenticatedGuard)
@Post()
getActivity(
  @Body()
  params: {
    username?: string;

    email?: string;

    item_id?: string;

    mode?: string;

    version?: string;
  },
) {
  return this.activityService.findAll(params);
}

```

1.4 Текст сервіса для активності activity.service.ts

```

import { Injectable } from '@nestjs/common';
import { InjectModel } from '@nestjs/sequelize';
import { Activity } from './activity.model';
import { CreateActivityDto } from './dto/create-activity.dto';

@Injectable()
export class ActivityService {
  constructor(
    @InjectModel(Activity)
    private activityModel: typeof Activity,
  ) {}

  findAll(filter: {
    id?: string;
    username?: string;
    email?: string;
    mode?: string;
    version?: string;
    item_id?: string;
  }): Promise<Activity[]> {
    return this.activityModel.findAll({ where: { ...filter } });
  }
}

```

44165850.01426-01 12

```

async create(
  createActivityDto: CreateActivityDto,
): Promise<Activity | { warningMessage: string }> {
  const activity = new Activity();

  activity.username = createActivityDto.username;
  activity.email = createActivityDto.email;
  activity.item_id = createActivityDto.item_id;

  activity.mode = createActivityDto.mode;
  activity.version = createActivityDto.version;

  return activity.save();
}

```

1.5 Текст моделі бази даних для часової активності activity-timer.model.ts

```

import { Column, Model, Table } from 'sequelize-typescript';

@Table
export class ActivityTimer extends Model {
  @Column
  username: string;

  @Column
  page: string;

  @Column
  time: number;

  @Column
  version: string;
}

```

1.6 Текст контролера для часової активності activity-timer.controller.ts

```

import {
  Body,
  Controller,
  Header,
  HttpStatusCode,
  HttpStatus,
  Post,
  UseGuards,
} from '@nestjs/common';
import { AuthenticatedGuard } from 'src/auth/authenticated.guard';
import { ActivityTimerService } from './activity-timer.service';
import { CreateActivityTimerDto } from './dto/create-activity-timer.dto';

@Controller('activity-timer')
export class ActivityTimerController {
  constructor(
    private readonly activityTimerService: ActivityTimerService
  ) {}

  @Post('/timer')
  @HttpCode(HttpStatus.CREATED)
  @Header('Content-type', 'application/json')
  createActivity(
    @Body() createActivityTimerDto: CreateActivityTimerDto
  ) {
    return this.activityTimerService.create(createActivityTimerDto);
  }

  @UseGuards(AuthenticatedGuard)
  @Post()
  getActivity(
    @Body()
    params: {
      username?: string;
    }
  ) {}
}

```

44165850.01426-01 12

```

page?: string;

time?: number;

version?: string;
},
    ) {
      return this.activityTimerService.findAll(params);
    }
  }
}

```

1.7 Текст сервіса для часової активності activity-timer.service.ts

```

import { Injectable } from '@nestjs/common';
import { ActivityTimer } from './activity-timer.model';
import { InjectModel } from '@nestjs/sequelize';
import { CreateActivityTimerDto } from './dto/create-activity-timer.dto';

@Injectable()
export class ActivityTimerService {
  constructor(
    @InjectModel(ActivityTimer)
    private activityTimerModel: typeof ActivityTimer,
  ) {}

  findAll(filter: {
    id?: string;
    username?: string;
    page?: string;
    time?: number;
    version?: string;
  }): Promise<ActivityTimer[]> {
    return this.activityTimerModel.findAll({ where: { ...filter } });
  }

  async create(
    createActivityDto: CreateActivityTimerDto,
  ): Promise<ActivityTimer | { warningMessage: string }> {
    const activityTimer = new ActivityTimer();

    activityTimer.username = createActivityDto.username;
    activityTimer.page = createActivityDto.page;
    activityTimer.time = createActivityDto.time;
    activityTimer.version = createActivityDto.version;

    return activityTimer.save();
  }
}

```

1.8 Текст фронтенд частини index.tsx

```

import Layout from '@components/layout/Layout'
import Breadcrumbs from '@components/modules/Breadcrumbs/Breadcrumbs'
import DashboardPage from '@components/templates/DashboardPage/DashboardPage'
import { useActivityTimer } from '@hooks/useActivityTimer'
import useRedirectByUserCheck from '@hooks/useRedirectByUserCheck'
import Head from 'next/head'

import { useEffect } from 'react'

function Dashboard() {
  const { shouldLoadContent } = useRedirectByUserCheck()
  const getDefaultTextGenerator = () => ''
  const getTextGenerator = () => ''
}

```

44165850.01426-01 12

```

const sendTimerData = useActivityTimer()

useEffect(() => () => {
  sendTimerData('DashboardPage')
})

return (
  <>
  <Head>
    <title>Candy Shop Style | {shouldLoadContent ? 'Главная' : ''}</title>
    <meta charset="UTF-8" />
    <meta httpEquiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="icon" type="image/svg" sizes="32x32" href="/img/logo.svg" />
  </Head>
  {shouldLoadContent && (
    <Layout>
      <main>
        <Breadcrumbs
          getDefaultTextGenerator={getDefaultTextGenerator}
          getTextGenerator={getTextGenerator}
        />
        <DashboardPage />
        <div className="overlay" />
      </main>
    </Layout>
  )}
</>
)
}

export default Dashboard

import AuthPage from '@components/templates/AuthPage/AuthPage'
import useRedirectByUserCheck from '@hooks/useRedirectByUserCheck'
import { useVersion } from '@hooks/useVersion'
import Head from 'next/head'
import { useEffect } from 'react'

function Auth() {
  const { shouldLoadContent } = useRedirectByUserCheck(true)

  const { toggleVersion } = useVersion()

```

```

useEffect(() => {
  toggleVersion()
}, [])

return (
  <>
  <Head>
    <title>
      Candy Shop Style | {shouldLoadContent ? 'Авторизация' : ''}
    </title>
    <meta charset="UTF-8" />
    <meta httpEquiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="icon" type="image/svg" sizes="32x32" href="/img/logo.svg" />
  </Head>
  {shouldLoadContent && <AuthPage />}
</>
)
}

export default Auth

import Layout from '@components/layout/Layout'
import Breadcrumbs from '@components/modules/Breadcrumbs/Breadcrumbs'
import CatalogPage from '@components/templates/CatalogPage/CatalogPage'
import { useActivityTimer } from '@hooks/useActivityTimer'
import useRedirectByUserCheck from '@hooks/useRedirectByUserCheck'
import { IQueryParams } from '@types/catalog'
import Head from 'next/head'
import { useCallback, useEffect } from 'react'

function Catalog({ query }: { query: IQueryParams }) {
  const { shouldLoadContent } = useRedirectByUserCheck()
  const getDefaultTextGenerator = useCallback(() => 'Каталог', [])
  const getTextGenerator = useCallback((param: string) => ({})[param], [])

  const sendTimerData = useActivityTimer()

  useEffect(() => () => {
    sendTimerData('CatalogPage')
  })

```

```

return (
  <>
  <Head>
    <title>Candy Shop Style | {shouldLoadContent ? 'Каралор' :
"</title>
    <meta charSet="UTF-8" />
    <meta httpEquiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <link rel="icon" type="image/svg" sizes="32x32"
href="/img/logo.svg" />
  </Head>
  {shouldLoadContent && (
  <Layout>
    <main>
      <Breadcrumbs
        getDefaultTextGenerator={getDefaultTextGenerator}
        getTextGenerator={getTextGenerator}
      />
      <CatalogPage query={query} />
    </main>
  </Layout>
  </div className="overlay" />
</main>
</Layout>
)}
</>
)
}

export async function getServerSideProps(context: { query:
IQueryParams }) {
  return {
    props: { query: { ...context.query } },
  }
}

export default Catalog

```

1.9 Текст фронтенд частини запитів activity-timer.ts

```

import { HTTPStatus } from '@/constans'
import { IActivityTimerFx } from '@/types/activity'
import { AxiosError } from 'axios'
import { createEffect } from 'effector-next'
import { toast } from 'react-toastify'
import api from '../axiosClient'

export const sendUserActivityTimerFx = createEffect(
  async ({ url, username, page, time, version }: IActivityTimerFx) =>
  {
    const { data } = await api.post(url, {
      username,
      page,
      time,
      version,
    })

    if (data.warningMessage) {
      toast.warning(data.warningMessage)
      return
    }

    return data
  }
)

export const getActivityTimerFx = createEffect(
  async ({ url, username, page, time, version }: IActivityTimerFx) =>
  {
    try {
      const { data } = await api.post(url, {
        username,
        page,
        time,
        version,
      })

      return data
    } catch (error) {
      const axiosError = error as AxiosError
    }
  }
)

```

44165850.01426-01 12

```

    }
    }
    }
    )
  }
}

toast.error((error as Error).message)

```

1.10 Текст фронтенд частини запитів activity.ts

```

import { HTTPStatus } from '@/constans'
import { AxiosError } from 'axios'
import { createEffect } from 'effector-next'
import { toast } from 'react-toastify'
import api from './axiosClient'
import { IActivityFx } from '@/types/activity'

export const sendUserActivityFx = createEffect(
  async ({ url, username, email, item_id, mode, version }: IActivityFx)
=> {
  const { data } = await api.post(url, {
    username,
    email,
    item_id,
    mode,
    version,
  })

  if (data.warningMessage) {
    toast.warning(data.warningMessage)
    return
  }

  return data
}
)

export const getActivityFx = createEffect(
  async ({ url, username, email, item_id, mode, version }: IActivityFx)
=> {
  try {
    const { data } = await api.post(url, {
      username,
      email,
      item_id,
      mode,
      version,
    })

    return data
  } catch (error) {
    const axiosError = error as AxiosError

    if (axiosError.response) {
      if (axiosError.response.status === HTTPStatus.FORBIDDEN) {
        return false
      }
    }

    toast.error((error as Error).message)
  }
}
)

```

ДОДАТОК В

не вірна назва додатку

Технічно завдання

ЗАТВЕРДЖУЮ

Проректор Українського державного
університету науки і технологій
Анатолій РАДКЕВИЧ

«ДОСЛІДЖЕННЯ ЯКОСТІ ІНТЕРФЕЙСІВ У ВЕБ-САЙТАХ З
ВИКОРИСТАННЯМ СИСТЕМИ ДЛЯ А/В ТЕСТУВАННЯ»

Керівництво користувача
44165850.01426 – 01 ІЗ 01

Завідувач кафедри КІТ

✓ _____Вадим ГОРЯЧКІН

Керівник розробки

✓ _____Іван КЛИМЕНКО

Виконавець

✓ _____Олексій БЕЗРЕБРИЙ

Нормоконтролер

✓ _____Світлана ВОЛКОВА

2024

ЗАТВЕРДЖЕНО
44165850.01426-01 ІЗ 01

«ДОСЛІДЖЕННЯ ЯКОСТІ ІНТЕРФЕЙСІВ У ВЕБ-САЙТАХ З
ВИКОРИСТАННЯМ СИСТЕМИ ДЛЯ А/В ТЕСТУВАННЯ»

Керівництво користувача
44165850.01351 – 01 ІЗ 01

Листів 9

АНОТАЦІЯ

Документ 1116130.01426 – 01 ІЗ 01 «Дослідження якості інтерфейсів у веб-сайтах з використанням системи для А/В тестування».

Програми написані на мові Python з використанням мови програмування TypeScript для реалізації фронтенду та бекенду на фреймворках Nextjs та Nestjs у програмному середовищі Visual Studio Code.

44165850.01426-01 ІЗ 01
ЗМІСТ

1 Введення.....	4
2 Призначення та умови застосування.....	5
3 Підготовка до роботи.....	6
4 Опис операцій.....	7
5 Аварійні ситуації.....	9
6 Рекомендації щодо застосування.....	10

44165850.01426-01 ІЗ 01
1 ВВЕДЕННЯ

Програма “Conversion” призначений для аналізу та оптимізації якості веб-інтерфейсів за допомогою методики А/В тестування.

Головною метою розробки є надання інструментів для збору та аналізу даних про поведінку користувачів на різних варіантах інтерфейсу, обчислення ключових показників, таких як конверсія, та підтримка прийняття рішень щодо оптимізації інтерфейсу.

Програма призначений для використання як науковцями, що досліджують взаємодію користувачів із веб-програмами, так і розробниками та дизайнерами, які хочуть покращити якість інтерфейсів. Він не вимагає високого рівня досвіду для роботи та може використовуватись будь-ким із базовими знаннями роботи з комп'ютером.

Програма не потребує ознайомлення з додатковою експлуатаційною документацією, оскільки його інтерфейс інтуїтивно зрозумілий і простий у використанні.

44165850.01426-01 ІЗ 01
2 ПРИЗНАЧЕННЯ ТА УМОВИ ЗАСТОСУВАННЯ

Функціональним призначенням програми є проведення аналізу якості інтерфейсів веб-сайтів шляхом використання системи А/В тестування для визначення оптимального варіанту інтерфейсу в конкретних умовах.

Експлуатаційне призначення програми “Conversion” полягає у зборі даних про активності користувачів, обчисленні метрик ефективності (наприклад, конверсії) та створенні рекомендацій щодо покращення веб-інтерфейсів.

Для сталого функціонування програми необхідно дотримуватися таких умов: наявність стабільного інтернет-з'єднання.

Програма розрахована на використання на пристроях із такими характеристиками:

- процесор із тактовою частотою не менше 2.5 ГГц;
- оперативна пам'ять не менше 8 ГБ.

Для роботи програми потрібна операційна система Windows 10 або новіша, macOS версії 10.14 і вище, або сучасний дистрибутив Linux із ядром 5.0 і новішим.

44165850.01426-01 ІЗ 01
3 ПІДГОТОВКА ДО РОБОТИ

Для початку роботи програми необхідно додати до бази даних таблиці з активностями юзера та часом, що він провів. Також потрібно передати в головний Python скрипт дані для під'єднання до бази даних.

Після встановлення та запуску програми “Conversion” на формі програми присутні:

- поле вибору метрики конверсії: дозволяє вибрати метрику для аналізу з випадального списку;
- секція для варіанту А: містить інформацію про кількість конверсій та довірчий інтервал для цього варіанту;
- секція для варіанту В: аналогічна до секції А, але для другого варіанту інтерфейсу;
- кнопка "Calculate": виконує обчислення конверсій та довірчих інтервалів для обох варіантів, а також порівнює їх;
- результуючий текст: показує висновок порівняння варіантів А і В.

Основні операції програми:

- поле вибору метрики конверсії: дозволяє вибрати метрику для аналізу з випадального списку;
- вибір метрики: користувач обирає метрику із запропонованого списку (наприклад, кнопка замовлення чи фільтр каталогу);
- отримання даних із бази: програма звертається до бази даних для визначення кількості конверсій кожного варіанту інтерфейсу та розміру вибірки;
- обчислення показників: розрахунок коефіцієнта конверсії для обох варіантів та обчислення довірчого інтервалу на основі заданого рівня довіри (наприклад, 95%);
- порівняння варіантів: якщо довірчі інтервали перекриваються, програма робить висновок, що варіанти статистично не відрізняються або, якщо інтервали не перекриваються, програма визначає, який варіант значно кращий;

44165850.01426-01 ІЗ 01

- виведення результатів: усі результати відображаються у вигляді тексту для кожного варіанту та загального висновку.

44165850.01426-01 ІЗ 01
5 АВАРІЙНІ СИТУАЦІЇ

У разі запуску програми на пристрої без інтернет-з'єднання програма виведе повідомлення про помилку: “Помилка: відсутнє інтернет-з'єднання” та автоматично завершить роботу. Для продовження використання програми необхідно забезпечити стабільне підключення до інтернету та повторно запустити програму.

Якщо під час роботи програма поводить себе некоректно або виникає помилка, необхідно закрити її та перезапустити. Це дозволить відновити нормальне функціонування. У разі повторення проблеми рекомендується перевірити налаштування системи або звернутися до технічної підтримки.

44165850.01426-01 ІЗ 01
6 РЕКОМЕНДАЦІЇ ЩОДО ЗАСТОСУВАННЯ

Після встановлення програми на пристрій його потрібно запустити через виконуваний файл. Під час запуску відкриється головний екран із усім необхідним функціоналом для роботи.

Користувач може ознайомитися з функціями програми, виконати необхідні операції та завершити його роботу, скориставшись апаратною кнопкою вимкнення або стандартним інструментом для закриття програм у операційній системі.