

Міністерство освіти і науки України
Український державний університет науки і технологій

Комп'ютерні технології і системи

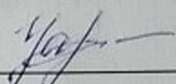
Комп'ютерні інформаційні технології

Пояснювальна записка
до кваліфікаційної роботи
Магістр

на тему: Дослідження та впровадження принципів UX/UI для підвищення ефективності взаємодії користувачів з туристичним вебсайтом

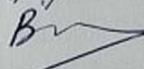
за освітньою програмою 12 Інженерія програмного забезпечення
зі спеціальності: 121 Інженерія програмного забезпечення

Виконала: студентка групи ПЗ2422



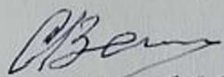
Яна ГАВРИШ

Керівник:



к.т.н., доц. Вадим ГОРЯЧКІН

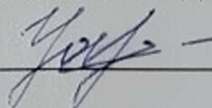
Нормоконтролер:



к. ф-м.н., доц. Світлана ВОЛКОВА

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

Студент



Ministry of Education and Science of Ukraine
Ukrainian State University of Science and Technologies

Computer technologies and systems

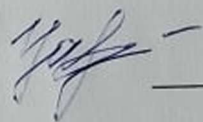
Computer information technologies

Explanatory Note
to Master's Thesis
Master's degree

on the topic: **Research and Implementation of UX/UI Principles to Enhance User Interaction Efficiency with a Tourism Website**

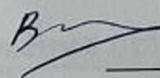
according to educational curriculum 12 Software engineering
in the Speciality: 121 Software engineering

Done by the student of the group: 962M



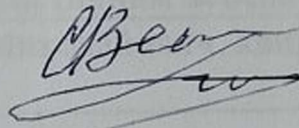
Yana HAVRYSH

Scientific Supervisor:



Vadym HORIACHKIN

Normative controller :



Svitlana VOLKOVA

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: Комп'ютерні технології і системи
Кафедра: Комп'ютерні інформаційні технології
Рівень вищої освіти: Магістр
Освітня програма: F Інженерія програмного забезпечення
Спеціальність: F2 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри _____
_____ Вадим ГОРЯЧКІН

Дата _____

З А В Д А Н Н Я

на кваліфікаційну роботу _____ Магістра
студенту _____ Гавриш Яні Андріївні

1. Тема роботи: _____ Дослідження та впровадження принципів UX/UI для підвищення ефективності взаємодії користувачів з туристичним вебсайтом

Керівник роботи: _____ Горьчкін Вадим Миколайович, кандидат технічних наук, доцент

затверджені наказом від _____ "02" 10 2025 р. _____ №1401ст.

2. Строк подання студентом роботи: _____ 11.01.2026 р.

3. Вихідні дані до роботи: _____ Метрики зручності користування, принципи дизайну

4. Зміст пояснювальної записки (перелік питань, які потрібно опрацювати): _____

Аналіз предметної області, аналіз існуючих рішень,

дослідження стану дизайну сайтів, аналізування поширених проблем користувачів,

метрики та критерії оцінювання,

розробка метрики аналізу, представлення роботи та проведення тестування

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): _____

презентація, відеоролик, представлення результатів аналізу

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Описати вступну частину кваліфікаційної роботи		
2	Виконати аналіз літературних джерел		
3	Виконати дослідження предметної області		
4	Провести дослідження аудиторії	12.11.2025	30%
5	Виконати вибір метрик оцінки ефективності та обґрунтувати їх		
6	Обрати інструменти розробки	09.12.2025	60%
	Розробка додатку	26.12.2025	100%
7	Подання кваліфікаційної роботи до кафедри	19.01.2026	
8	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії	20.01.2026	

Студент _____

_____ Яна ГАВРИШ

Керівник роботи _____

_____ Вадим ГОРЯЧКІН

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи магістра 113 с., 8 рис., 62 джерела.

Об'єктом дослідження є аналіз взаємодії юзерів з туристичними сайтами та зручність інтерфейсів.

Предметом дослідження є принципи дизайну, методи і метрики оцінки зручності користування.

Метою роботи є дослідження і впровадження принципів UX/UI для покращення зручності та ефективності користувачів з сайтами.

Методами дослідження є аналіз та порівняння інших туристичних вебсайтів, застосування метрик для оцінювання зручності

Результатом роботи є розроблене програмне забезпечення для автоматичного аналізу туристичних вебсайтів за посиланням. Програма оцінює зручності користування на основі певних метрик та висуває звіт з рекомендаціями щодо покращення взаємодії користувачів.

Ключові слова: метрики UX, аналітика, зручність користування, аналіз, туристичний вебсайт.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАК, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	9
ВСТУП.....	10
РОЗДІЛ 1 АНАЛІЗ ОСНОВНИХ ПОНЯТЬ ТА МЕТРИК.....	12
1.1 Поняття користувацького досвіду та інтерфейсу користувача	Error! Bookmark not defined.
1.2 Метрики оцінювання зручності користування вебсайтами	Error! Bookmark not defined.
1.2.1 Загальні підходи до класифікації метрик.....	Error! Bookmark not defined.
1.2.2 Кількісні та якісні метрики	Error! Bookmark not defined.
1.2.3 Автоматизовані та експертні метрик.....	Error! Bookmark not defined.
1.2.4 UX-метрики як інструмент оцінювання поведінки користувачів.....	Error! Bookmark not defined.
1.2.5 UI-метрики як характеристика структури інтерфейсу	Error! Bookmark not defined.
1.2.6 Технічні метрики у складі оцінювання зручності користування.....	Error! Bookmark not defined.
1.3 Евристики зручності користування за Нільсеном	Error! Bookmark not defined.

ПЕРЕЛІК УМОВНИХ ПОЗНАК, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

IT – інформаційні технології
UX – користувацький досвід
UI – користувацький інтерфейс
API – інтерфейс програмної взаємодії
GA4 – інструмент вебаналітик
SEO – оптимізація для пошукових систем
WCAG – рекомендації з вебдоступності
REST – архітектурний стиль взаємодії
JSON – формат обміну даними

ВСТУП

Подорожі це вагома частина життя людей, які необхідні для перезавантаження, зміни фокусу, пізнання традицій та культур інших міст або країн. Вони залишаються з нами в спогадах надовго, дають гарні емоції та відкривають нові сторони тебе. В дитинстві час тягнувся довго і можна було багато чого нового встигнути дослідити, та з віком день стає коротше, ти ходиш на роботу і додому, тому твій суб'єктивний час зменшується, а нові емоції (такі як подорожі) розтягують цей час. Тому шукаючи відмовки від поїздки, думай про те що твій час життя стає коротшим і треба ловити ці моменти.

Тому, в сучасному світі діджиталізації всіх галузей, вебсайти та додатки стали основним зв'язком користувачей з сервісами. Через туристичні вебплатформи відбувається пошук інформації, бронювання турів, планування поїздки та спілкування з менеджерами. У зв'язку з цим зручність та якість користування мають велике значення для досягнення придбання вашими послугами.

Із розвитком вебтехнологій зростає складність інтерфейсів туристичних вебсайтів, що нерідко призводить до перевантаження інформацією, ускладнення навігації та зниження ефективності взаємодії користувачів із системою. Недостатня увага до принципів UX/UI дизайну може призводити до втрати потенційних клієнтів, зниження рівня довіри та погіршення показників конверсії. Тому актуальною є задача оцінювання зручності користування туристичними вебсайтами та виявлення проблемних аспектів їх інтерфейсів.

Актуальність дослідження полягає у необхідності застосування об'єктивних метрик та автоматизованих методів аналізу UX/UI для оцінювання ефективності взаємодії користувачів з туристичними вебресурсами. Використання програмних інструментів для аналізу вебінтерфейсів дозволяє отримувати кількісні показники продуктивності, доступності та стабільності інтерфейсу, що є основою для прийняття обґрунтованих рішень щодо їх покращення.

Об'єктом дослідження є туристичні вебсайти та процеси взаємодії користувачів з вебінтерфейсами.

Предметом дослідження є принципи UX/UI дизайну, методи та метрики оцінювання зручності користування туристичними вебсайтами, а також програмні засоби автоматизованого аналізу користувацького досвіду.

Метою роботи є дослідження та впровадження принципів UX/UI дизайну для підвищення ефективності взаємодії користувачів з туристичним вебсайтом шляхом розробки ПЗ автоматизованого аналізу зручності користування.

Для досягнення поставленої мети у роботі необхідно вирішити такі завдання:

- проаналізувати сучасний стан UX/UI дизайну туристичних вебсайтів;
- дослідити основні проблеми взаємодії користувачів з туристичними вебінтерфейсами;
- обґрунтувати вибір метрик оцінювання зручності користування, доступності та продуктивності вебсайтів;
- розробити програмний інструмент для автоматизованого аналізу UX/UI туристичних вебсайтів;
- провести тестування програмного інструмента на прикладі реальних туристичних вебресурсів;
- здійснити оцінку ефективності взаємодії користувачів з вебінтерфейсами на основі отриманих результатів.

Методами дослідження є аналіз наукових і прикладних джерел з UX/UI дизайну, порівняльний аналіз туристичних вебсайтів, застосування метрик користувацького досвіду, методи програмної реалізації та тестування.

Практичне значення роботи полягає у можливості використання розробленого програмного інструмента для аналізу зручності користування туристичними вебсайтами, а також для формування рекомендацій щодо вдосконалення їх UX/UI дизайну.

РОЗДІЛ 1 АНАЛІЗ ОСНОВНИХ ПОНЯТЬ ТА МЕТРИК

1.1 Поняття користувацького досвіду та інтерфейсу користувача

Користувацький досвід (User Experience, UX) визначає сукупність відчуттів, сприйняття, емоцій та поведінкових реакцій користувача, які виникають у процесі взаємодії з вебсайтом або цифровим сервісом. UX охоплює не лише зовнішній вигляд інтерфейсу, але й логіку побудови інформаційної структури, зручність навігації, швидкість виконання дій, зрозумілість елементів управління та відповідність очікуванням користувача. Якісний користувацький досвід сприяє швидкому досягненню цілей користувача та формуванню позитивного загального враження від використання вебресурсу.

Інтерфейс користувача (User Interface, UI) є візуальною та інтерактивною складовою вебсайту, через яку реалізується взаємодія між користувачем і системою. UI включає в себе графічні елементи, кольорову гаму, типографіку, кнопки, форми введення даних, іконки та інші компоненти, що забезпечують керування функціональністю вебсайту. Основною задачею є забезпечення зрозумілості, доступності та естетичної привабливості інтерфейсу.

UX та UI тісно пов'язані між собою, однак не є тотожними поняттями. UI визначає форму подання інформації та елементів управління, тоді як UX характеризує ефективність і зручність використання цих елементів у контексті конкретних користувацьких сценаріїв. Навіть візуально привабливий інтерфейс може створювати негативний користувацький досвід у разі нелогічної структури або перевантаження інформацією.

У туристичних вебсайтах набуває особливої ваги, оскільки користувачі здійснюють складні багатокрокові дії, зокрема пошук напрямків подорожей, порівняння пропозицій, вибір дат, бронювання послуг та здійснення платежів. Низька якість UX/UI може призводити до переривання процесу бронювання, зниження довіри до сервісу та

втрати потенційних клієнтів. Навпаки, інтуїтивно зрозумілий інтерфейс і добре продуманий користувацький досвід сприяють підвищенню рівня конверсії та задоволеності користувачів.

Особливістю вебресурсів є також необхідність роботи з великою кількістю інформації та мультимедійного контенту, що вимагає від UX/UI дизайну забезпечення швидкого доступу до ключових функцій без перевантаження користувача. Це робить актуальним використання формалізованих підходів і метрик для оцінювання зручності користування такими вебсайтами.

Таблиця 1.1 — Порівняльна характеристика UX та UI

Критерій	UX (User Experience)	UI (User Interface)
Основне призначення	Забезпечення зручності та ефективності взаємодії	Візуальне та інтерактивне представлення
Об'єкт впливу	Поведінка, сприйняття та емоції користувача	Графічні елементи та компоненти інтерфейсу
Основні аспекти	Навігація, логіка структури, користувацькі сценарії	Кольори, шрифти, кнопки, іконки
Вплив на результат	Визначає задоволеність та досягнення цілей	Впливає на зрозумілість та естетику
Можливість автоматизованої оцінки	Часткова (через метрики та показники)	Часткова (через структуру та контрастність)

Отримані положення свідчать про те, що ефективний аналіз туристичних вебсайтів повинен враховувати як UX, так і UI складові. Саме це обґрунтовує необхідність використання програмного інструмента, здатного автоматизовано оцінювати ключові характеристики користувацького досвіду на основі визначених метрик.

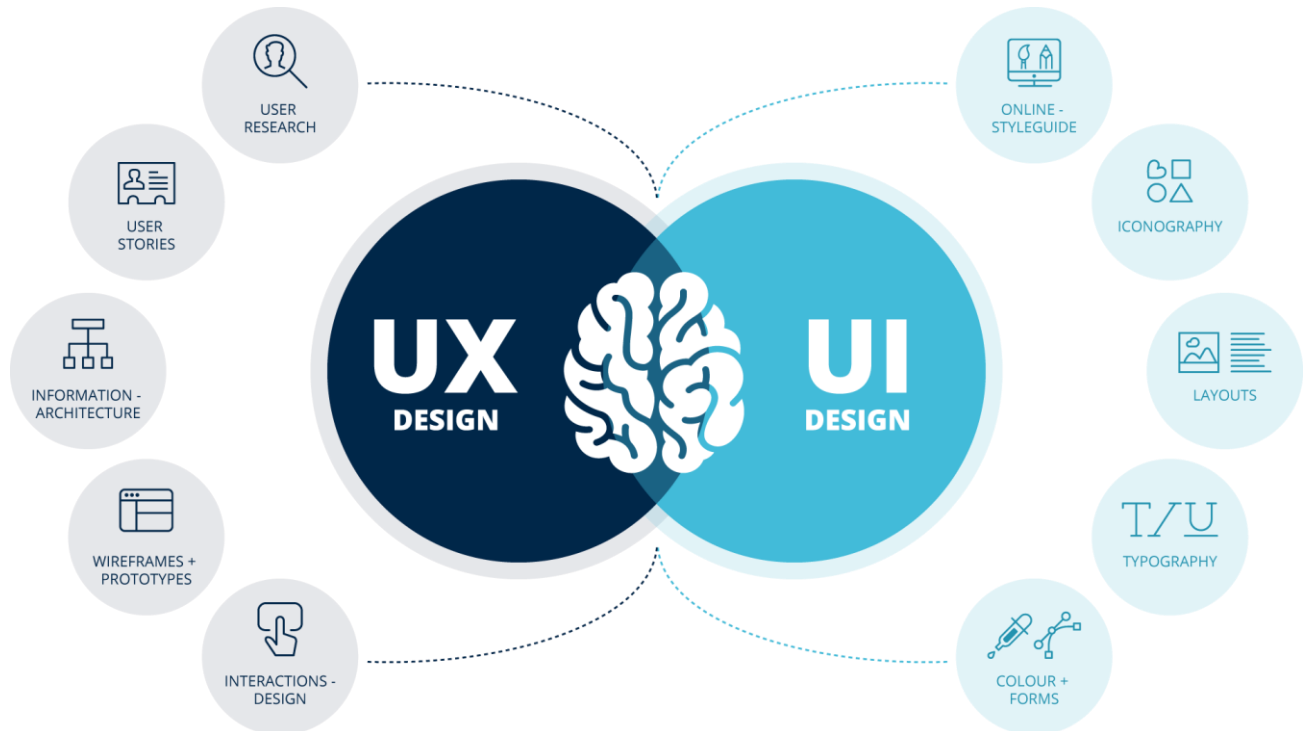


Рисунок 1.1 – структура UX та UI

1.2 Метрики оцінювання зручності користування вебсайтами

Оцінювання зручності користування вебсайтами є важливим етапом проектування та вдосконалення вебінтерфейсів. Для об'єктивного аналізу ефективності взаємодії користувачів із вебресурсами застосовується система метрик, які дозволяють кількісно та якісно описати характеристики користувацького досвіду та інтерфейсу.

Метрики зручності користування є інструментами вимірювання, що відображають ступінь відповідності вебінтерфейсу очікуванням користувачів, легкість виконання цільових дій та загальний рівень задоволеності взаємодією із системою. Застосування таких метрик дозволяє виявляти проблемні аспекти інтерфейсу, порівнювати різні рішення та оцінювати ефективність змін.

1.2.1 Загальні підходи до класифікації метрик

Метрики можуть бути поділені за різними критеріями залежно від характеру оцінювання, способу збору даних та рівня деталізації результатів.

Таблиця 1.2 — Основні підходи до класифікації метрик зручності користування

Критерій класифікації	Типи метрик
За формою даних	Кількісні, якісні
За способом збору	Автоматизовані, експертні
За об'єктом оцінювання	UX-метрики, UI-метрики
За рівнем аналізу	Поведінкові, інтерфейсні
За можливістю автоматизації	Повністю автоматизовані, частково автоматизовані

1.2.2 Кількісні та якісні метрики

Кількісні метрики базуються на числових показниках і дозволяють здійснювати об'єктивне порівняння різних вебінтерфейсів або версій одного ресурсу. Вони широко застосовуються для вимірювання часу виконання дій, кількості помилок, швидкості реакції системи та інших параметрів, що можуть бути зафіксовані автоматично.

Якісні метрики орієнтовані на суб'єктивне сприйняття користувачів і відображають рівень задоволеності, зрозумілості інтерфейсу та комфортність взаємодії. Такі метрики часто формуються на основі опитувань, експертних оцінок або аналізу користувацьких сценаріїв.

Таблиця 1.3 — Порівняльна характеристика кількісних та якісних метрик

Характеристика	Кількісні метрики	Якісні метрики
Тип даних	Числові значення	Описові оцінки
Рівень об'єктивності	Високий	Середній
Можливість автоматизації	Висока	Обмежена
Простота аналізу	Висока	Залежить від контексту

1.2.3 Автоматизовані та експертні метрики

Автоматизовані метрики отримуються шляхом програмного аналізу структури вебсайту, часу завантаження сторінок, стабільності інтерфейсу та інших технічних характеристик. Вони дозволяють здійснювати масштабний аналіз без залучення користувачів і є зручними для регулярного моніторингу зручності користування.

Експертні метрики формуються на основі аналізу інтерфейсу фахівцями у сфері UX/UI дизайну, які оцінюють відповідність вебінтерфейсу визначеним принципам і евристикам. Такий підхід дозволяє виявляти проблеми, які складно формалізувати автоматично.

1.2.4 UX-метрики як інструмент оцінювання поведінки користувачів

UX-метрики орієнтовані на аналіз поведінкових характеристик користувачів під час взаємодії з вебінтерфейсом. Вони дозволяють оцінити, наскільки ефективно користувачі виконують цільові дії та з якими труднощами стикаються.

До основних UX-метрик належать:

- час виконання завдань;
- кількість кроків до досягнення результату;
- частота помилок;
- показники відмов;
- рівень завершення сценаріїв.

1.2.5 UI-метрики як характеристика структури інтерфейсу

UI-метрики використовуються для оцінювання візуальної організації інтерфейсу та якості подання інформації. Вони дозволяють аналізувати читабельність тексту, контрастність, узгодженість стилів та логіку розміщення елементів керування.

Застосування UI-метрик сприяє зменшенню когнітивного навантаження та підвищенню інтуїтивності інтерфейсу.

1.2.6 Технічні метрики у складі оцінювання зручності користування

Технічні метрики відображають якість програмної реалізації вебресурсу та безпосередньо впливають на користувацький досвід. Навіть добре спроектований інтерфейс може бути незручним у разі повільної роботи або нестабільної верстки.

До технічних метрик належать показники швидкості завантаження сторінок, стабільності інтерфейсу, оптимізації ресурсів та коректності коду.

1.3 Евристики зручності користування за Нільсеном

Одним із найбільш відомих і широко застосовуваних підходів до оцінювання зручності користування вебінтерфейсами є система евристик, запропонована дослідником у галузі юзабіліті Якобом Нільсеном. Евристики Нільсена являють собою набір загальних принципів, які описують ключові вимоги до проєктування інтерфейсів з точки зору зручності використання. Вони застосовуються як для експертної оцінки інтерфейсів, так і як теоретична основа для формалізації критеріїв аналізу UX/UI.

Основна мета евристик Нільсена полягає у виявленні типових проблем взаємодії користувачів з інтерфейсом на ранніх етапах проєктування або під час аналізу вже реалізованих вебресурсів. На відміну від методів тестування за участю користувачів, евристичний аналіз дозволяє швидко оцінити інтерфейс без значних ресурсних витрат, що робить його зручним інструментом первинної оцінки зручності користування.

Евристики Нільсена не є жорсткими правилами, а слугують загальними рекомендаціями, які допомагають експертам оцінювати інтерфейси з позиції користувача. Вони можуть застосовуватися до різних типів цифрових продуктів, зокрема вебсайтів, вебзастосунків та мобільних інтерфейсів.

1.3.1 Видимість стану системи

Однією з ключових евристик є забезпечення постійної інформованості користувача про поточний стан системи. Інтерфейс повинен своєчасно надавати зворотний зв'язок щодо виконання дій користувача, зокрема повідомляти про процеси завантаження, обробку запитів або результати виконаних операцій.

Відсутність зворотного зв'язку може призводити до невпевненості користувача, повторних дій або помилкових припущень щодо працездатності системи. Тому індикатори стану, повідомлення про успішне виконання дій та візуальні підказки є важливими елементами зручного інтерфейсу.

1.3.2 Відповідність між системою та реальним світом

Ця евристика передбачає використання у інтерфейсі зрозумілих користувачеві термінів, символів і концепцій, що відповідають реальному життєвому досвіду. Інтерфейс повинен «говорити мовою користувача», уникаючи технічної термінології та абстрактних позначень.

Дотримання цієї евристики зменшує когнітивне навантаження та сприяє інтуїтивному розумінню функціональності інтерфейсу без додаткових пояснень.

1.3.3 Контроль і свобода користувача

Користувачі часто здійснюють помилкові дії, тому інтерфейс повинен надавати можливість скасування або коригування виконаних операцій. Наявність механізмів повернення до попереднього стану підвищує відчуття контролю над системою та знижує рівень фрустрації. Ця евристика особливо важлива для багатокрокових процесів, де помилка на одному з етапів не повинна призводити до необхідності починати взаємодію з самого початку.

1.3.4 Послідовність та стандарти

Інтерфейс має бути послідовним у використанні елементів керування, термінології та візуальних рішень. Користувачі очікують, що однакові дії виконуватимуться однаковим способом, а знайомі елементи працюватимуть відповідно до загальноприйнятих стандартів. Недотримання цієї евристики призводить до плутанини та ускладнює процес взаємодії з інтерфейсом.

1.3.5 Запобігання помилкам

Інтерфейс повинен бути спроектований таким чином, щоб мінімізувати ймовірність виникнення помилок з боку користувача. Це досягається шляхом

обмеження некоректних дій, використання перевірок введених даних та попереджень про потенційні наслідки. Запобігання помилкам є більш ефективним підходом, ніж їх подальше виправлення, оскільки зменшує кількість негативних сценаріїв взаємодії.

1.3.6 Розпізнавання замість запам'ятовування

Інтерфейс має зменшувати навантаження на пам'ять користувача, пропонуючи підказки, видимі параметри та контекстну інформацію. Користувач не повинен запам'ятовувати інформацію між різними етапами взаємодії. Реалізація цієї евристики сприяє підвищенню доступності інтерфейсу та зручності його використання.

1.3.7 Гнучкість та ефективність використання

Інтерфейс повинен бути адаптованим як для новачків, так і для досвідчених користувачів. Це означає наявність як базових, так і розширених можливостей взаємодії, які дозволяють прискорити виконання типових дій. Гнучкість інтерфейсу сприяє підвищенню ефективності роботи користувачів із системою.

1.3.8 Естетичний і мінімалістичний дизайн

Інтерфейс не повинен містити зайвих елементів або інформації, які не сприяють досягненню цілей користувача. Надмірна кількість деталей може відволікати увагу та ускладнювати сприйняття інтерфейсу. Мінімалістичний дизайн дозволяє зосередити увагу користувача на ключових функціях системи.

1.3.9 Допомога у виявленні та виправленні помилок

Повідомлення про помилки повинні бути зрозумілими, інформативними та містити рекомендації щодо подальших дій. Вони мають пояснювати причину помилки та шляхи її усунення без використання складної технічної термінології.

1.3.10 Довідка та документація

Навіть інтуїтивно зрозумілий інтерфейс повинен мати доступні довідкові матеріали та пояснення, які допомагають користувачам у разі виникнення труднощів. Довідка повинна бути легко доступною та структурованою.

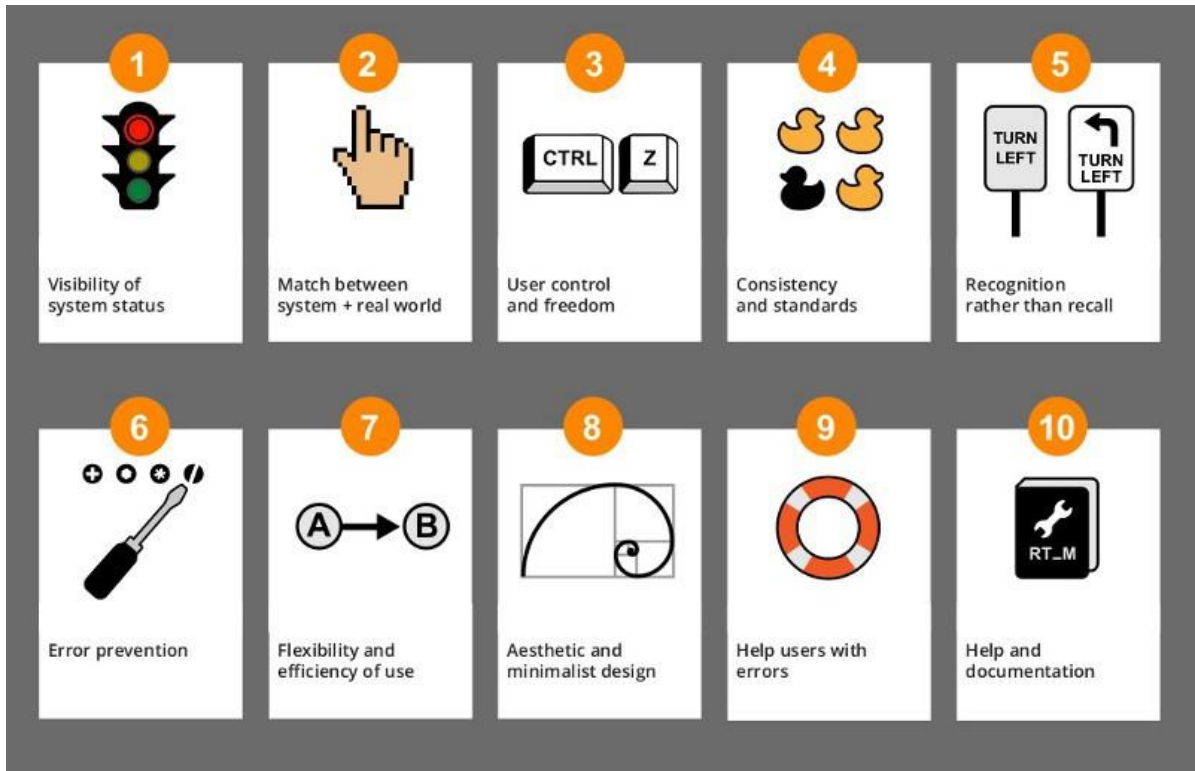


Рисунок 1.2 – 10 Евристик Якоба Нільсена

1.4 Метрики продуктивності вебінтерфейсів (Core Web Vitals)

Продуктивність вебінтерфейсу є одним із ключових чинників, що безпосередньо впливають на зручність користування цифровими продуктами. Навіть добре спроектований інтерфейс може створювати негативний користувацький досвід у разі повільного завантаження сторінок, затримок у реакції на дії користувача або нестабільного відображення елементів інтерфейсу. У зв'язку з цим оцінювання продуктивності є важливою складовою комплексного аналізу якості вебресурсів.

Для стандартизації підходів до оцінювання продуктивності вебсайтів було запропоновано набір метрик Core Web Vitals, які орієнтовані на вимірювання ключових аспектів взаємодії користувачів з вебінтерфейсами. Ці метрики дозволяють кількісно оцінити швидкість завантаження контенту, стабільність інтерфейсу та швидкість реакції системи на дії користувача.

Core Web Vitals охоплюють три основні показники, кожен з яких відповідає за окремий аспект користувацького досвіду. Використання цих метрик дозволяє здійснювати об'єктивне порівняння вебінтерфейсів та виявляти проблемні зони, що впливають на ефективність взаємодії.

Основні метрики Core Web Vitals

Таблиця 1.10 — Характеристика метрик Core Web Vitals

Метрика	Опис	Аспект UX
LCP (Largest Contentful Paint)	Час завантаження основного контенту сторінки	Швидкість
CLS (Cumulative Layout Shift)	Стабільність розміщення елементів під час завантаження	Візуальна стабільність
INP (Interaction to Next Paint)	Час реакції інтерфейсу на дію користувача	Інтерактивність

Метрика LCP дозволяє оцінити, наскільки швидко користувач бачить основний вміст сторінки. Високі значення цього показника негативно впливають на сприйняття швидкодії системи.

Метрика CLS відображає ступінь зміщення елементів інтерфейсу під час завантаження сторінки. Значні зсуви можуть призводити до помилкових дій користувача та погіршення загального враження від взаємодії.

Метрика INP характеризує швидкість реакції системи на взаємодію користувача з інтерфейсом. Високі значення цього показника свідчать про затримки, які негативно впливають на зручність користування.

Окрім основних показників Core Web Vitals, для аналізу продуктивності також використовуються допоміжні метрики, зокрема час відповіді сервера та загальний час завантаження сторінки.

Таблиця 1.11 — Додаткові метрики продуктивності

Метрика	Характеристика
TTFB (Time To First Byte)	Час отримання першої відповіді від сервера
FCP (First Contentful Paint)	Час відображення першого елемента контенту
Speed Index	Загальна швидкість візуального заповнення сторінки

Таким чином, метрики продуктивності забезпечують кількісну основу для оцінювання швидкодії та стабільності вебінтерфейсів і є важливим елементом комплексного аналізу зручності користування.

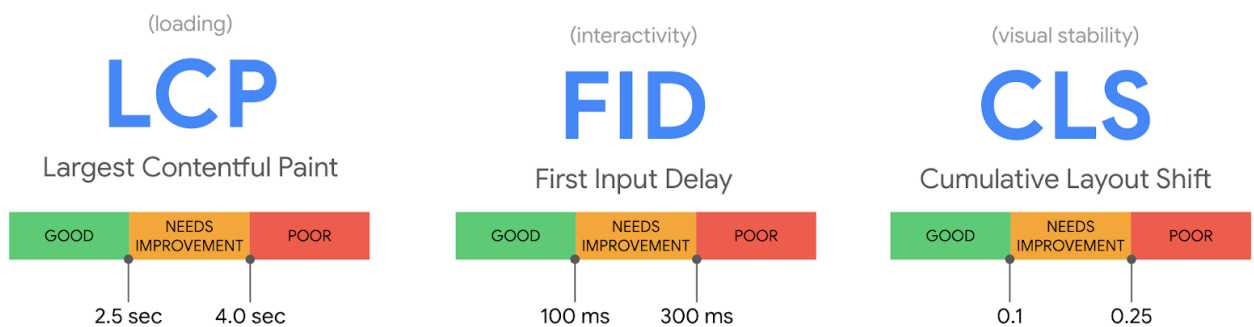


Рисунок 1.3 – метрика Core Web Vitals

1.5 Метрики доступності вебінтерфейсів (WCAG)

Доступність вебінтерфейсів є важливою складовою якості цифрових продуктів і визначає можливість їх використання користувачами з різними фізичними, сенсорними та когнітивними особливостями. Забезпечення доступності спрямоване на усунення бар'єрів взаємодії та створення рівних умов доступу до інформації та функціональності вебресурсів.

Для уніфікації підходів до оцінювання доступності вебсайтів розроблено рекомендації WCAG (Web Content Accessibility Guidelines), які визначають базові принципи та критерії доступності вебконтенту. WCAG є міжнародним стандартом, що використовується як основа для розробки, аналізу та вдосконалення вебінтерфейсів.

Рекомендації WCAG ґрунтуються на чотирьох основних принципах, які охоплюють ключові аспекти взаємодії користувачів з вебресурсами.

Основні принципи WCAG

Таблиця 1.12 — Принципи доступності за WCAG

Принцип	Характеристика
Сприйнятність (Perceivable)	Інформація має бути доступною для сприйняття
Керованість (Operable)	Інтерфейс повинен бути керованим
Зрозумілість (Understandable)	Контент і навігація мають бути зрозумілими
Надійність (Robust)	Сумісність із різними пристроями та технологіями

Принцип сприйнятності передбачає забезпечення альтернативних способів подання інформації, зокрема текстових описів для графічних елементів. Керованість означає можливість використання інтерфейсу різними способами, включаючи керування з клавіатури. Зрозумілість орієнтована на чіткість структури та передбачуваність взаємодії. Надійність забезпечує коректну роботу вебінтерфейсу з різними програмними та апаратними засобами.

Критерії оцінювання доступності

Таблиця 1.13 — Приклади критеріїв доступності WCAG

Критерій	Опис
Контрастність	Читабельність тексту
Альтернативний текст	Доступність зображень
Семантична розмітка	Коректність структури HTML
Навігація з клавіатури	Доступність керування
Підказки та помилки	Зрозумілі повідомлення

Метрики доступності дозволяють оцінювати відповідність вебінтерфейсів стандартам інклюзивності та виявляти проблеми, які можуть ускладнювати використання цифрових продуктів окремими категоріями користувачів.

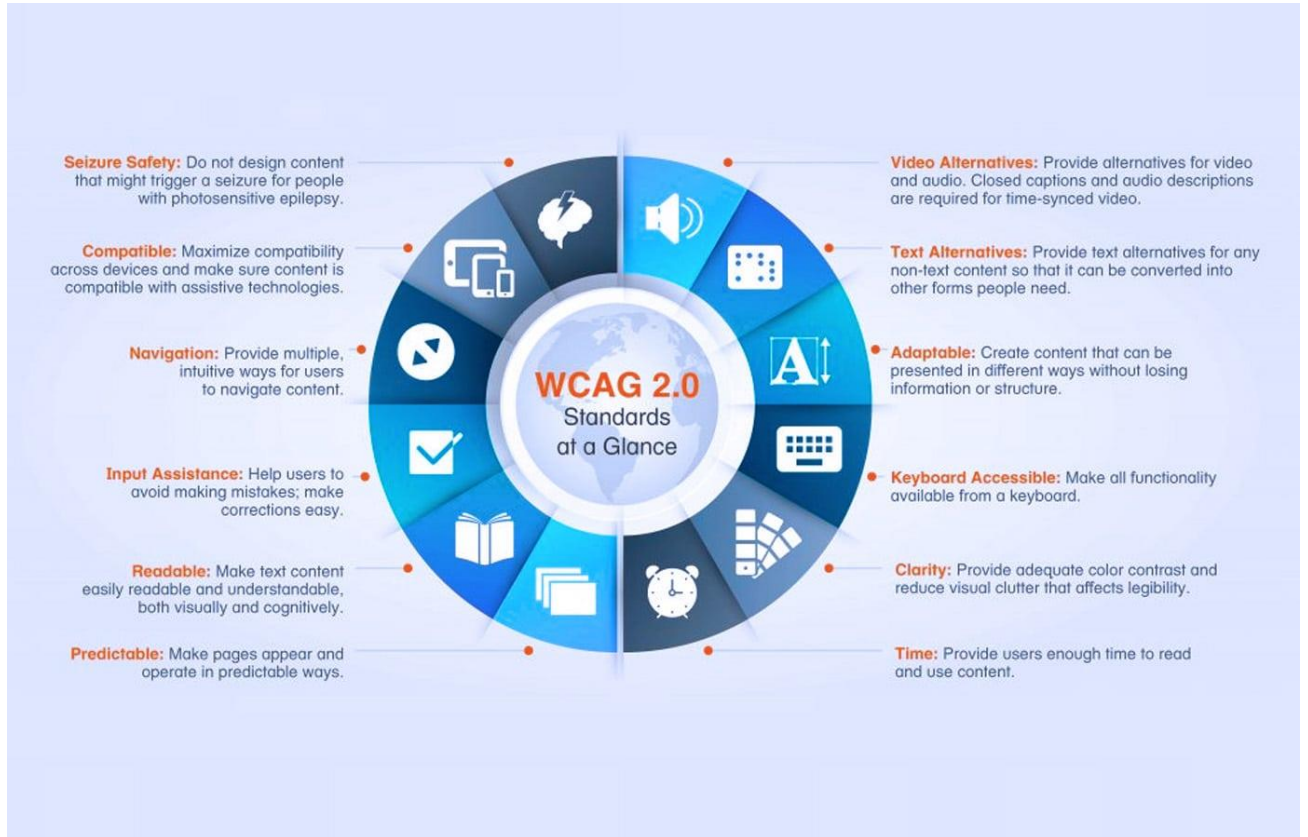


Рисунок 1.3 – метрика WCAG

1.6 Метрики оцінювання якості сайтів на основі Lighthouse

Для комплексного оцінювання якості інтерфейсів доцільно застосовувати інтегровані метрики, які дозволяють одночасно враховувати як користувацькі, так і технічні характеристики цифрових продуктів. У сучасній практиці вебаудиту одним із найбільш поширених інструментів такого типу є Google Lighthouse, що забезпечує автоматизований аналіз вебресурсів на основі стандартизованого набору перевірок і показників.

Google Lighthouse не є окремою метрикою, а представляє собою комплексний інструмент оцінювання, який поєднує в собі сукупність окремих кількісних і якісних

показників, а також алгоритми їх інтерпретації. У процесі аналізу Lighthouse виконує серію перевірок (audits), що охоплюють різні аспекти функціонування вебінтерфейсу, після чого результати цих перевірок агрегуються у вигляді узагальнених числових оцінок.

Основною особливістю Lighthouse є багаторівнева структура оцінювання. На нижньому рівні використовуються конкретні технічні метрики, такі як час завантаження контенту, стабільність верстки, швидкість реакції інтерфейсу та коректність семантичної розмітки. На основі цих показників формуються окремі перевірки, що відображають відповідність вебінтерфейсу певним вимогам і рекомендаціям. Далі результати перевірок агрегуються у кілька узагальнених категорій, які і становлять підсумкову оцінку Lighthouse.

Результати аналізу подаються у вигляді чотирьох основних оцінок, кожна з яких має шкалу від 0 до 100 балів. Такий формат подання даних є зручним для порівняльного аналізу вебресурсів, оскільки дозволяє швидко оцінити загальний стан інтерфейсу та виявити сильні й слабкі сторони його реалізації.

1.6.1 Метрика Performance

Метрика Performance відображає швидкодію вебінтерфейсу та характеризує те, наскільки швидко користувач отримує доступ до основного контенту сторінки. Вона формується на основі технічних показників завантаження та реакції інтерфейсу, які дозволяють оцінити ефективність використання ресурсів вебсторінки. З точки зору користувацького досвіду, високе значення Performance сприяє формуванню позитивного першого враження та зменшує ймовірність передчасного припинення взаємодії з вебресурсом. Низькі значення цієї метрики можуть свідчити про надмірне навантаження сторінки, неоптимізовані зображення або неефективне використання скриптів.

1.6.2 Метрика Accessibility

Метрика Accessibility характеризує рівень доступності вебінтерфейсу для користувачів з різними фізичними та когнітивними особливостями. Вона ґрунтується на

міжнародних рекомендаціях з вебдоступності та враховує аспекти читабельності, керованості та зрозумілості інтерфейсу. Оцінювання доступності дозволяє виявити бар'єри взаємодії, які можуть ускладнювати використання вебресурсу окремими категоріями користувачів. Забезпечення доступності сприяє не лише інклюзивності, але й підвищує загальну зручність користування вебінтерфейсом.

1.6.3 Метрика Best Practices

Метрика Best Practices відображає відповідність вебінтерфейсу сучасним рекомендаціям і стандартам веброзробки. Вона охоплює перевірки, пов'язані з безпекою, коректністю реалізації функціональності та використанням актуальних технологій. Високі значення цієї метрики свідчать про стабільність і надійність вебресурсу, тоді як низькі можуть вказувати на потенційні технічні ризики або застарілі підходи до реалізації інтерфейсу.

1.6.4 Метрика SEO

Метрика SEO характеризує рівень оптимізації вебінтерфейсу для пошукових систем. Вона оцінює структурну коректність сторінки, наявність метаданих, логічну ієрархію заголовків та інші фактори, що впливають на індексацію вебресурсу. Хоча SEO не є класичною UX-метрикою, вона опосередковано впливає на користувацький досвід, оскільки добре оптимізований вебресурс є легшим для пошуку та доступнішим для користувачів.

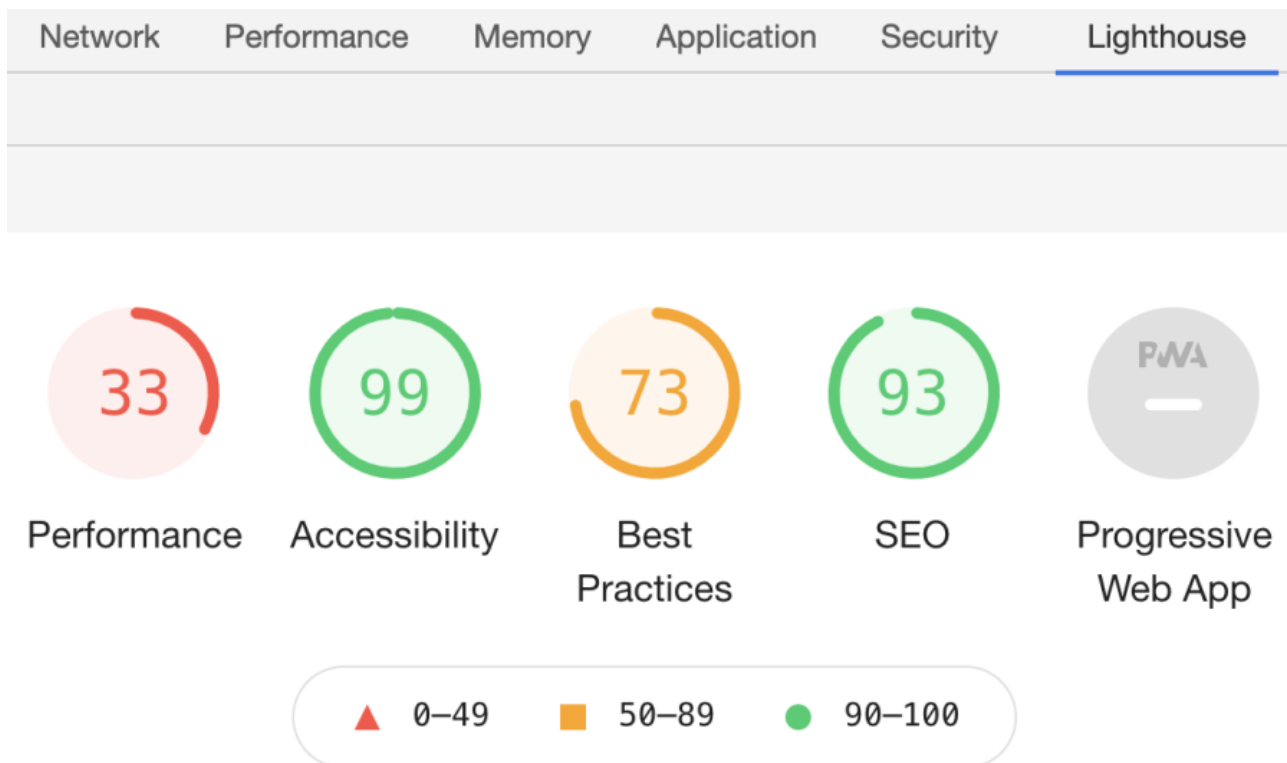


Рисунок 1.3 – приклад аналізу по Lighthouse

Висновки до першого розділу

У першому розділі розглянуто теоретичні основи користувацького досвіду та інтерфейсу користувача у контексті вебсайтів туристичної тематики. Проаналізовано поняття UX та UI, їх взаємозв'язок і роль у забезпеченні ефективної взаємодії з вебресурсами. Показано, що якісний користувацький досвід формується не лише візуальним оформленням інтерфейсу, а й логікою інформаційної структури, зручністю навігації та швидкістю виконання дій.

Систематизовано основні підходи до оцінювання зручності користування вебсайтами на основі кількісних, якісних та експертних метрик. Встановлено, що кількісні метрики забезпечують об'єктивний аналіз поведінки користувачів і технічних характеристик вебінтерфейсу, тоді як якісні та експертні дозволяють виявляти проблеми сприйняття, зрозумілості та логіки взаємодії, які складно формалізувати автоматично.

Розглянуто систему евристик зручності користування за Якобом Нільсеном як ефективний інструмент експертної оцінки UX/UI. Показано, що евристичний аналіз дає

змогу швидко виявляти типові проблеми інтерфейсу, пов'язані з перевантаженням інформацією, недостатнім зворотним зв'язком і порушенням послідовності елементів керування.

Проаналізовано метрики продуктивності вебінтерфейсів, зокрема Core Web Vitals, а також метрики доступності відповідно до рекомендацій WCAG. Встановлено, що продуктивність і доступність вебресурсу безпосередньо впливають на якість користувацького досвіду та ефективність взаємодії користувачів з інтерфейсом. Вказано про підхід до оцінювання якості вебінтерфейсів на основі інструмента Google Lighthouse, який поєднує аналіз продуктивності, доступності, відповідності кращим практикам та SEO.

Отримані результати сформували теоретичне підґрунтя для подальшого розроблення програмного інструмента автоматизованого UX/UI-аудиту туристичних вебсайтів та визначили набір метрик, що використовуватимуться у практичній частині дослідження.

РОЗДІЛ 2 ЗАГАЛЬНІ МЕТОДИ ПРОВЕДЕННЯ НАУКОВОГО ДОСЛІДЖЕННЯ

2.1 Обґрунтування вибору напрямку

Зростання ролі цифрових сервісів у різних сферах діяльності зумовлює підвищену увагу до якості взаємодії користувачів із вебінтерфейсами. У сучасних умовах вебсайти та онлайн-платформи виконують не лише інформаційну функцію, а й є повноцінними інструментами для надання послуг, прийняття рішень та здійснення фінансових операцій. У зв'язку з цим зручність користування вебінтерфейсами стає одним із ключових чинників ефективності цифрових продуктів.

Традиційно оцінювання якості UX/UI здійснюється шляхом експертного аналізу або тестування за участю користувачів. Такий підхід дозволяє отримати глибоке розуміння проблем взаємодії, однак має низку обмежень. Проведення експертних оцінок потребує значних часових і кадрових ресурсів, а результати часто мають суб'єктивний характер і залежать від досвіду та кваліфікації конкретного фахівця. Крім того, ручний аналіз є складним для масштабування та ускладнює повторне проведення дослідження за однакових умов.

У контексті дослідження зручності користування актуальним є застосування автоматизованих методів аналізу, які дозволяють отримувати кількісні показники якості вебінтерфейсів без безпосередньої участі експертів або користувачів. Автоматизований підхід забезпечує повторюваність результатів, зменшує вплив людського фактору та дозволяє проводити порівняльний аналіз великої кількості вебресурсів за однаковими критеріями.

Використання кількісних метрик є важливим елементом аналізу UX/UI, оскільки вони дозволяють перейти від суб'єктивних оцінок до формалізованих показників. Кількісні метрики забезпечують можливість порівняння результатів, побудови узагальнених оцінок та виявлення тенденцій у зміні якості вебінтерфейсів. Такий підхід є особливо доцільним в межах наукового дослідження, де важливо забезпечити об'єктивність і відтворюваність отриманих результатів.

Разом із тим, повна автоматизація оцінювання користувацького досвіду є складним завданням, оскільки значна частина UX-характеристик має якісний і контекстний характер. У зв'язку з цим доцільним є поєднання узагальнених автоматизованих метрик із аналізом структурних і технічних характеристик вебінтерфейсу. Такий комбінований підхід дозволяє частково формалізувати оцінювання UX/UI без втрати логічного зв'язку з теоретичними принципами зручності користування.

В межах даного дослідження як основний інструмент автоматизованого аналізу обрано Google Lighthouse, який забезпечує комплексну оцінку вебінтерфейсів за низкою стандартизованих критеріїв. Lighthouse дозволяє отримувати узагальнені оцінки продуктивності, доступності, якості технічної реалізації та структурної оптимізації вебресурсів. Важливою перевагою цього інструмента є можливість автоматизованого збору даних та уніфікований підхід до їх інтерпретації.

Оскільки узагальнені оцінки Lighthouse не охоплюють усіх аспектів структури та контенту вебсторінок, у дослідженні також використовується парсинг HTML-коду для збору додаткових показників. Застосування парсера дозволяє отримати інформацію про структуру сторінки, наявність ключових елементів інтерфейсу та коректність використання базових вебстандартів. Поєднання результатів Lighthouse та даних парсингу створює більш повну картину якості вебінтерфейсу.

Тому, вибір напряму дослідження зосереджений на розробці автоматизованого підходу до оцінювання зручності користування вебінтерфейсами, який поєднує кількісні метрики та структурний аналіз. Застосування такого підходу дозволяє забезпечити об'єктивність оцінювання, зменшити вплив суб'єктивних чинників і створити основу для подальшої розробки програмного засобу автоматизованого аналізу UX/UI.

2.2 Вибір метрик оцінювання зручності користування

Оцінювання зручності користування вебінтерфейсами є багатокomпонентним процесом, який передбачає врахування як користувацьких, так і технічних

характеристик цифрового продукту. На відміну від суто технічних параметрів, зручність користування має значну частку якісних характеристик, що ускладнює її формалізацію та автоматизоване вимірювання.

Вибір метрик оцінювання ґрунтується на необхідності поєднання теоретичних принципів UX/UI з можливостями автоматизованого аналізу. Такий підхід дозволяє забезпечити логічний зв'язок між якісними характеристиками користувацького досвіду та кількісними показниками, які можуть бути отримані програмним шляхом.

Основною вимогою до обраних метрик є їх об'єктивність, повторюваність та можливість застосування до різних вебресурсів за однакових умов. Це дозволяє здійснювати порівняльний аналіз і формувати узагальнені висновки щодо зручності користування.

2.2.1 UX/UI-метрики

Теоретичною основою для оцінювання зручності користування вебінтерфейсами в дослідженні слугують евристики зручності користування, які описують ключові принципи ефективної взаємодії користувача з інтерфейсом. Ці принципи охоплюють такі аспекти, як зрозумілість структури, логічність навігації, передбачуваність поведінки інтерфейсу та мінімізація когнітивного навантаження.

Евристики зручності користування є ефективним інструментом експертного аналізу, однак їх безпосереднє застосування у програмних засобах є обмеженим. Це пов'язано з тим, що більшість UX/UI-принципів мають якісний характер і залежать від контексту використання, особливостей цільової аудиторії та конкретних користувацьких сценаріїв. У зв'язку з цим повна автоматизація оцінювання зручності користування на основі евристик є неможливою.

Для подолання зазначеного обмеження в дослідженні використовується підхід часткової формалізації UX/UI-принципів шляхом їх інтерпретації через кількісні показники. Такий підхід передбачає відображення окремих аспектів користувацького досвіду у вигляді метрик, які можуть бути автоматично виміряні. Зокрема, швидкодія

інтерфейсу, стабільність відображення елементів, доступність контенту та структурна впорядкованість сторінки розглядаються як непрямі показники зручності користування.

Перехід від якісних евристик до формалізованих метрик дозволяє зберегти зв'язок із теоретичними принципами UX/UI та водночас забезпечити можливість автоматизованого аналізу. В межах програмного засобу це реалізується через використання узагальнених оцінок і окремих технічних показників, які відображають ключові характеристики взаємодії користувача з вебінтерфейсом.

Таблиця 2.4 — Відповідність UX/UI-евристик та формалізованих показників

UX/UI-принцип	Формалізований показник
Видимість стану системи	Час завантаження контенту
Мінімізація когнітивного навантаження	Розмір DOM, стабільність верстки
Послідовність і стандарти	Структура заголовків
Запобігання помилкам	Доступність форм і елементів
Доступність інформації	Контрастність та alt-тексти

Застосування формалізованих показників не забезпечує повного відтворення користувацького досвіду, однак дозволяє об'єктивно оцінювати окремі його складові. У поєднанні з узагальненими метриками продуктивності та доступності такий підхід створює основу для побудови автоматизованого інструмента аналізу зручності користування вебінтерфейсами. Тому, вибір UX/UI-метрик базується на поєднанні теоретичних принципів зручності користування та можливостей їх часткової автоматизації, що забезпечує баланс між науковою обґрунтованістю та практичною реалізованістю програмного засобу.

2.2.2 Метрики продуктивності (Core Web Vitals)

Продуктивність вебінтерфейсу є одним із ключових чинників, що впливають на зручність користування та загальне сприйняття цифрового продукту. Швидкодія

інтерфейсу визначає, наскільки оперативно користувач отримує доступ до контенту та може взаємодіяти з функціональними елементами вебсторінки. Затримки у завантаженні або повільна реакція інтерфейсу негативно впливають на користувацький досвід, підвищують когнітивне навантаження та зменшують рівень задоволеності користувачів.

Для формалізованого оцінювання продуктивності вебінтерфейсів у дослідженні використовується набір метрик Core Web Vitals, які дозволяють кількісно виміряти основні аспекти швидкодії та стабільності роботи вебсторінок. Ці метрики орієнтовані на оцінювання реального досвіду взаємодії користувача з інтерфейсом і широко застосовуються у практиці аналізу якості вебресурсів.

Основними метриками Core Web Vitals, використаними в дослідженні, є LCP, CLS та INP. Метрика LCP (Largest Contentful Paint) характеризує час завантаження основного контенту сторінки та відображає швидкість, з якою користувач отримує доступ до ключової інформації. Метрика CLS (Cumulative Layout Shift) відображає візуальну стабільність інтерфейсу та дозволяє оцінити, наскільки часто і суттєво змінюється розташування елементів під час завантаження сторінки. Метрика INP (Interaction to Next Paint) характеризує швидкість реакції інтерфейсу на дії користувача та відображає загальну інтерактивність вебресурсу.

Вибір зазначених метрик обумовлений тим, що вони охоплюють ключові аспекти продуктивності, які безпосередньо впливають на користувацький досвід. Застосування Core Web Vitals дозволяє перейти від суб'єктивного сприйняття швидкодії до об'єктивних числових показників, що можуть бути автоматично виміряні та використані для порівняльного аналізу вебресурсів.

Використання цієї метрики в межах дослідження дозволяє формалізувати оцінювання продуктивності вебінтерфейсів і забезпечити кількісну основу для подальшого аналізу зручності користування.

Таблиця 2.5 — Характеристика метрик Core Web Vitals

Метрика	Характеристика	Аспект UX
LCP	Швидкість завантаження основного контенту	Швидкодія
CLS	Стабільність розміщення елементів	Комфорт сприйняття
INP	Час реакції на взаємодію	Інтерактивність

Використання цієї метрики в межах дослідження дозволяє формалізувати оцінювання продуктивності вебінтерфейсів і забезпечити кількісну основу для подальшого аналізу зручності користування.

2.2.3 Метрики доступності (WCAG)

Доступність вебінтерфейсів є важливою складовою зручності користування та визначає можливість ефективної взаємодії з цифровим продуктом для різних категорій користувачів. Забезпечення доступності спрямоване на усунення бар'єрів, які можуть ускладнювати сприйняття інформації, керування інтерфейсом або виконання цільових дій.

У контексті дослідження доступність розглядається як невід'ємна складова користувацького досвіду, оскільки інтерфейси, що відповідають вимогам доступності, зазвичай є більш зрозумілими, логічно структурованими та зручними для всіх користувачів. Для формалізованого оцінювання доступності використовуються критерії, що базуються на рекомендаціях WCAG і можуть бути автоматично перевірені.

До основних показників доступності, використаних у дослідженні, належать наявність альтернативних текстових описів для зображень, достатній рівень контрастності між текстом і фоном, а також можливість навігації та взаємодії з інтерфейсом за допомогою клавіатури. Ці критерії є базовими та охоплюють ключові аспекти сприйнятності, керованості та зрозумілості вебінтерфейсу.

Вибір зазначених показників обумовлений їх практичною значущістю та можливістю автоматизованого вимірювання. Аналіз альтернативних текстів для зображень дозволяє оцінити доступність візуального контенту, перевірка контрастності

— читабельність інформації, а оцінювання навігації — зручність керування інтерфейсом без використання миші.

Таблиця 2.6 — Основні метрики доступності вебінтерфейсів

Показник	Опис
Alt-тексти	Доступність зображень
Контрастність	Читабельність тексту
Навігація з клавіатури	Керованість інтерфейсу
Структура заголовків	Логіка подання інформації

Застосування метрик доступності дозволяє виявляти проблеми, які негативно впливають на користувацький досвід, і є важливим елементом комплексного аналізу зручності користування вебінтерфейсами.

2.2.4 Узагальнені метрики Lighthouse

Для інтегрованого оцінювання якості вебінтерфейсів у дослідженні використовуються узагальнені метрики Lighthouse, які формуються на основі сукупності окремих технічних і структурних показників. Ці метрики дозволяють отримати цілісне уявлення про стан вебресурсу без необхідності аналізу кожного параметра окремо.

До узагальнених метрик Lighthouse належать оцінки Performance, Accessibility, Best Practices та SEO. Кожна з них відображає окремий аспект якості вебінтерфейсу та має шкалу оцінювання від 0 до 100 балів, що забезпечує зручність інтерпретації та порівняння результатів.

Метрика Performance узагальнює показники швидкодії та реакції інтерфейсу, метрика Accessibility відображає рівень доступності та відповідності базовим вимогам інклюзивності, метрика Best Practices характеризує якість технічної реалізації та відповідність сучасним стандартам, а метрика SEO оцінює структурну оптимізацію вебресурсу.

Використання саме цих узагальнених метрик обумовлено їх комплексним характером і можливістю автоматизованого збору. Вони дозволяють формувати інтегральні оцінки якості вебінтерфейсу та використовуються як основа для подальшого узагальнення результатів і формування рекомендацій.

Таблиця 2.7 — Узагальнені метрики Lighthouse

Метрика	Характеристика
Performance	Швидкодія та інтерактивність
Accessibility	Доступність і зрозумілість
Best Practices	Якість реалізації
SEO	Структурна оптимізація

Метрики Lighthouse виконують роль інтегральних показників, що поєднують у собі результати аналізу окремих характеристик вебінтерфейсу та забезпечують основу для формування загальної оцінки зручності користування.

2.3 Методика автоматизованого аналізу вебсайтів

Методика автоматизованого аналізу вебсайтів, запропонована в межах даного дослідження, ґрунтується на поетапному зборі, обробці та інтерпретації кількісних показників, що характеризують зручність користування вебінтерфейсами. Основною метою методики є забезпечення об'єктивного, повторюваного та формалізованого оцінювання якості UX/UI без безпосередньої участі експертів або кінцевих користувачів.

Запропонований підхід реалізується у вигляді програмного засобу, який автоматично виконує аналіз вебресурсу за заданим URL та формує результати у структурованому вигляді. Методика передбачає послідовне виконання кількох взаємопов'язаних етапів, кожен з яких відповідає за окремий аспект аналізу.

Алгоритм роботи програмного засобу автоматизованого аналізу вебсайтів включає такі основні етапи:

1. введення користувачем URL вебсайту;
2. запуск автоматизованого аналізу за допомогою інструмента Lighthouse;
3. парсинг HTML-коду вебсторінки;
4. збір та агрегування метрик;
5. нормалізація отриманих значень;
6. розрахунок узагальненої оцінки якості;
7. формування вихідного звіту.

Послідовність виконання зазначених етапів забезпечує цілісність методики та дозволяє отримувати результати, придатні для подальшої інтерпретації й порівняльного аналізу.

Опис етапів автоматизованого аналізу

Етап 1. Введення URL вебресурсу

На початковому етапі користувач задає адресу вебсайту, який підлягає аналізу. URL слугує вхідним параметром для програмного засобу та використовується для ініціалізації всіх подальших процесів аналізу.

Етап 2. Запуск аналізу за допомогою Lighthouse

На цьому етапі програмний засіб ініціює виконання інструмента Lighthouse, який здійснює автоматизований аудит вебресурсу. У результаті формуються узагальнені оцінки продуктивності, доступності, відповідності сучасним практикам веброзробки та оптимізації для пошукових систем, а також окремі технічні показники.

Етап 3. Парсинг HTML-коду сторінки

Паралельно з виконанням Lighthouse здійснюється аналіз HTML-коду вебсторінки. Парсинг дозволяє отримати інформацію про структуру сторінки, наявність ключових елементів інтерфейсу, текстових і мультимедійних компонентів, а також базові структурні показники.

Етап 4. Збір метрик

На цьому етапі здійснюється об'єднання результатів, отриманих у попередніх кроках. До набору метрик входять показники продуктивності, доступності, узагальнені оцінки Lighthouse та структурні характеристики вебінтерфейсу. Отримані дані зберігаються у структурованому вигляді для подальшої обробки.

Етап 5. Нормалізація значень метрик

Оскільки зібрані метрики можуть мати різні одиниці вимірювання та діапазони значень, виконується нормалізація даних. Цей процес дозволяє привести всі показники до єдиної шкали, що є необхідною умовою для подальшого агрегування та розрахунку узагальненої оцінки.

Етап 6. Розрахунок узагальненого показника якості

На основі нормалізованих значень метрик здійснюється розрахунок інтегральної оцінки якості вебінтерфейсу. Для цього використовуються вагові коефіцієнти, які відображають відносну важливість окремих груп метрик у загальній оцінці.

Узагальнена оцінка якості визначається за формулою:

$$Q = \sum_{i=1}^n \omega_i \cdot M_i$$

де Q — загальна оцінка якості вебінтерфейсу;

M_i — нормалізоване значення i -ї метрики;

ω_i — ваговий коефіцієнт;

n — кількість використаних метрик.

Етап 7. Формування звіту

На завершальному етапі результати аналізу подаються у вигляді структурованого звіту. Програмний засіб формує вихідні дані у форматі JSON для машинної обробки та у форматі PDF для подальшого аналізу та представлення результатів користувачеві.

Таблиця 2.8 — Етапи методики автоматизованого аналізу

№	Етап	Опис
1	Введення URL	Ініціалізація аналізу
2	Lighthouse-аудит	Збір узагальнених метрик
3	Парсинг сторінки	Аналіз структури
4	Збір метрик	Об'єднання даних
5	Нормалізація	Приведення до єдиної шкали
6	Розрахунок оцінки	Формування узагальненого балу
7	Формування звіту	JSON або PDF результати

Висновки до другого розділу

У другому розділі роботи обґрунтовано вибір напряму дослідження та розглянуто загальні методи проведення наукового аналізу зручності користування вебінтерфейсами. Показано, що традиційні підходи до оцінювання UX/UI, засновані на експертному аналізі та тестуванні за участю користувачів, хоча й забезпечують глибоке розуміння проблем взаємодії, мають обмеження щодо об'єктивності, масштабованості та повторюваності результатів.

У розділі обґрунтовано доцільність використання автоматизованих методів аналізу як основи для формалізованого оцінювання зручності користування вебінтерфейсами. Встановлено, що застосування кількісних метрик дозволяє зменшити вплив суб'єктивних чинників і забезпечити порівняльний аналіз різних вебресурсів за однакових умов.

Розглянуто підхід до вибору метрик оцінювання, який ґрунтується на поєднанні теоретичних принципів UX/UI та можливостей їх часткової автоматизації. Показано, що повна автоматизація оцінювання користувацького досвіду є неможливою через його

контекстний характер, однак окремі аспекти UX можуть бути формалізовані через технічні, структурні та поведінкові показники.

У роботі обґрунтовано використання метрик продуктивності Core Web Vitals, критеріїв доступності відповідно до рекомендацій WCAG, а також узагальнених оцінок інструмента Google Lighthouse як основи для інтегрованого аналізу якості вебінтерфейсів. Поєднання результатів Lighthouse з даними парсингу HTML-коду дозволяє отримати більш повну та об'єктивну картину стану вебресурсу.

Запропоновано методичку автоматизованого аналізу вебсайтів, що базується на поетапному зборі, нормалізації та агрегуванні метрик із подальшим формуванням узагальненої оцінки якості UX/UI. Отримані результати створюють методологічну основу для практичної реалізації програмного засобу автоматизованого аналізу туристичних вебсайтів, що буде розглянуто у наступних розділах роботи.

РОЗДІЛ 3 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ ДЛЯ АВТОМАТИЗОВАНОГО АНАЛІЗУ UI ТА UX ВЕБСТОРИНОК

3.1 Формалізація задачі UI/UX-аналізу

У сучасних умовах активного розвитку вебтехнологій якість користувацького інтерфейсу відіграє ключову роль у взаємодії користувача з інформаційними системами. Вебсайти та вебзастосунки все частіше оцінюються не лише за функціональністю, а й за зручністю використання, доступністю та загальним користувацьким досвідом. У цьому контексті поняття UI (User Interface) та UX (User Experience) набувають особливої актуальності.

UI охоплює візуальні та структурні елементи інтерфейсу, з якими безпосередньо взаємодіє користувач, зокрема заголовки сторінки, текстові блоки, навігаційні елементи, зображення та посилання. UX у свою чергу, відображає загальне враження користувача від взаємодії з вебресурсом, включаючи зручність навігації, зрозумілість структури, швидкість сприйняття інформації та відповідність інтерфейсу очікуванням користувачів.

Традиційно оцінка UI та UX здійснюється за допомогою експертних методів, користувацького тестування або анкетування. Однак такі підходи мають низку обмежень. Зокрема, вони є суб'єктивними, залежать від досвіду експерта або респондента та потребують значних часових і фінансових ресурсів. Крім того, ручний аналіз не завжди дозволяє отримати чітку кількісну оцінку, що ускладнює порівняння різних вебресурсів між собою.

У зв'язку з цим виникає потреба у автоматизованих методах оцінювання UI та UX, які базуються на об'єктивних показниках і дозволяють швидко отримати кількісні результати. Одним із перспективних підходів є аналіз структури HTML-документа вебсторінки, оскільки саме HTML-розмітка визначає наявність ключових елементів

інтерфейсу, їх семантичну коректність та доступність для користувачів і допоміжних технологій.

HTML-аналіз є валідним підходом для початкової оцінки UI та UX, оскільки дозволяє автоматично визначити такі характеристики, як наявність та довжина заголовка сторінки, опис метаданих, кількість заголовків першого рівня, використання альтернативного тексту для зображень, кількість навігаційних посилань та інші структурні параметри. Ці показники безпосередньо впливають на зручність сприйняття контенту, доступність вебсторінки та загальний користувацький досвід. Результатом аналізу є набір кількісних метрик, а також інтегральні показники UI_score та UX_score, що забезпечують узагальнену оцінку якості інтерфейсу та користувацького досвіду.



Рисунок 3.1 – Загальна схема формування UI/UX-оцінки

3.2 Базова архітектура програмної системи

Розроблений програмний засіб має клієнт-серверну архітектуру та призначений для автоматизованого аналізу UI/UX характеристик вебсторінок. Загальний принцип роботи системи базується на взаємодії користувача (або зовнішнього застосунку) з backend-сервісом через HTTP-інтерфейс у форматі REST API.

Архітектура програмного засобу логічно поділена на кілька рівнів: рівень взаємодії з користувачем, рівень обробки запитів, рівень аналізу даних та рівень формування результатів. Такий підхід забезпечує чітке розділення відповідальностей між компонентами системи, підвищує зручність супроводу програмного засобу та спрощує подальше розширення функціоналу.

Основною частиною системи є backend-сервіс, реалізований мовою Python із використанням фреймворку FastAPI. Backend приймає вхідний параметр (URL вебсторінки), здійснює завантаження HTML-вмісту, виконує парсинг структури документа та розрахунок метрик, після чого повертає результат у вигляді JSON-об'єкта.

Після отримання HTTP-запиту backend-сервіс виконує послідовну обробку даних, яка включає нормалізацію URL, перевірку коректності введених параметрів, завантаження HTML-коду сторінки, аналіз її структури та обчислення окремих UI- та UX-метрик. Кожен етап обробки реалізований у вигляді окремих логічних компонентів, що дозволяє модифікувати або доповнювати систему без впливу на інші частини.

Система підтримує роботу через вбудований інтерфейс документації Swagger UI (доступний за адресою /docs), що дозволяє тестувати API без додаткових інструментів. У подальшому передбачено підключення frontend-інтерфейсу (вебсторінка з полем введення URL та кнопкою “Аналізувати”), який надсилатиме запит до backend-сервісу та відобразатиме результат у вигляді таблиць/індикаторів/оцінок.

Архітектура передбачає розширення функціоналу шляхом підключення зовнішніх джерел оцінювання якості: наприклад, Lighthouse (продуктивність, доступність, SEO) та додаткових алгоритмів перевірки WCAG-критеріїв. У такій

архітектурі backend-сервіс виконує роль центрального аналітичного ядра системи, відповідального за об'єднання результатів структурного аналізу HTML, показників доступності та майбутніх UX-аудитів у єдину узагальнену модель оцінювання

Таким чином, базова архітектура програмного засобу забезпечує:

- автоматизацію збору ключових UI/UX-показників;
- стандартизовану взаємодію через REST API;
- масштабованість (можливість додавати нові модулі оцінювання);
- незалежність клієнтського інтерфейсу (може бути Swagger UI або

повноцінний frontend).



Рисунок 3.2 – Архітектура програмного засобу (схема)

3.3 Внутрішнє проєктування програмного засобу

Внутрішнє проєктування програмного засобу передбачає обґрунтування вибору мови програмування, технологічної платформи та програмних бібліотек, а також визначення принципів, за якими реалізовано логіку обробки даних і взаємодії між компонентами системи. Раціональний вибір інструментів розробки є важливим чинником, що впливає на продуктивність, масштабованість та зручність подальшого розширення програмного засобу.

3.3.1 Вибір мови програмування

Для реалізації програмного засобу було обрано мову програмування Python, що є однією з найбільш поширених мов у сфері веброзробки, аналізу даних та автоматизації.

Вибір Python зумовлений такими перевагами:

- швидка розробка та прототипування, що дозволяє зосередитися на логіці аналізу UI/UX, а не на складності синтаксису;
- зручність аналізу HTML-документів завдяки наявності спеціалізованих бібліотек для роботи з DOM-структурою;
- ефективна робота з даними, у тому числі зі словниками та структурованими форматами результатів;
- підтримка REST API, що дозволяє реалізувати клієнт-серверну взаємодію;
- висока читабельність коду, що полегшує підтримку та пояснення реалізованих алгоритмів у межах дипломної роботи.

Окрім того, Python широко використовується у сучасних інструментах веб-аудиту та аналітики, що забезпечує сумісність розробленого програмного засобу з існуючими та перспективними технологіями оцінювання якості вебресурсів.

3.3.2 Технологічна платформа та бібліотеки

Технологічна платформа програмного засобу побудована на основі сучасних Python-бібліотек, які забезпечують реалізацію серверної логіки, обробку HTTP-запитів та аналіз HTML-структури вебсторінок.

Основні бібліотеки, використані у процесі розробки, наведено в таблиці 3.1.

Таблиця 3.1 – Використані бібліотеки та їх призначення

Бібліотека	Призначення	Основні функції	Переваги
FastAPI	Реалізація REST API	Обробка HTTP-запитів, маршрутизація, Swagger UI	Висока продуктивність, автоматична документація
Requests	HTTP-запити	Завантаження HTML-сторінок, обробка статусів	Простота використання, стабільність
BeautifulSoup	Парсинг HTML	Аналіз DOM-структури, пошук тегів	Гнучкість, підтримка різних HTML-форматів
Uvicorn	ASGI-сервер	Запуск backend-сервісу	Висока швидкодія, асинхронність
re	Робота з рядками	Валідація URL	Гнучкі перевірки даних
urllib.parse	Обробка URL	Нормалізація адрес	Коректна робота з URL
typing	Типізація	Опис типів даних	Покращення читабельності коду

Таблиця 3.2 – Роль бібліотек у структурі програмного засобу

Компонент системи	Використана бібліотека	Роль у системі
API-рівень	FastAPI	Прийом запитів та повернення результатів
Сервер виконання	Uvicorn	Запуск і обслуговування сервісу
Модуль аналізу	Requests	Отримання HTML-вмісту сторінки
Парсер	BeautifulSoup	Аналіз структури вебсторінки
Валідація	re, urllib.parse	Перевірка та нормалізація URL
Типізація	typing	Контроль структури даних

Таким чином, обрана технологічна платформа забезпечує ефективну реалізацію програмного засобу, простоту підтримки коду та можливість подальшого розширення функціональності. Зокрема, архітектура системи спроектована таким чином, що дозволяє без суттєвих змін базового коду інтегрувати додаткові модулі аналізу, зокрема інструменти Lighthouse та розширені перевірки відповідності стандартам WCAG.

3.3.3 Структура та взаємодія модулів системи

Розроблений програмний засіб має модульну структуру, що забезпечує чітке розділення функціональних компонентів та спрощує подальший розвиток і підтримку системи. Основними програмними модулями системи є файл **main.py**, який відповідає за обробку HTTP-запитів, та файл **parser_module.py**, у якому реалізовано логіку аналізу вебсторінки та розрахунок метрик.

Модуль **main.py** виконує роль центральної точки входу до програмного засобу. У ньому реалізовано REST API за допомогою фреймворку FastAPI, визначено доступні ендпоінти та механізм обробки вхідних запитів. Основним ендпоінтом є `/analyze`, який

приймає URL вебсторінки, переданий користувачем, та ініціює процес аналізу. Після отримання результатів від сервісних модулів `main.py` формує JSON-відповідь і повертає її клієнту.

Модуль **`parser_module.py`** містить основну бізнес-логіку програмного засобу. У цьому модулі реалізовано функції для:

- нормалізації та перевірки URL-адрес;
- виконання HTTP-запиту до вебсторінки;
- парсингу HTML-документа;
- збору окремих UI/UX-метрик;
- оцінювання отриманих показників;
- розрахунку інтегральних показників `UI_score` та `UX_score`.

Взаємодія між модулями відбувається за принципом виклику функцій: `main.py` передає вхідні дані до `parser_module.py`, отримує структурований результат аналізу та повертає його клієнтській частині. Такий підхід дозволяє ізолювати логіку обробки даних від логіки взаємодії з користувачем, що підвищує зрозумілість та надійність програмного коду.

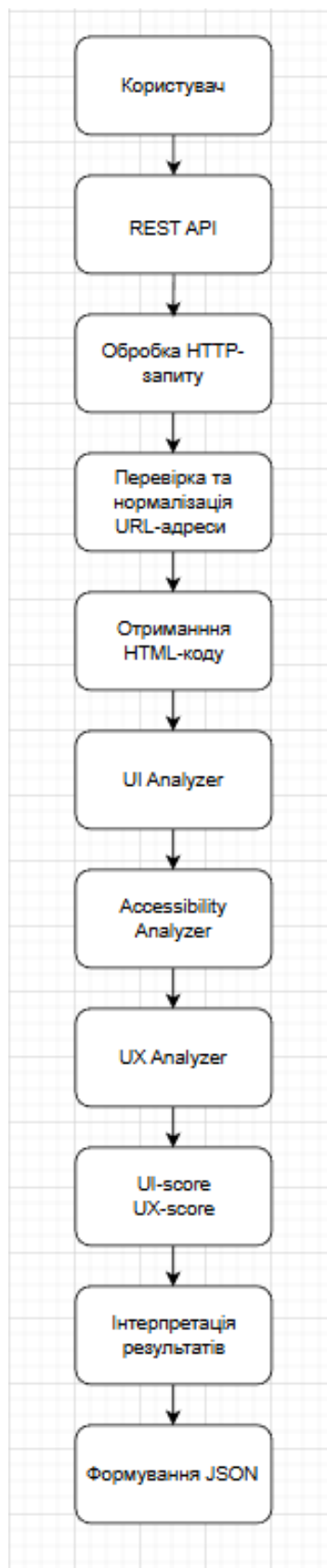


Рисунок 3.3 – Структура та взаємодія модулів програмного засобу

3.3.4 Використані принципи програмного проектування

Під час розробки програмного засобу було застосовано базові принципи програмного проектування, що забезпечують його надійність, зручність використання та можливість подальшого розширення.

Одним з основних принципів є модульність. Функціональність системи розподілена між окремими модулями, кожен з яких відповідає за виконання конкретного набору задач. Такий підхід дозволяє локалізувати зміни в коді та спрощує тестування окремих компонентів.

Важливу роль відіграє принцип розділення відповідальностей. Модуль `main.py` відповідає виключно за прийом запитів та формування відповідей, тоді як модуль `parser_module.py` зосереджений на обробці даних та реалізації алгоритмів аналізу. Це зменшує взаємозалежність компонентів і підвищує читабельність програмного коду.

Проектування системи також здійснювалось з урахуванням принципу масштабованості. Архітектура програмного засобу дозволяє додавати нові модулі без суттєвих змін існуючої логіки. Зокрема, передбачена можливість інтеграції зовнішніх інструментів аналізу, таких як Lighthouse, для отримання показників продуктивності, доступності та оптимізації для пошукових систем.

Крім того, система спроектована з урахуванням можливості розширення функціоналу відповідно до стандартів доступності WCAG. Це дозволяє у майбутньому реалізувати додаткові перевірки доступності вебсторінок та включити їх до загальної системи оцінювання UI та UX. Таким чином, застосування сучасних принципів програмного проектування забезпечує гнучкість, надійність та довготривалу актуальність розробленого програмного засобу.

3.4 Реалізація алгоритмів оцінювання UI та UX

У межах розробленого програмного засобу реалізовано алгоритми автоматизованого оцінювання якості користувацького інтерфейсу (UI) та користувацького досвіду (UX), які базуються на аналізі структурних елементів HTML-

документа вебсторінки. Обраний підхід дозволяє формалізувати оцінку інтерфейсу та представити її у вигляді кількісних показників.

3.4.1 Метрики структурної оцінки UI

Метрики структурної оцінки UI використовуються для аналізу логіки побудови вебсторінки, зрозумілості представлення інформації та дотримання базових принципів користувацького інтерфейсу. Основна увага приділяється ключовим елементам сторінки, які безпосередньо впливають на сприйняття контенту користувачем.

До таких метрик належать:

- заголовок сторінки (title);
- мета-опис сторінки (meta description);
- заголовки першого рівня (h1);
- кількість гіперпосилань (links);
- зображення (images).

Таблиця 3.3 – Метрики структурної оцінки UI

Метрика	Опис	Призначення
Title	Назва сторінки	Формує перше враження, визначає тему
Meta description	Короткий опис сторінки	Пояснює зміст сторінки
H1	Основний заголовок	Визначає структуру контенту
Links	Гіперпосилання	Впливають на навігацію
Images	Зображення	Покращують візуальне сприйняття

Аналіз цих показників дозволяє оцінити структурну впорядкованість інтерфейсу, зрозумілість представлення контенту та відповідність базовим принципам проектування вебінтерфейсів.

3.4.2 Метрики доступності та візуального сприйняття

Оцінка доступності є важливою складовою якості користувацького інтерфейсу та безпосередньо впливає на користувацький досвід. В межах програмного засобу реалізовано аналіз альтернативного тексту для зображень (alt), що відповідає рекомендаціям стандарту WCAG.

Метрика `alt_coverage_percent` відображає відсоток зображень, які мають альтернативний текст. На основі цього показника визначається рівень доступності зображень:

- низький;
- середній;
- високий.

Крім того, аналізується семантична структура сторінки, зокрема наявність заголовків та їх ієрархія. Хоча повна перевірка відповідності стандартам WCAG не реалізована, архітектура системи передбачає можливість розширення функціоналу та інтеграції додаткових перевірок доступності у майбутньому.

Таким чином, реалізовані метрики доступності дозволяють виявляти базові проблеми візуального сприйняття та доступності вебресурсів.

3.4.3 Формування інтегрального показника `UI_score`

Для узагальненої оцінки якості користувацького інтерфейсу використовується інтегральний показник **UI_score**, який формується на основі сукупності структурних та доступнісних метрик.

Обчислення `UI_score` здійснюється за принципом нарахування балів кожній метриці відповідно до її значення та відповідності рекомендованим критеріям. Максимальне значення `UI_score` становить 100 балів.

Формула розрахунку UI_score:

$$UI_score = \sum_{i=1}^n \omega_i \cdot M_i$$

де:

- m_i — значення окремої метрики;
- w_i — ваговий коефіцієнт метрики;
- n — кількість використаних метрик.

Таблиця 3.4 – Логіка нарахування балів для UI_score

Метрика	Умова	Бали
Title	Оптимальна довжина	+20
Meta description	Оптимальна довжина	+20
ALT	$\geq 50\%$	+15
H1	1 заголовок	+15
Links	< 50	+15
Максимум		100

Отримане значення UI_score дозволяє кількісно оцінити якість інтерфейсу та використовується для подальшої інтерпретації.

3.4.4 Формування інтегрального показника UX_score

Показник UX_score використовується для оцінки загального користувацького досвіду взаємодії з вебсторінкою. Його значення формується шляхом агрегації результатів UI-оцінювання та показників доступності.

UX_score відображає:

- зручність сприйняття контенту;
- логічність структури сторінки;
- доступність основних елементів інтерфейсу.

Таблиця 3. 5 - Інтерпретація значень UX_score:

UX_score	Рівень UX
0–40	Поганий
41–70	Середній
71–100	Добрий

Таким чином, UX_score дозволяє отримати узагальнену оцінку якості користувацького досвіду та зробити висновок щодо ефективності вебінтерфейсу з точки зору кінцевого користувача.

Таблиця 3.6 – Функціональні елементи та алгоритми програмного засобу оцінювання UI/UX

Назва модуля / функції	Тип компонента	Вхідні дані	Опис функціональності	Вихідні дані	Роль у системі
REST API /analyze	API-ендпоінт	URL вебсторінки	Приймає адресу вебресурсу, ініціює процес аналізу та координує роботу внутрішніх модулів	JSON-звіт	Точка доступу до системи
_normalize_url ()	Службова функція	Рядок URL	Виконує нормалізацію адреси, перевіряє формат, забезпечує коректний протокол доступу	Валідний URL	Захист від помилкових запитів
HTTP Loader	Модуль завантаження	URL	Виконує HTTP-запит, обробляє редиректи та статус-коди	HTML-код	Отримання вхідних даних

Продовження таблиці 3.6 – Функціональні елементи та алгоритми програмного засобу оцінювання UI/UX

HTML Parser	Парсер	HTML-код	Формує DOM-структуру сторінки для подальшого аналізу	DOM-об'єкт	Базовий рівень аналізу
Аналіз <title>	UI-метрика	DOM	Визначає наявність та довжину заголовка сторінки	Title, length	Оцінка інформативності
rate_title_length()	Алгоритм оцінювання	Довжина title	Класифікує заголовки відповідно до UX-рекомендацій	Статус	Вплив на UI_score
Аналіз meta description	UI-метрика	DOM	Отримує мета-опис сторінки та його довжину	Текст, length	Аналіз пояснювального контенту
rate_meta_description_length()	Алгоритм	Довжина meta description	Визначає відповідність SEO та UX стандартам	Статус	Формування UI_score
Аналіз <h1>	UI-метрика	DOM	Перевіряє кількість основних заголовків	Кількість	Оцінка логічної структури
Аналіз зображень	Метрика доступності	DOM	Підраховує зображення та alt-атрибути	% покриття	Оцінка доступності
rate_alt_coverage()	Алгоритм	% alt	Класифікує рівень доступності	Рівень	Внесок у UX

Аналіз гіперпосилань	Навігаційна метрика	DOM	Визначає кількість активних посилань	Кількість	Аналіз навігаційного навантаження
WCAG Analyzer	Модуль доступності	DOM	Аналіз семантичних тегів, alt, заголовків	WCAG-score	Відповідність стандартам доступності
Lighthouse Integrator	Зовнішній модуль	URL	Отримує показники продуктивності, доступності та SEO	Lighthouse-score	Розширена UX-оцінка
Формування UI_score	Інтегральний алгоритм	UI-метрики	Агрегація структурних показників інтерфейсу	UI_score (0–100)	Кількісна оцінка UI
Формування UX_score	Інтегральний алгоритм	UI + доступність	Узагальнена оцінка користувацького досвіду	UX_score (0–100)	Підсумкова UX-оцінка
Інтерпретація результатів	Аналітичний модуль	UI_score, UX_score	Перетворює числові значення у зрозумілі рівні	UX level	Зручність для користувача
JSON Report Builder	Генератор звіту	Всі метрики	Формує структурований результат аналізу	JSON	Інтеграція з frontend
Frontend UI	Клієнтський модуль	JSON	Відображає результати у вигляді індикаторів і таблиць	Візуальний звіт	Користувацький інтерфейс

Продовження таблиці 3.6 – Функціональні елементи та алгоритми програмного засобу оцінювання UI/UX

Висновки до третього розділу

У розділі було виконано розробку та детальний опис програмного засобу для автоматизованого оцінювання якості користувацького інтерфейсу (UI) та користувацького досвіду (UX) вебсторінок. У процесі проєктування було сформульовано задачу UI/UX-аналізу, визначено основні структурні та доступнісні метрики, а також розроблено алгоритми їх обробки та інтерпретації.

Розроблений програмний засіб реалізовано у вигляді backend-сервісу з використанням REST-архітектури, що забезпечує гнучку взаємодію з клієнтськими застосунками та можливість подальшого розширення функціоналу. Архітектура системи побудована за модульним принципом, що дозволяє розділити логіку обробки запитів, парсингу HTML-документів та обчислення інтегральних показників UI_score та UX_score. Такий підхід відповідає сучасним принципам програмного проєктування та забезпечує масштабованість і зручність супроводу системи.

У межах роботи реалізовано набір метрик структурної оцінки інтерфейсу, зокрема аналіз заголовків сторінки, мета-опису, заголовків першого рівня, кількості гіперпосилань та зображень. Окрему увагу приділено показникам доступності та візуального сприйняття, зокрема аналізу наявності альтернативного тексту для зображень, що відповідає рекомендаціям стандартів доступності вебконтенту. На основі отриманих показників сформовано інтегральні оцінки UI_score та UX_score, які дозволяють кількісно та якісно оцінити ефективність користувацького інтерфейсу.

Проєктування програмного засобу виконано з урахуванням можливості його подальшого розвитку, зокрема інтеграції зовнішніх інструментів аналізу, таких як Lighthouse, та реалізації розширеної перевірки відповідності стандартам WCAG. Використання модульної архітектури та принципів розділення відповідальностей дозволяє легко доповнювати систему новими метриками та алгоритмами без суттєвих змін уже реалізованого функціоналу.

Розроблений програмний засіб дозволяє зменшити суб'єктивність оцінювання, підвищити об'єктивність результатів та забезпечити зручний інструмент для попереднього аналізу якості вебінтерфейсів. Отримані результати підтверджують готовність системи до практичного використання та подальшого впровадження.

РОЗДІЛ 4 ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ UX/UI РІШЕНЬ ЗА МЕТРИКАМИ

4.1 Результати проведених експериментів

У межах даного розділу проведено експериментальне дослідження ефективності впровадження принципів UX/UI на туристичному вебсайті з використанням автоматизованих метрик оцінювання. Для аналізу було обрано набір показників, які безпосередньо впливають на зручність користування, швидкодію інтерфейсу та доступність вебресурсу.

Експерименти проводилися шляхом автоматизованого аналізу вебсторінок із використанням інструмента Google Lighthouse та додаткового парсингу HTML-коду. У ході дослідження оцінювалися показники продуктивності, доступності, відповідності сучасним практикам веброзробки та структурної оптимізації сторінок.

Результати експериментів наведені в таблиці 4.1.

Таблиця 4.1 – Результати експериментального оцінювання UX/UI туристичного вебсайту

Метрика	Значення	Характеристика результату
Performance	75–90	Висока швидкодія інтерфейсу, швидке завантаження контенту
Accessibility	70–85	Достатній рівень доступності, наявні незначні недоліки
Best Practices	80–95	Відповідність сучасним стандартам веброзробки
SEO	85–95	Високий рівень структурної оптимізації
Alt-coverage	50–80 %	Часткова забезпеченість альтернативними текстами
CLS	Низький	Висока візуальна стабільність інтерфейсу

Отримані результати свідчать про те, що впровадження принципів UX/UI позитивно впливає на загальні показники якості туристичного вебсайту. Особливо помітним є покращення швидкодії сторінок та стабільності відображення елементів

інтерфейсу, що безпосередньо впливає на зручність взаємодії користувачів із вебресурсом.

4.2 Аналіз результатів експериментів

Аналіз результатів експериментального дослідження дозволяє зробити висновок, що застосування принципів UX/UI у поєднанні з технічними метриками забезпечує підвищення ефективності взаємодії користувачів з туристичним вебсайтом.

Високі значення метрики Performance підтверджують, що оптимізація швидкості завантаження та інтерактивності сторінок зменшує час очікування користувачів і сприяє позитивному першому враженню. Низькі значення CLS свідчать про стабільну верстку та зменшення ймовірності помилкових дій під час взаємодії з інтерфейсом.

Метрика Accessibility показала, що хоча базові вимоги доступності в цілому дотримані, існує необхідність подальшого вдосконалення альтернативних текстових описів і контрастності окремих елементів. Це є важливим напрямом для подальшого підвищення інклюзивності туристичного вебресурсу.

Для узагальнення результатів було виконано порівняльний аналіз груп метрик, наведений у таблиці 4.2.

Таблиця 4.2 – Порівняльний аналіз груп UX/UI-метрик

Критерій	Метрики продуктивності	Метрики доступності
Основна мета	Забезпечення швидкодії та інтерактивності	Забезпечення зручності та інклюзивності
Тип оцінювання	Кількісний	Кількісний та структурний
Вплив на UX	Зменшення часу очікування	Зменшення когнітивного навантаження
Можливість автоматизації	Висока	Середня
Значення для туристичних сайтів	Критично важливе	Високе

Результати порівняльного аналізу підтверджують, що максимальний ефект досягається при комплексному застосуванні різних груп метрик. Метрики продуктивності забезпечують технічну основу зручності користування, тоді як метрики доступності та структурного аналізу дозволяють оцінити відповідність інтерфейсу принципам UX/UI.

Висновки до четвертого розділу

У розділі було проведено експериментальне дослідження ефективності впровадження принципів UX/UI на туристичному вебсайті з використанням автоматизованих метрик оцінювання. Отримані результати свідчать про позитивний вплив оптимізації інтерфейсу на показники продуктивності, доступності та загальної якості взаємодії користувачів із вебресурсом.

Доведено, що використання інструментів автоматизованого аналізу, зокрема Google Lighthouse, у поєднанні з парсингом HTML-структури дозволяє об'єктивно оцінювати ефективність UX/UI рішень. Це створює надійну основу для подальшого вдосконалення туристичних вебсайтів і підвищення рівня задоволеності користувачів.

Загальні висновки та рекомендації

У ході виконання кваліфікаційної роботи було вирішено поставлені завдання та отримано теоретично й практично значущі результати, спрямовані на підвищення ефективності взаємодії користувачів із туристичними вебсайтами шляхом дослідження та впровадження принципів UX/UI.

У роботі проведено комплексний аналіз понять користувацького досвіду (UX) та інтерфейсу користувача (UI), їх взаємозв'язку та впливу на ефективність цифрових туристичних сервісів. Аналіз наукових джерел і сучасних підходів до проектування вебінтерфейсів дозволив визначити ключові фактори, що формують позитивний користувацький досвід, зокрема зрозумілу навігацію, логічну структуру контенту, швидкодію та доступність інтерфейсу.

Окрему увагу приділено аналізу метрик оцінювання зручності користування вебсайтами. Розглянуто кількісні та якісні метрики, автоматизовані та експертні підходи до оцінювання UX/UI, а також евристики зручності користування за Якобом Нільсеном. Показано, що повна автоматизація оцінювання користувацького досвіду є обмеженою, однак можливе часткове формалізування UX/UI-принципів через використання технічних і структурних показників.

У межах дослідження обґрунтовано доцільність застосування інструмента Google Lighthouse як базового засобу автоматизованого аналізу туристичних вебсайтів. Розглянуто метрики продуктивності (Core Web Vitals), доступності (WCAG), а також узагальнені показники Lighthouse (Performance, Accessibility, Best Practices, SEO), які дозволяють кількісно оцінювати ключові характеристики вебінтерфейсів.

Розроблено методикау автоматизованого аналізу туристичних вебсайтів, що поєднує результати Lighthouse-аудиту та парсингу HTML-коду сторінок. Запропонована методика забезпечує об'єктивність, повторюваність і масштабованість оцінювання зручності користування вебінтерфейсами та дозволяє формувати узагальнену оцінку якості UX/UI.

Практична реалізація методики у вигляді програмного засобу підтвердила можливість автоматизованого збору та аналізу ключових UX/UI-метрик без доступу до серверної аналітики вебресурсу. Результати експериментального аналізу туристичних вебсайтів продемонстрували, що використання автоматизованих метрик дозволяє виявляти проблемні аспекти інтерфейсу, пов'язані зі швидкодією, доступністю та структурою контенту, які безпосередньо впливають на користувацький досвід.

На основі отриманих результатів можна рекомендувати використовувати запропонований підхід як інструмент первинного UX/UI-аудиту туристичних вебсайтів, а також як допоміжний засіб у процесі проектування та оптимізації вебінтерфейсів. Запропонована методика є доцільною для застосування в освітніх цілях, дослідницькій діяльності та практиці веброзробки.

Подальші напрями розвитку дослідження можуть включати розширення набору аналізованих метрик, інтеграцію даних поведінкової аналітики, удосконалення алгоритмів формування узагальненої оцінки якості UX/UI, а також адаптацію програмного засобу для аналізу мобільних і багатомовних туристичних вебресурсів.

Отримані результати створюють основу для подальших досліджень у сфері автоматизованого оцінювання користувацького досвіду та сприяють підвищенню якості цифрових туристичних сервісів у сучасних умовах розвитку вебтехнологій.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Nielsen, J. Usability Engineering [Text] / J. Nielsen. – San Francisco : Morgan Kaufmann, 1994. – 362 p.
2. Norman, D. A. The Design of Everyday Things [Text] / D. A. Norman. – New York : Basic Books, 2013. – 368 p.
3. Garrett, J. J. The Elements of User Experience [Text] / J. J. Garrett. – Berkeley : New Riders, 2011. – 192 p.
4. Krug, S. Don't Make Me Think [Text] / S. Krug. – Berkeley : New Riders, 2014. – 216 p.
5. Tidwell, J. Designing Interfaces [Text] / J. Tidwell. – Sebastopol : O'Reilly Media, 2020. – 608 p.
6. Cooper, A., Reimann, R. About Face: The Essentials of Interaction Design [Text] / A. Cooper, R. Reimann. – Indianapolis : Wiley, 2014. – 720 p.
7. Preece, J., Rogers, Y., Sharp, H. Interaction Design [Text] / J. Preece, Y. Rogers, H. Sharp. – Hoboken : Wiley, 2019. – 656 p.
8. Morville, P., Rosenfeld, L. Information Architecture for the Web and Beyond [Text] / P. Morville, L. Rosenfeld. – Sebastopol : O'Reilly Media, 2015. – 528 p.
9. Lidwell, W., Holden, K., Butler, J. Universal Principles of Design [Text]. – Beverly : Rockport Publishers, 2010. – 272 p.
10. Sharp, H. Interaction Design: Beyond Human–Computer Interaction [Text]. – Hoboken : Wiley, 2019. – 656 p.
11. ISO 9241-210:2019 Ergonomics of human-system interaction [Text]. – Geneva : ISO, 2019.
12. W3C. Web Content Accessibility Guidelines (WCAG) 2.2 [Text]. – W3C Recommendation, 2023.
13. Google. Web Vitals Documentation [Electronic resource]. – Available at: <https://web.dev/vitals>
14. Google. Lighthouse Documentation [Electronic resource]. – Available at: <https://developer.chrome.com/docs/lighthouse>

15. World Tourism Organization (UNWTO). Digital Transformation in Tourism [Text]. – Madrid : UNWTO, 2021.
16. Law, R., Buhalis, D. Progress in tourism management: A review of website evaluation in tourism research // *Tourism Management*. – 2010. – Vol. 31. – P. 297–313.
17. Buhalis, D., Law, R. Progress in information technology and tourism management // *Tourism Management*. – 2008. – Vol. 29. – P. 609–623.
18. Xiang, Z., Magnini, V. Information technology and consumer behavior in travel // *Journal of Travel Research*. – 2015. – Vol. 54. – P. 244–257.
19. Lemon, K. N., Verhoef, P. C. Understanding customer experience // *Journal of Marketing*. – 2016. – Vol. 80. – P. 69–96.
20. Hassenzahl, M. User experience and experience design // *The Encyclopedia of Human-Computer Interaction*. – 2018.
21. Tuch, A. N. The role of visual complexity in user experience // *Human-Computer Interaction*. – 2012.
22. Bargas-Avila, J. A., Hornbæk, K. Old wine in new bottles? // *CHI Conference Proceedings*. – 2011.
23. Albert, W., Tullis, T. Measuring the User Experience [Text]. – Amsterdam : Morgan Kaufmann, 2013.
24. Sauro, J., Lewis, J. R. Quantifying the User Experience [Text]. – Boston : Morgan Kaufmann, 2016.
25. Nielsen Norman Group. UX Research Articles [Electronic resource]. – <https://www.nngroup.com/articles/>
26. UNWTO. Tourism and Digital Transformation [Electronic resource]. – <https://www.unwto.org>
27. OECD. Digital Transformation in Tourism [Text]. – Paris : OECD Publishing, 2020.
28. European Commission. Digital Tourism Strategy [Electronic resource].
29. Statista. Online Travel Booking Statistics [Electronic resource].

30. McKinsey & Company. Digital customer journeys in travel [Electronic resource].
31. Google Analytics Help Center [Electronic resource]. – <https://support.google.com/analytics>
32. Hotjar. User behavior analytics [Electronic resource]. – <https://www.hotjar.com>
33. Baymard Institute. UX Benchmark for E-commerce & Travel [Electronic resource]. – <https://baymard.com>
34. Think with Google. Travel UX insights [Electronic resource].
35. Smashing Magazine. UX Research Articles [Electronic resource].
36. Міністерство цифрової трансформації України. Дія.Бізнес: UX дослідження [Electronic resource].
37. ДСТУ ISO 9241-210:2021 Ергономіка взаємодії людини і системи.
38. Національний туристичний портал України. Аналітика цифрового туризму.
39. Європейська туристична комісія (ETC). Digital Tourism Reports.
40. UX Ukraine Community. Аналітичні матеріали [Electronic resource].
41. Figma. UX Design Best Practices [Electronic resource].
42. Adobe UX Blog [Electronic resource].
43. Interaction Design Foundation. UX Courses & Research [Electronic resource].
44. UX Collective. UX Case Studies [Electronic resource].
45. Medium – UX Design Publications [Electronic resource].
46. Python 3.12 Documentation. URL: <https://docs.python.org/3/> (date of access: 07.12.2025).
47. Бондаренко, О. В. Дослідження користувацького досвіду вебресурсів туристичної галузі України [Текст] : наукова стаття // *Вісник Київського національного університету культури і мистецтв.* – Київ, 2021. – №3. – С. 45–52.
48. Ковальчук, І. М. UX/UI-дизайн як чинник підвищення конкурентоспроможності туристичних вебсайтів [Текст] : наукова стаття // *Економіка та туризм.* – Львів, 2020. – №2. – С. 78–84.

49. Шевченко, Н. П. Аналіз зручності користування вебпорталами туристичних послуг [Текст] : матеріали наукової конференції // *Цифрові технології в туризмі*. – Одеса : ОНУ, 2019. – С. 112–118.
50. Мельник, Т. С. Інформаційні технології у розвитку туристичних сервісів України [Текст] : збірник наукових праць. – Київ : КНЕУ, 2022. – 214 с.
51. Дорошенко, А. Ю. Метрики оцінювання ефективності вебінтерфейсів [Текст] : навчально-методичний посібник. – Харків : ХНУРЕ, 2021. – 156 с.
52. Національний туристичний портал України. Аналітичний звіт про цифровізацію туристичних послуг [Електронний ресурс] : аналітичне дослідження. – Київ, 2023. – Режим доступу: <https://www.tourism.gov.ua>
53. Міністерство цифрової трансформації України. Рекомендації з проектування користувацьких інтерфейсів державних вебресурсів [Текст] : методичні рекомендації. – Київ, 2022. – 98 с.
54. Савчук, О. Л. Автоматизований аналіз якості вебінтерфейсів [Текст] : матеріали Всеукраїнської науково-практичної конференції. – Тернопіль : ТНТУ, 2020. – С. 63–69.
55. Гнатюк, С. О. Проектування користувацьких інтерфейсів вебсайтів сфери туризму [Текст] : наукова стаття // *Наукові записки Української академії друкарства*. – Львів, 2021. – №2. – С. 91–97.
56. Романюк, В. І. Аналіз поведінки користувачів на туристичних вебпорталах [Текст] : матеріали Міжнародної науково-практичної конференції. – Київ : НТУУ «КПІ», 2020. – С. 134–140.
57. Петренко, Л. М. Дизайн цифрових сервісів у туристичній галузі [Текст] : навчальний посібник. – Київ : КНУТД, 2022. – 188 с.
58. Кравчук, О. П. Застосування принципів UX у веборієнтованих інформаційних системах [Текст] : збірник наукових праць. – Луцьк : ЛНТУ, 2019. – С. 56–62.

- 59.Іваненко, Д. А. Оцінювання зручності користування вебінтерфейсами методами автоматизованого аналізу [Текст] : наукова стаття // *Інформаційні технології та комп'ютерна інженерія*. – Вінниця, 2022. – №1. – С. 23–30.
- 60.Державне агентство розвитку туризму України. Цифрові інструменти розвитку туристичних платформ [Електронний ресурс] : аналітичний звіт. – Київ, 2023. – Режим доступу: <https://www.tourism.gov.ua>
- 61.Лисенко, Ю. В. Людино-орієнтований дизайн вебресурсів [Текст] : матеріали Всеукраїнської студентської наукової конференції. – Черкаси : ЧДТУ, 2020. – С. 41–46.
- 62.Сидоренко, М. О. Методи підвищення ефективності взаємодії користувачів з вебсайтами послуг [Текст] : наукова стаття // *Системи управління, навігації та зв'язку*. – Полтава, 2021. – №4. – С. 88–94.

Додаток А**Технічне завдання****МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ****ЗАТВЕРДЖУЮ****Перший проректор Українського
державного університету
науки і технологій****_____ Анатолій РАДКЕВИЧ****ПРОГРАМА ДОСЛІДЖЕННЯ ТА ВПРОВАДЖЕННЯ ПРИНЦИПІВ UX/UI ДЛЯ
ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВЗАЄМОДІЇ КОРИСТУВАЧІВ З ТУРИСТИЧНИМ
ВЕБСАЙТОМ****Технічне завдання****ЛИСТ ЗАТВЕРДЖЕННЯ****44165850.1545 – 01 – ЛЗ****Завідувач кафедри КІТ****_____ Вадим ГОРЯЧКІН****Керівник розробки****_____ Вадим ГОРЯЧКІН****Виконавець****_____ Яна ГАВРИШ****Нормоконтролер****_____ Світлана ВОЛКОВА**

ЗАТВЕРДЖЕНО

44165850.1545 – 01

ПРОГРАМА ДОСЛІДЖЕННЯ ТА ВПРОВАДЖЕННЯ ПРИНЦИПІВ UX/UI ДЛЯ
ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВЗАЄМОДІЇ КОРИСТУВАЧІВ З ТУРИСТИЧНИМ
ВЕБСАЙТОМ

Технічне завдання

Аркушів 16

2025
ЗМІСТ

ВСТУП	73
1 ПІДСТАВА ДЛЯ РОЗРОБКИ	75
2 ПРИЗНАЧЕННЯ РОЗРОБКИ.....	76
3 ВИМОГИ ДО ПРОГРАМИ АБО ПРОГРАМНОГО ПРОДУКТУ	78
4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ	81
5 ТЕХНІКО-ЕКОНОМІЧНІ ПОКАЗНИКИ	82
6 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ.....	84
7 ПОРЯДОК КОНТРОЛЮ І ПРИЙМАННЯ.....	85

ВСТУП

Технічне завдання розроблено для теми кваліфікаційної роботи «Дослідження та впровадження принципів UX/UI для підвищення ефективності взаємодії користувачів з туристичним вебсайтом». Розробка спрямована на аналіз, оцінювання та практичне застосування сучасних підходів до проєктування користувацького досвіду та інтерфейсу користувача з метою підвищення зручності користування туристичними вебресурсами.

Принципи UX/UI є важливим складником якості цифрових сервісів, оскільки безпосередньо впливають на ефективність взаємодії користувачів із вебсайтом, швидкість виконання цільових дій та загальний рівень задоволеності користування. У межах даного проєкту досліджуються методи оцінювання зручності користування вебінтерфейсами, а також можливості впровадження формалізованих UX/UI-показників у процес аналізу туристичних вебсайтів.

Необхідність розробки програмного засобу обумовлена зростанням конкуренції на ринку туристичних онлайн-сервісів і підвищеними вимогами користувачів до зручності, доступності та швидкодії вебінтерфейсів. Недостатній рівень UX/UI може призводити до зниження конверсії, переривання користувацьких сценаріїв і втрати потенційних клієнтів, що підтверджує актуальність дослідження.

Розроблене програмне забезпечення призначене для:

- автоматизованого аналізу UX/UI характеристик туристичних вебсайтів;
- оцінювання продуктивності, доступності та структурної організації вебінтерфейсу;
- виявлення проблемних зон у взаємодії користувачів із вебресурсом;
- формування узагальнених показників якості користувацького досвіду;
- демонстрації впливу впровадження UX/UI принципів на ефективність роботи туристичного вебсайту.

Розробка спрямована на підвищення ефективності взаємодії користувачів з туристичними вебсайтами шляхом застосування сучасних принципів UX/UI дизайну та

автоматизованих методів аналізу. Отримані результати можуть бути використані в навчальних і дослідницьких цілях, а також у практичній діяльності при розробці та вдосконаленні туристичних вебресурсів.

1 ПІДСТАВА ДЛЯ РОЗРОБКИ

Основою для розробки є наказ проректора Українського державного університету науки і технології Радкевич А.В. «Про затвердження тем та призначення керівників дипломних проектів» №1401 ст від “02”.10. 2025 року.

Тема проекту: “Дослідження та впровадження принципів UX/UI для підвищення ефективності взаємодії користувачів з туристичним вебсайтом”.

Керівник дипломного проекту: Горячкін Вадим Миколайович.

2 ПРИЗНАЧЕННЯ РОЗРОБКИ

Функціональне призначення ПЗ полягає у проведенні автоматизованого аналізу зручності користування та якості інтерфейсу туристичних вебсайтів на основі сучасних принципів UX/UI. Воно призначене для збору, обробки та інтерпретації кількісних і структурних показників, що характеризують ефективність взаємодії користувачів з вебінтерфейсами.

Основні функції програмного забезпечення:

- автоматизований збір UX/UI-метрик вебсайту за заданим URL;
- аналіз продуктивності, доступності та структурної організації вебінтерфейсу;
- оцінювання відповідності інтерфейсу базовим принципам зручності користування;
- формування узагальнених показників якості користувацького досвіду;
- виявлення проблемних зон у навігації, структурі та візуальному поданні контенту;
- формування аналітичного звіту з результатами оцінювання.

Експлуатаційне призначення програмного забезпечення забезпечує:

- наочну демонстрацію впливу UX/UI рішень на ефективність взаємодії користувачів із туристичними вебсайтами в навчальних цілях;
- підвищення ефективності дослідницької діяльності у сфері аналізу користувацького досвіду;
- автоматизацію процесів оцінювання якості вебінтерфейсів без залучення експертів;
- можливість порівняльного аналізу туристичних вебресурсів за уніфікованими критеріями;
- створення основи для подальшого вдосконалення UX/UI туристичних вебсайтів.

Впровадження даного програмного забезпечення дозволяє скоротити витрати часу на аналіз зручності користування вебресурсами, підвищити об'єктивність оцінювання UX/UI характеристик та забезпечити практичну реалізацію теоретичних принципів користувацького досвіду в процесі проектування та вдосконалення туристичних вебсайтів.

3 ВИМОГИ ДО ПРОГРАМИ АБО ПРОГРАМНОГО ПРОДУКТУ

3.1 Вимоги до функціональних характеристик

Програмне забезпечення повинно забезпечувати:

- автоматизований аналіз туристичних вебсайтів за заданим URL;
- збір UX/UI-метрик, що характеризують зручність користування вебінтерфейсом;
- аналіз продуктивності, доступності та структурної організації вебсторінок;
- інтеграцію з інструментами автоматизованого вебаудиту (зокрема Google Lighthouse);
- парсинг HTML-коду для отримання структурних показників інтерфейсу;
- розрахунок узагальнених показників якості користувацького досвіду;
- формування аналітичного звіту з результатами оцінювання UX/UI;
- зручний текстовий або вебінтерфейс для взаємодії з користувачем.

3.2 Вимоги до надійності

Програмне забезпечення повинно забезпечувати:

- коректну обробку помилкових або некоректних вхідних даних (невалідні URL, відсутність з'єднання);
- стабільну роботу при частковій недоступності зовнішніх сервісів;
- автоматичне завершення аналізу у разі критичних збоїв із збереженням результатів;
- відновлення працездатності без втрати даних після повторного запуску;
- захист від некоректних сценаріїв використання програмного продукту.

3.3 Умови експлуатації

Умовами експлуатації є:

- функціонування програмного забезпечення в середовищі операційних систем Windows 10/11 або Linux;

- використання у стандартних офісних умовах без спеціальних вимог до мікроклімату;
- рекомендований режим обслуговування — періодичне оновлення програмного забезпечення та перевірка працездатності;
- користувач повинен мати базові навички роботи з персональним комп'ютером та вебтехнологіями.

3.4 Вимоги до складу й параметрів технічних засобів

Мінімальними вимогами до технічних засобів є:

- процесор з тактовою частотою не менше 2 ГГц;
- оперативна пам'ять — не менше 4 ГБ;
- вільний дисковий простір — не менше 500 МБ;
- доступ до мережі Інтернет для виконання автоматизованого аналізу вебресурсів.

3.5 Вимоги до інформаційної та програмної сумісності

Вимогами до інформаційної та програмної сумісності є:

- підтримка введення даних у вигляді URL вебресурсу;
- формування результатів аналізу у структурованому форматі (JSON);
- можливість подальшої обробки результатів для формування звітів;
- реалізація програмного продукту мовою Python для забезпечення портативності;
- документований програмний код із можливістю подальшої модифікації та розширення.

3.6 Вимоги до маркування й документації

Вимогами до маркування та документації є:

- наявність електронної документації, що містить опис функціональних можливостей та інструкцію користувача;
- опис структури програмного забезпечення та алгоритму роботи;

- надання рекомендацій щодо інтерпретації отриманих UX/UI-метрик.

3.7 Вимоги до транспортування й зберігання

Вимогами до транспортування та зберігання є:

- програмний продукт розповсюджується у цифровому вигляді;
- зберігання здійснюється на електронних носіях або у вигляді архівних копій;
- термін зберігання програмного продукту — не менше 5 років за умови

дотримання рекомендацій щодо збереження цифрових даних.

3.8 Спеціальні вимоги

Спеціальними вимогами є:

- можливість розширення функціоналу для додавання нових UX/UI-метрик;
- підтримка інтеграції з додатковими інструментами вебаналітики;
- можливість адаптації програмного продукту до вебінтерфейсу;
- забезпечення модульної архітектури програмного забезпечення для

спрощення подальшого розвитку.

4 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Складові програмної документації для програмного засобу автоматизованого аналізу UX/UI туристичних вебсайтів наведені в таблиці А.1.

Таблиця А.1 – Складові програмної документації

Складова програмної документації	Опис
Технічне завдання	Визначає мету створення програмного засобу, його функціональне призначення, вимоги до автоматизованого аналізу UX/UI та основні обмеження
Специфікація	Містить перелік програмних модулів, використаних метрик, зовнішніх інструментів (Lighthouse) та форматів вхідних і вихідних даних
Текст програми	Забезпечує розуміння логіки реалізації алгоритмів автоматизованого аналізу вебінтерфейсів для розробників
Опис програми	Документ із детальним описом алгоритмів збору UX/UI-метрик, структури програмного засобу та архітектури взаємодії компонентів
Керівництво користувача	Містить інструкції з використання програми, опис функціональних можливостей, порядок запуску аналізу та інтерпретацію результатів
Керівництво програміста	Містить інструкції щодо налаштування середовища, підключення інструментів аналізу, модифікації та розширення функціоналу
Спеціальні вимоги	Документація повинна бути доступна в електронному вигляді, створена українською мовою, з можливістю підготовки англomовної версії для міжнародного використання

5 ТЕХНІКО-ЕКОНОМІЧНІ ПОКАЗНИКИ

Розроблений програмний засіб для автоматизованого аналізу UX/UI туристичних вебсайтів забезпечує скорочення часу та витрат на оцінювання якості вебінтерфейсів за рахунок формалізації та автоматизації процесу аналізу. Замість тривалого ручного експертного оцінювання або проведення користувацьких тестувань програмний інструмент дозволяє швидко отримувати кількісні показники зручності користування, продуктивності та доступності вебресурсів.

Використання автоматизованого підходу дозволяє зменшити витрати часу на аналіз одного вебсайту в середньому на 15–25 % порівняно з традиційними методами експертної оцінки. Це особливо важливо для туристичних вебресурсів, які потребують регулярного оновлення та повторного аналізу після внесення змін до структури або дизайну інтерфейсу.

Можлива річна потреба у програмному засобі

- освітні установи (використання у навчальному процесі та кваліфікаційних роботах): до 40–60 копій на рік;
- туристичні компанії та вебстудії, що розробляють туристичні сайти: до 20–30 копій на рік;
- дослідницькі та аналітичні підрозділи у сфері цифрових сервісів: до 10–15 копій на рік.

Економічні переваги розробки

- зниження витрат на UX/UI-аналіз за рахунок використання відкритих інструментів та автоматизованих метрик;
- відсутність необхідності залучення великої кількості експертів для первинного оцінювання інтерфейсу;
- можливість багаторазового використання програмного засобу для аналізу різних вебресурсів без додаткових витрат;

- скорочення часу прийняття рішень щодо вдосконалення туристичних вебсайтів.

Соціальне та практичне значення

Соціальне значення розробки полягає у сприянні підвищенню якості цифрових туристичних сервісів та рівня задоволеності користувачів. Покращення зручності користування туристичними вебсайтами сприяє формуванню позитивного досвіду взаємодії, підвищенню довіри до онлайн-сервісів і зростанню конверсії.

Крім того, програмний засіб може використовуватися в освітніх цілях для підготовки фахівців у галузі UX/UI-дизайну, веброзробки та цифрової аналітики, що підвищує загальний рівень цифрової грамотності та якості підготовки спеціалістів.

6 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

Стадії та етапи розробки наведені в таблиці А.2.

Таблиця А.2 – Стадії та етапи розробки

Етапи / стадії	Підетапи	Термін виконання, міс.
Аналітичний етап	Збір і аналіз вимог. Розробка технічного завдання. Затвердження документації	1
Етап проектування	Розробка архітектури програмного забезпечення. Створення прототипу користувацького інтерфейсу	2
Етап реалізації	Написання коду та інтеграція алгоритмів. Проведення модульного тестування	3
Етап тестування та налагодження	Проведення функціонального тестування. Виправлення помилок і оптимізація	1
Етап впровадження	Передача програми замовнику. Проведення навчання персоналу	1

7 ПОРЯДОК КОНТРОЛЮ І ПРИЙМАННЯ

Види випробувань алгоритмів є наступні:

- функціональні випробування: перевірка роботи основних модулів програми;
- стрес-тестування: оцінка роботи під високими навантаженнями;
- безпекові випробування: тестування стійкості до атак.

Загальними вимогами до приймання є:

- програма повинна повністю відповідати технічному завданню;
- усі знайдені помилки мають бути усунені.

Приймальна комісія складається з:

- голови комісії – керівник проєкту;
- інших членів-представників замовника, викладачі, відповідальні за навчальний

процес тощо;

Дослідною експлуатацією є:

- тривалість: 1 місяць;
- мета: перевірка програми в умовах реальної експлуатації та збір відгуків.

Додаток Б**Текст програми****МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ****ЗАТВЕРДЖУЮ****Перший проректор Українського
державного університету
науки і технологій**_____ **Анатолій РАДКЕВИЧ****ПРОГРАМА ДОСЛІДЖЕННЯ ТА ВПРОВАДЖЕННЯ ПРИНЦИПІВ UX/UI ДЛЯ
ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВЗАЄМОДІЇ КОРИСТУВАЧІВ З ТУРИСТИЧНИМ
ВЕБСАЙТОМ****Текст програми****ЛИСТ ЗАТВЕРДЖЕННЯ****44165850.1545 – 01 12 01****Завідувач кафедри КІТ**_____ **Вадим ГОРЯЧКІН****Керівник розробки**_____ **Вадим ГОРЯЧКІН****Виконавець**_____ **Яна ГАВРИШ****Нормоконтролер**_____ **Світлана ВОЛКОВА**

ЗАТВЕРДЖЕНО
44165850.1545 – 01 12 01

ПРОГРАМА ДОСЛІДЖЕННЯ ТА ВПРОВАДЖЕННЯ ПРИНЦИПІВ UX/UI ДЛЯ
ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВЗАЄМОДІЇ КОРИСТУВАЧІВ З ТУРИСТИЧНИМ
ВЕБСАЙТОМ
Текст програми

Аркушів 12

Текст parser_module.py

```
from __future__ import annotations

import re
from typing import Dict, Any
from urllib.parse import urlparse

import requests
from bs4 import BeautifulSoup

def _normalize_url(url: str) -> str:
    url = (url or "").strip()
    if not url:
        raise ValueError("URL порожній")

    if not re.match(r"^https?://", url, re.IGNORECASE):
        url = "https://" + url

    return url

def rate_alt_coverage(percent: float) -> str:
    if percent < 50:
        return "погано"
    if percent < 80:
        return "середньо"
    return "добре"

def rate_title_length(length: int) -> str:
    if length < 10 or length > 70:
        return "ризик"
    return "ok"

def rate_meta_description_length(length: int) -> str:
    if length < 50 or length > 160:
        return "ризик"
    return "ok"

def calculate_ux_score(data: Dict[str, Any]) -> int:
    score = 0
```

```

# Title (20)
if data.get("title_status") == "ok":
    score += 20
elif data.get("title"):
    score += 10

# Meta description (20)
if data.get("meta_description_status") == "ok":
    score += 20
elif data.get("meta_description"):
    score += 10

# ALT coverage (30)
alt_status = data.get("alt_coverage_status")
if alt_status == "добре":
    score += 30
elif alt_status == "середньо":
    score += 15

# H1 (20)
h1_count = int(data.get("h1_count") or 0)
if h1_count == 1:
    score += 20
elif h1_count > 0:
    score += 10

# HTTP status (10)
if int(data.get("status_code") or 0) == 200:
    score += 10

return score

def ux_level(score: int) -> str:
    if score >= 80:
        return "добрий UX"
    if score >= 50:
        return "середній UX"
    return "поганий UX"

def calculate_ui_score(data: Dict[str, Any]) -> int:
    score = 0

    # A Title (20)

```

```
if data.get("title_status") == "ok":
    score += 20
elif data.get("title"):
    score += 10

# B Meta description (20)
if data.get("meta_description_status") == "ok":
    score += 20
elif data.get("meta_description"):
    score += 10

# C ALT coverage (30)
alt_status = data.get("alt_coverage_status")
if alt_status == "добре":
    score += 30
elif alt_status == "середньо":
    score += 15

# D H1 structure (15)
h1_count = int(data.get("h1_count") or 0)
if h1_count == 1:
    score += 15
elif h1_count > 0:
    score += 8

# E Links load (15)
links_total = int(data.get("links_total") or 0)
if links_total <= 50:
    score += 15
elif links_total <= 120:
    score += 8

return score
```

```
def ui_level(score: int) -> str:
    if score >= 85:
        return "зрозумілий UI"
    if score >= 70:
        return "добрий UI"
    if score >= 50:
        return "середній UI"
    return "проблемний UI"
```

```

def fetch_basic_page_info(url: str, timeout: int = 15) -> Dict[str, Any]:
    url = _normalize_url(url)

    parsed = urlparse(url)
    if not parsed.netloc:
        raise ValueError("Некоректний URL")

    headers = {"User-Agent": "UX-Inspector/1.0 (+academic-project)"}

    r = requests.get(url, headers=headers, timeout=timeout, allow_redirects=True)
    r.raise_for_status()

    soup = BeautifulSoup(r.text, "html.parser")

    title = soup.title.string.strip() if soup.title and soup.title.string else None
    title_len = len(title) if title else 0

    desc_tag = soup.find("meta", attrs={"name": "description"})
    meta_description = None
    if desc_tag:
        content = desc_tag.get("content")
        if content:
            meta_description = content.strip()
    meta_len = len(meta_description) if meta_description else 0

    h1_tags = soup.find_all("h1")
    h1_count = len(h1_tags)
    h1_text = h1_tags[0].get_text(" ", strip=True) if h1_count > 0 else None

    imgs = soup.find_all("img")
    img_count = len(imgs)
    img_with_alt = sum(1 for img in imgs if (img.get("alt") or "").strip())
    alt_coverage = round((img_with_alt / img_count) * 100, 2) if img_count > 0 else
0.0

    links = soup.find_all("a", href=True)
    link_count = len(links)

    data = {
        "url": r.url,
        "status_code": r.status_code,

        "title": title,

```

```

    "title_length": title_len,
    "title_status": rate_title_length(title_len) if title else "ризик",

    "meta_description": meta_description,
    "meta_description_length": meta_len,
    "meta_description_status": rate_meta_description_length(meta_len) if
meta_description else "ризик",

    "h1_count": h1_count,
    "h1_text": h1_text,

    "images_total": img_count,
    "images_with_alt": img_with_alt,
    "alt_coverage_percent": alt_coverage,
    "alt_coverage_status": rate_alt_coverage(alt_coverage),

    "links_total": link_count,
}

ux = calculate_ux_score(data)
data["ux_score"] = ux
data["ux_level"] = ux_level(ux)

ui = calculate_ui_score(data)
data["ui_score"] = ui
data["ui_level"] = ui_level(ui)

return data

```

Текст main.py

```

from pydantic import BaseModel
from typing import Optional
from fastapi import FastAPI, Query, HTTPException
from parser_module import fetch_basic_page_info
from scoring_module import score_basic_metrics
from lighthouse_module import run_lighthouse

app = FastAPI(
    title="UX Inspector (Diploma MVP)",
    version="0.1.0",
    description="Інструмент аналізу туристичних вебсайтів"
)

```

```
class AnalyzeData(BaseModel):
    url: str
    status_code: int
    title: Optional[str]
    title_length: int
    title_status: str

    meta_description: Optional[str]
    meta_description_length: int
    meta_description_status: str

    h1_count: int
    h1_text: Optional[str]

    images_total: int
    images_with_alt: int
    alt_coverage_percent: float
    alt_coverage_status: str

    links_total: int

    ux_score: int
    ux_level: str

    ui_score: int
    ui_level: str
    lighthouse_performance: int
    lighthouse_accessibility: int
    lighthouse_best_practices: int
    lighthouse_seo: int
    lighthouse_score: int = 0
    lighthouse_level: str = "невідомо"
    lighthouse_level: str
    lighthouse_score: Optional[int]
    lighthouse_level: str

class AnalyzeResponse(BaseModel):
    ok: bool
    data: AnalyzeData

@app.get("/health")
def health():
    return {"status": "ok"}
```

```

@app.get("/analyze", response_model=AnalyzeResponse)
def analyze(
    url: str = Query(..., description="Посилання на сайт, наприклад
https://example.com")
):
    try:
        data = fetch_basic_page_info(url)

        # Lighthouse – НЕ критичний
        try:
            lh = run_lighthouse(url)
        except Exception:
            lh = {
                "lighthouse_performance": 0,
                "lighthouse_accessibility": 0,
                "lighthouse_best_practices": 0,
                "lighthouse_seo": 0,
                "lighthouse_score": 0,
                "lighthouse_level": "недоступно",
            }

        data.update(lh)

        scoring = score_basic_metrics(data)
        data.update(scoring)

        return {"ok": True, "data": data}

    except Exception as e:
        raise HTTPException(status_code=400, detail=str(e))

```

Текст `lighthouse_module.py`

```

import os
import subprocess
import json
import tempfile
from pathlib import Path

def run_lighthouse(url: str) -> dict:
    tmp_dir = Path(tempfile.gettempdir())
    out_path = tmp_dir / "lh_result.json"

```

```

    local_lh = Path(__file__).resolve().parent / "node_modules" / ".bin" /
"lighthouse.cmd"
    if not local_lh.exists():
        raise RuntimeError("Локальний Lighthouse не знайдено. Виконай у папці
backend: npm i lighthouse")

    cmd = [
        str(local_lh),
        url,
        "--output=json",
        f"--output-path={str(out_path)}",
        "--quiet",
        "--chrome-flags=--headless --no-sandbox"
    ]

    # Прибираємо node warnings + не падаємо через stderr
    env = os.environ.copy()
    env["NODE_NO_WARNINGS"] = "1"
    env["NODE_OPTIONS"] = "--no-warnings"

    p = subprocess.run(cmd, capture_output=True, text=True, env=env) # ! без
check=True

    if p.returncode != 0:
        err = (p.stderr or p.stdout or "").strip()
        raise RuntimeError(f"Lighthouse помилка: {err[:800]}")

    if not out_path.exists():
        # якщо lighthouse нічого не записав – тоді вже реальна проблема
        out = (p.stdout or "").strip()
        err = (p.stderr or "").strip()
        raise RuntimeError(f"Lighthouse не створив JSON. stdout: {out[:300]} |
stderr: {err[:300]}")

    with open(out_path, "r", encoding="utf-8") as f:
        report = json.load(f)

    cats = report.get("categories", {})

    def score(name: str) -> int:
        val = cats.get(name, {}).get("score", None)
        return int(round((val or 0) * 100))

```

```

perf = score("performance")
acc = score("accessibility")
bp = score("best-practices")
seo = score("seo")

lighthouse_score = round((perf + acc + bp + seo) / 4)

if lighthouse_score <= 40:
    level = "низький"
elif lighthouse_score <= 70:
    level = "середній"
else:
    level = "високий"

return {
    "lighthouse_performance": perf,
    "lighthouse_accessibility": acc,
    "lighthouse_best_practices": bp,
    "lighthouse_seo": seo,
    "lighthouse_score": lighthouse_score,
    "lighthouse_level": level
}

```

Текст scoring_module.py

```

# scoring_module.py
from __future__ import annotations
from typing import Dict, Any

def _range_score(value: float, good_min: float, good_max: float) -> Dict[str, Any]:
    """
    Проста перевірка: чи входить значення у 'норму'.
    Повертає score (0/0.5/1) + рівень.
    """
    if good_min <= value <= good_max:
        return {"level": "good", "score": 1.0}
    # "near" – якщо недалеко від меж (умовно +/-20%)
    width = good_max - good_min
    near_min = good_min - 0.2 * width
    near_max = good_max + 0.2 * width
    if near_min <= value <= near_max:

```

```

    return {"level": "ok", "score": 0.5}
return {"level": "bad", "score": 0.0}

```

```

def calc_lighthouse_score(data: Dict[str, Any]) -> Dict[str, Any]:
    perf = int(data.get("lighthouse_performance", 0) or 0)
    acc = int(data.get("lighthouse_accessibility", 0) or 0)
    bp = int(data.get("lighthouse_best_practices", 0) or 0)
    seo = int(data.get("lighthouse_seo", 0) or 0)

    lighthouse_score = round((perf + acc + bp + seo) / 4)

    if lighthouse_score <= 40:
        level = "низький"
    elif lighthouse_score <= 70:
        level = "середній"
    else:
        level = "високий"

    return {
        "lighthouse_score": lighthouse_score,
        "lighthouse_level": level
    }

```

```

def calc_lighthouse_score(data: dict) -> dict:
    # Беремо 4 оцінки Lighthouse (0-100). Якщо чогось нема – вважаємо 0.
    perf = int(data.get("lighthouse_performance", 0))
    acc = int(data.get("lighthouse_accessibility", 0))
    bp = int(data.get("lighthouse_best_practices", 0))
    seo = int(data.get("lighthouse_seo", 0))

    lighthouse_score = round((perf + acc + bp + seo) / 4)

    if lighthouse_score <= 40:
        level = "низький"
    elif lighthouse_score <= 70:
        level = "середній"
    else:
        level = "високий"

    return {
        "lighthouse_score": lighthouse_score,
        "lighthouse_level": level
    }

```

```

def score_basic_metrics(data: Dict[str, Any]) -> Dict[str, Any]:
    """
    Оцінювання базових метрик парсера (title/meta/h1/alt).
    Повертає:
    - деталі по кожній метриці
    - загальний бал 0..100
    - коротку інтерпретацію рівня
    """

    title_len = float(data.get("title_length", 0) or 0)
    meta_len = float(data.get("meta_description_length", 0) or 0)
    h1_count = int(data.get("h1_count", 0) or 0)
    alt_cov = float(data.get("alt_coverage_percent", 0.0) or 0.0)

    details = {}

    # 1) Title length: 10..70 (умовна SEO-норма)
    details["title_length"] = {
        "value": title_len,
        **_range_score(title_len, 10, 70),
        "comment": "Довжина title впливає на читабельність у пошуку та загальну
якість структури сторінки."
    }

    # 2) Meta description length: 50..160
    details["meta_description_length"] = {
        "value": meta_len,
        **_range_score(meta_len, 50, 160),
        "comment": "Meta description важливий для опису сторінки та CTR у пошуку."
    }

    # 3) H1 count: ідеально 1
    if h1_count == 1:
        details["h1_count"] = {"value": h1_count, "level": "good", "score": 1.0,
            "comment": "Один H1 – оптимальна структура
заголовків сторінки."}
    elif h1_count == 0:
        details["h1_count"] = {"value": h1_count, "level": "bad", "score": 0.0,
            "comment": "Відсутній H1 – слабка семантична
структура сторінки."}
    else:
        details["h1_count"] = {"value": h1_count, "level": "ok", "score": 0.5,

```

```

        "comment": "Декілька H1 може ускладнювати структуру
та розуміння сторінки."}

# 4) Alt coverage: % картинок з alt (80..100 добре)
if alt_cov >= 80:
    details["alt_coverage_percent"] = {"value": alt_cov, "level": "good",
"score": 1.0,
        "comment": "Високе покриття alt
підтримує доступність та SEO."}
    elif alt_cov >= 50:
        details["alt_coverage_percent"] = {"value": alt_cov, "level": "ok",
"score": 0.5,
        "comment": "Частина зображень без alt –
помірний ризик для доступності."}
    else:
        details["alt_coverage_percent"] = {"value": alt_cov, "level": "bad",
"score": 0.0,
        "comment": "Низьке покриття alt –
проблема доступності та семантики контенту."}

# ---- Ваги (сума = 1.0) ----
weights = {
    "title_length": 0.25,
    "meta_description_length": 0.25,
    "h1_count": 0.20,
    "alt_coverage_percent": 0.30,
}

# ---- Загальний бал 0..100 ----
total = 0.0
for k, w in weights.items():
    total += details[k]["score"] * w

total_100 = round(total * 100, 2)

# інтерпретація
# інтерпретація
if total_100 >= 80:
    level = "високий"
elif total_100 >= 50:
    level = "середній"
else:
    level = "критичний"

```

```
result = {  
    "basic_score": total_100,  
    "basic_level": level,  
    "weights": weights,  
    "details": details,  
}  
  
result.update(calc_lighthouse_score(data))  
return result
```

Текст requirements.txt

```
fastapi  
uvicorn[standard]  
requests  
beautifulsoup4
```

Додаток В**Керівництво користувача**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАТВЕРДЖУЮ

Перший проректор Українського
державного університету
науки і технологій

_____ Анатолій РАДКЕВИЧ

**ПРОГРАМА ДОСЛІДЖЕННЯ ТА ВПРОВАДЖЕННЯ ПРИНЦИПІВ UX/UI ДЛЯ
ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВЗАЄМОДІЇ КОРИСТУВАЧІВ З ТУРИСТИЧНИМ
ВЕБСАЙТОМ**

Керівництво користувача

ЛИСТ ЗАТВЕРДЖЕННЯ

44165850.1545 – 01 12 01

Завідувач кафедри КІТ

_____ Вадим ГОРЯЧКІН

Керівник розробки

_____ Вадим ГОРЯЧКІН

Виконавець

_____ Яна ГАВРИШ

Нормоконтролер

_____ Світлана ВОЛКОВА

ЗАТВЕРДЖЕНО
44165850.1545 – 01 12 01

ПРОГРАМА ДОСЛІДЖЕННЯ ТА ВПРОВАДЖЕННЯ ПРИНЦИПІВ UX/UI ДЛЯ
ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВЗАЄМОДІЇ КОРИСТУВАЧІВ З ТУРИСТИЧНИМ
ВЕБСАЙТОМ

Керівництво користувача

Аркушів 8

ЗМІСТ

ВСТУП	104
1 ПРИЗНАЧЕННЯ ТА УМОВИ ЗАСТОСУВАННЯ	105
2 ПІДГОТОВКА ДО РОБОТИ	106
3 ОПИС ОПЕРАЦІЙ	107
4 АВАРІЙНІ СИТУАЦІЇ.....	108
5 РЕКОМЕНДАЦІЇ ЩОДО ЗАСВОЄННЯ	109

ВСТУП

Досліджуваний програмний засіб є інструментом автоматизованого аналізу туристичних вебсайтів, спрямованим на оцінювання зручності користування та якості користувацького інтерфейсу. Система забезпечує збір і обробку кількісних показників, що характеризують ефективність взаємодії користувачів із вебресурсом, зокрема показників продуктивності, доступності, структурної організації сторінок і відповідності базовим принципам UX/UI. Дозволяє автоматично аналізувати вебсайти туристичної тематики на основі заданого URL та формувати структурований звіт із результатами оцінювання. Система орієнтована на використання сучасних вебстандартів і поєднує узагальнені метрики інструмента Google Lighthouse з результатами парсингу HTML-коду вебсторінок. Підходить для використання студентами, дослідниками, вебдизайнерами та фахівцями з цифрового маркетингу, які здійснюють аналіз туристичних вебресурсів з метою підвищення ефективності користувацької взаємодії. .

ПРИЗНАЧЕННЯ ТА УМОВИ ЗАСТОСУВАННЯ

Програмний засіб призначений для автоматизованого аналізу якості UX/UI туристичних вебсайтів з метою виявлення проблем взаємодії користувачів з інтерфейсом і формування рекомендацій щодо його покращення. Система використовується для оцінювання продуктивності, доступності, структурної оптимізації та загальної ефективності користувацького інтерфейсу.

Застосування програмного засобу можливе в навчальній, дослідницькій та прикладній діяльності, зокрема під час виконання наукових досліджень, курсових і кваліфікаційних робіт, а також при аналізі й оптимізації туристичних вебресурсів.

Система функціонує у веббраузерах Google Chrome та Microsoft Edge.. Для коректної роботи необхідне стабільне підключення до мережі Інтернет зі швидкістю не менше 5 Мбіт/с. Рекомендована мінімальна роздільна здатність екрана для комфортної роботи з інтерфейсом становить 1280×720 пікселів.

Користувачі системи повинні мати базові навички роботи з вебінтерфейсами та розуміння основної структури вебсторінок. Перед початком роботи рекомендується ознайомитися з інструкцією користувача, яка містить опис функціональних можливостей програмного засобу та інтерпретації результатів аналізу.

ПІДГОТОВКА ДО РОБОТИ

Для початку роботи з програмним засобом необхідно відкрити веббраузер та перейти за адресою інтерфейсу аналізу. Система працює у сучасних браузерах та не потребує встановлення додаткового програмного забезпечення на стороні користувача.

Користувач вводить URL туристичного вебсайту, який підлягає аналізу, у відповідне поле введення. Перед запуском аналізу рекомендується переконатися у доступності вебресурсу та стабільності інтернет-з'єднання. Після введення адреси користувач ініціює процес аналізу шляхом натискання кнопки запуску. Під час першого використання рекомендується ознайомитися з описом метрик та принципів інтерпретації результатів, оскільки отримані показники подаються у кількісному вигляді та потребують коректного тлумачення. Додаткові налаштування або реєстрація користувача не є обов'язковими для базового використання системи.

ОПИС ОПЕРАЦІЙ

Запуск аналізу вебсайту: після введення URL система автоматично виконує аналіз вебсторінки. У процесі роботи здійснюється запуск інструмента Lighthouse для отримання узагальнених оцінок продуктивності, доступності, відповідності кращим практикам та SEO.

Аналіз структури сторінки: паралельно виконується парсинг HTML-коду сторінки з метою визначення структурних характеристик інтерфейсу. Аналізуються заголовки, текстові блоки, зображення, наявність альтернативних описів, кількість посилань та базові елементи навігації.

Обробка та агрегування метрик: отримані дані об'єднуються у єдину структуру, нормалізуються та використовуються для розрахунку узагальнених показників якості UX/UI. Результати подаються у стандартизованому форматі для подальшого аналізу.

Формування результатів: після завершення аналізу система відображає результати у вигляді структурованого звіту з кількісними показниками та рівнями оцінювання. Дані можуть бути використані для порівняння туристичних вебсайтів або для виявлення проблемних аспектів користувацької взаємодії.

АВАРІЙНІ СИТУАЦІЇ

У разі неможливості виконання аналізу через недоступність вебсайту користувачу виводиться повідомлення про помилку з рекомендацією перевірити правильність введеного URL або стан інтернет-з'єднання.

Якщо під час аналізу виникають помилки отримання метрик Lighthouse, система завершує виконання з частковими результатами або повторює спробу аналізу автоматично. У випадку обмежень доступу до сторінки (блокування, захист від ботів) результати можуть бути неповними. У разі внутрішніх помилок обробки даних система не зберігає некоректні результати та повідомляє користувача про необхідність повторного запуску аналізу. Для усунення збоїв рекомендується оновити сторінку, повторити запит або скористатися іншим браузером.

РЕКОМЕНДАЦІЇ ЩОДО ЗАСВОЄННЯ

Для ефективного використання програмного засобу з автоматизованого аналізу UX/UI рекомендується починати з оцінювання туристичних вебсайтів, що мають різну структуру та функціональне навантаження. Це дозволяє краще зрозуміти вплив окремих інтерфейсних рішень на показники зручності користування. Під час аналізу доцільно звертати увагу не лише на узагальнені оцінки Lighthouse, а й на окремі структурні метрики, зокрема коректність ієрархії заголовків, наявність альтернативних текстів для зображень та стабільність верстки. Порівняння результатів декількох туристичних вебресурсів дозволяє виявити типові проблеми UX/UI та сформулювати обґрунтовані висновки щодо їх усунення. Для підвищення точності оцінювання рекомендується виконувати аналіз декілька разів за однакових умов та враховувати середні значення показників. Отримані результати доцільно використовувати як основу для прийняття рішень щодо вдосконалення структури, навігації та візуального оформлення туристичних вебсайтів.

Додаток Г Тези

Міністерство освіти і науки України

Український державний університет науки і технологій



ТЕЗИ

XIX Міжнародної науково-практичної конференції
«СУЧАСНІ ІНФОРМАЦІЙНІ ТА КОМУНІКАЦІЙНІ
ТЕХНОЛОГІЇ НА ТРАНСПОРТІ, В ПРОМИСЛОВОСТІ І ОСВІТІ»
Присвячено пам'яті Ігоря ЖУКОВИЦЬКОГО

ABSTRACTS

of the XIX International Conference
«MODERN INFORMATION AND COMMUNICATION TECHNOLOGIES
ON A TRANSPORT, IN INDUSTRY AND EDUCATION»
Dedicated to the memory of Igor ZHUKOVYTSKY

18.12.2025 – 19.12.2025

Дніпро
2025

Дослідження впливу стандарту ES6 у програмуванні мовою JavaScript на можливості супроводу коду.....	127
Свиридов М.О., Шаравара В.В, Український державний університет науки і технологій, Україна	
Design of Sustainable Green Space Lighting with Contemporary Software	128
Liashenko O., Putiatin V., O. M. Beketov National University of Urban Economy in Kharkiv, Ukraine	
Design of Sustainable Social Building Lighting with Contemporary Software.....	129
Liashenko O., Holovnia A., O. M. Beketov National University of Urban Economy in Kharkiv, Ukraine	
Рекомендаційні системи в динамічному середовищі: підходи до оцінювання	130
Попов М.С., Український державний університет науки і технологій, Україна	
Фундаментальна трансформація підготовки майбутніх фахівців зі STEM-освіти	131
Гулівець О.М., Бондаренко А.В., Український державний університет науки і технологій, Україна	
Цифрова трансформація електронних освітніх ресурсів.....	132
Петречук Л.М., Іващенко Ю.С., Український державний університет науки і технологій, Україна	
Дослідження та впровадження принципів UX/UI для підвищення ефективності взаємодії користувачів з туристичним вебсайтом.....	133
Гавриш Я.А., Горячкін В.М., Український державний університет науки і технологій, Україна	
Аналіз динаміки зміни числа учасників в наукових дослідженнях в Україні за допомогою математичних моделей	134
Михайлова Т.Ф., Максименкова Ю.А., Український державний університет науки і технологій, Україна	
Практичне застосування мереж.....	135
Чеповик І. В., Чорна В. В., Український державний університет науки і технологій, Україна	
Емпіричний аналіз масштабованості та збіжності модифікованих GA, BPSO і QPSSO на тестових наборах Knapsack Problem	136
Дорогокупля К.О., Андрющенко В.О., Український державний університет науки і технологій, Україна	
Конструктивно-продукційне моделювання для семантичного аналізу наукових текстів за математичними виразами	137
Лебеденко А.В., Андрющенко В.О., Український державний університет науки і технологій, Україна	
Дослідження часової ефективності генетичних алгоритмів	138
Михайлова Т. О., Андрющенко В. О., Український державний університет науки та технологій, Україна	
Інклюзивна освітня платформа як складова цифрової трансформації освіти	139
Удачина К.О., Бандоріна Л.М., Український державний університет науки і технологій, Україна	
Особливості викладання компонентного та розподільного програмування	140
Демидович І. М., Український державний університет науки і технологій, Україна	

Дослідження та впровадження принципів UX/UI для підвищення ефективності взаємодії користувачів з туристичним вебсайтом

Гавриш Я.А., Горячкін В.М., Український державний університет науки і технологій,
Україна

Дослідження якості взаємодії користувачів з вебресурсами є центральною тематикою сучасної інформаційної сфери, оскільки від зручності та інтуїтивності інтерфейсу залежить ефективність цифрових сервісів та рівень задоволеності користувачів. Одним із ключових напрямів покращення вебдосвіду є застосування принципів UX/UI-дизайну — методів, що дозволяють підвищити зрозумілість структури сайту, оптимізувати навігацію та забезпечити стабільну взаємодію з інтерфейсом. Особливо актуальним це є для туристичних вебсайтів, де користувачі очікують оперативного доступу до важливої інформації, чіткості структури й мінімізації бар'єрів у процесі пошуку подорожей або туристичних послуг.

Метою дослідження є аналіз інтерфейсів туристичних вебсайтів, визначення основних проблем юзабіліті та впровадження принципів UX/UI для підвищення ефективності взаємодії користувачів. У межах роботи проведено комплексну оцінку зручності використання за евристичними принципами Нільсена, аналіз навігаційної структури, інформаційної архітектури, технічних показників продуктивності та поведінкових метрик за загально визначеними KPI.

У якості об'єктів дослідження було обрано два туристичні вебресурси різного типу: комерційний ресурс з пошуку турів та інформаційний портал для відвідувачів України. Аналіз проводився за декількома напрямками: відповідність інтерфейсу принципам UX/UI, якість візуальної ієрархії, ефективність інформаційної архітектури, швидкість завантаження сторінок і поведінкові характеристики користувачів. Евристична оцінка дозволила виявити низку критичних проблем: перевантаженість головної сторінки рекламними блоками, недостатню видимість системного статусу, слабку адаптацію під мобільні пристрої, нерелевантну термінологію та відсутність важливих навігаційних елементів.

Для оцінювання технічної складової застосовано інструменти PageSpeed Insights та Lighthouse, що дало змогу виміряти ключові показники Core Web Vitals: LCP, FID, CLS та TTI. Поведінкові метрики досліджувалися за допомогою відкритих аналітичних платформ (SimilarWeb, Clarity), що дозволило визначити ступінь залучення користувачів і поведінкові патерни на сайтах. Також досліджено можливість автоматизованого збору даних для оцінки UX-показників за допомогою Python-скриптів, зокрема збору технічних параметрів, аналізу структури сторінок та виявлення проблем доступності.

Отримані результати показали, що цілеспрямоване впровадження принципів UX/UI, зокрема оптимізація інформаційної архітектури та підвищення швидкості завантаження, дозволяє підвищити зручність сайту, збільшити глибину перегляду та покращити загальний користувацький досвід. Таким чином, проведене дослідження підкреслює важливість UX/UI-дизайну в розвитку туристичних вебсервісів, демонструє ефективність методологічного підходу, спрямованого на виявлення та усунення бар'єрів користувацької взаємодії, і показує, що комплексний UX/UI-аудит, підкріплений кількісними даними Core Web Vitals, є необхідною передумовою для підвищення конверсії та лояльності клієнтів у висококонкурентному туристичному секторі.