

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

ДЛЯ АНАЛІЗУ ВЕБСАЙТІВ

SOFTWARE FOR WEBSITE ANALYSIS

Виконав(ла): студент(ка) 4 курсу, групи КНТ-113сп

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

МАТИЦИН К.О.

(ПРИЗВИЩЕ та ініціали)

Керівник КОЛПАКОВА Т.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
“ ” 2026 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

МАТИЦИНА Кіріла Олексійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення для аналізу вебсайтів.
Software for Website Analysis

керівник проєкту (роботи) к.т.н., доцент, КОЛПАКОВА Тетяна Олексіївна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 7 ” квітня 2026 року № 139

2. Строк подання студентом проєкту (роботи) 03 червня 2026 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне
завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) _____

Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	КОЛПАКОВА Т.О., доцент		
Нормоконтроль	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання “ 7 ” квітня 2026 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

_____ Кіріл МАТИЦІН
(підпис) (Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Тетяна КОЛПАКОВА
(підпис) (Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
88 с., 3 табл., 28 рис., 3 дод., 19 джерел.

INTELLIJ IDEA, JAVA, ВЕБАНАЛІТИКА, ВЕБСАЙТ, МОВА
ПРОГРАМУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

Об'єкт дослідження – процес розробки програмного забезпечення для
аналізу вебсайтів.

Предмет дослідження – програмні засоби для аналізу вебсайтів.

Мета роботи – розробка програмного забезпечення для аналізу
вебсайтів для скорочення витрат на ручний аудит вебсайтів.

Матеріали, методи та технічні засоби: мова програмування Java,
середовище розробки IntelliJ IDEA.

Результати. Розроблено проєктні рішення для створення програмного
забезпечення для аналізу вебсайтів. Створено програмне забезпечення для
аналізу вебсайтів. Проведено тестування розробленої програмної системи для
аналізу вебсайтів.

Висновки. Мету роботи досягнуто. Розроблено програмне забезпечення
для аналізу вебсайтів за допомогою мови програмування Java та середовища
розробки IntelliJ IDEA.

Галузь використання – вебаналітика.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 88 pages, 3 tables, 28 figures, 3 appendixes, 19 sources.

INTELLIJ IDEA, JAVA, WEB ANALYTICS, WEBSITE, PROGRAMMING LANGUAGE, SOFTWARE.

The object of research is the process of developing software for website analysis.

The subject of the research is software for website analysis.

The purpose of this work is to develop software for website analysis to reduce the cost of manual website audit.

Materials, methods and technical tools: Java programming language, IntelliJ IDEA development environment.

Results. Project solutions for the creation of software for website analysis has been designed. The software to support the process of website analysis has been developed. The developed software system for website analysis has been tested.

Conclusions. The goal of the work has been achieved. The software tools for website analysis using the Java programming language and the IntelliJ IDEA development environment have been developed.

Scope of use – web analytics.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ	9
1 Аналіз предметної області	12
1.1 Програмне забезпечення для аналізу вебсайтів	12
1.2 Аналіз програмного забезпечення для аналізу вебсайтів	21
1.3 Висновки за розділом 1	30
2 Матеріали і методи	32
2.1 Вибір мови програмування.....	32
2.2 Вибір середовища розробки для створення програмного забезпечення для аналізу вебсайтів	36
2.3 Висновки за розділом 2	39
3 Опис програми	41
3.1 Структура програмного забезпечення для аналізу вебсайтів	41
3.2 Функціонування програмного забезпечення для аналізу вебсайтів.....	46
3.3 Розробка бази даних програмного забезпечення для аналізу вебсайтів .	48
3.4 Проектування інтерфейсу програмного забезпечення для аналізу вебсайтів	50
3.5 Висновки за розділом 3	53
4 Експлуатація, тестування та експериментальне дослідження програми.....	55
4.1 Призначення й умови застосування програми	55
4.2 Характеристики програми для аналізу вебсайтів.....	56
4.3 Інструкція по експлуатації програми.....	58
4.3.1 Звернення до програми.....	58
4.3.2 Вхідні й вихідні дані.....	58
4.3.3 Повідомлення.....	58
4.4 Виконання програмного забезпечення для аналізу вебсайтів	59
4.5 Тестування програмного забезпечення для аналізу вебсайтів	62
4.6 Висновки за розділом 4	62

Висновки.....	63
Перелік джерел посилання	66
Додаток А Технічне завдання.....	68
Додаток Б Фрагмент тексту програми	72
Додаток В Слайди презентації	82

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

IDE – Integrated Development Environment;

SEO – Search Engine Optimization;

UX – User eXperience;

БД – база даних;

ВА – веб аналітика;

ПЗ – програмне забезпечення.

ВСТУП

Актуальність програмного забезпечення для аналізу вебсайтів зумовлюється загальною тенденцією до цифровізації суспільства та зростанням ролі вебресурсів у діяльності організацій, установ і окремих користувачів. У сучасних умовах вебсайт розглядається не лише як інформаційний ресурс, а як повноцінний інструмент комунікації, надання послуг, реалізації товарів і формування іміджу, що потребує постійного контролю та оцінювання ефективності його функціонування. У зв'язку з цим виникає об'єктивна необхідність у використанні спеціалізованого програмного забезпечення, здатного забезпечити системний і комплексний аналіз вебресурсів [1], [2].

Застосування програмних засобів вебаналізу дозволяє здійснювати збір, обробку та інтерпретацію даних щодо структури вебсайту, поведінки користувачів, технічних характеристик і якості контенту. Отримана інформація створює основу для прийняття обґрунтованих управлінських і технічних рішень, спрямованих на підвищення ефективності функціонування вебресурсу. Особливої актуальності такі системи набувають у контексті зростання конкуренції в мережі Інтернет, коли навіть незначні недоліки у зручності використання або продуктивності можуть призвести до втрати цільової аудиторії [3], [4].

Важливим аспектом актуальності програмного забезпечення для аналізу вебсайтів є його роль у забезпеченні якості користувацького досвіду. Аналіз навігації, швидкодії, доступності та логіки взаємодії з інтерфейсом дозволяє виявляти проблемні місця та вдосконалювати структуру вебресурсу відповідно до потреб користувачів. Це, у свою чергу, сприяє підвищенню рівня задоволеності відвідувачів і досягненню стратегічних цілей власника сайту, незалежно від його призначення [5], [6].

Не менш значущою є актуальність таких програмних систем у сфері інформаційної безпеки та відповідності сучасним стандартам і вимогам.

Аналіз вебсайтів дає змогу виявляти технічні вразливості, помилки конфігурації та інші чинники, що можуть негативно впливати на стабільність і надійність роботи ресурсу. В умовах посилення вимог до захисту даних та дотримання нормативно-правових обмежень використання подібного програмного забезпечення стає необхідною складовою процесу супроводу вебсистем [5], [7].

Таким чином, актуальність програмного забезпечення для аналізу вебсайтів визначається його значенням для підвищення ефективності, якості та безпеки вебресурсів. Використання таких програмних рішень забезпечує науково обґрунтований підхід до оцінювання та оптимізації вебсайтів, що є особливо важливим у сучасному інформаційному середовищі та підтверджує доцільність подальших досліджень і розробок у даному напрямі [1]-[7].

Недоліки програмного забезпечення для аналізу вебсайтів проявляються насамперед у складності інтерпретації отриманих результатів, що зумовлює залежність ефективності використання таких систем від рівня підготовки користувача. Певним обмеженням є також технічна залежність результатів аналізу від коректності налаштування програмного забезпечення та якості вихідних даних. Неправильна інтеграція аналітичних інструментів у структуру вебсайту, використання блокувальників реклами або обмеження доступу до даних з боку браузерів і користувачів можуть знижувати повноту та достовірність зібраної інформації. Суттєвим недоліком є обмеженість універсальності програмних рішень для аналізу вебсайтів. Більшість інструментів орієнтовані на типові сценарії використання та стандартні метрики, що не завжди відповідає специфіці конкретного вебпроєкту або предметної області. Використання програмного забезпечення для аналізу вебсайтів часто пов'язане зі збиранням інформації про дії користувачів, що може викликати правові та етичні ризики [1]-[7].

Тому актуальною є розробка програмного забезпечення для аналізу вебсайтів. У дипломній кваліфікаційній роботі бакалавра розв'язується актуальне завдання розробки програмного забезпечення для аналізу вебсайтів.

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для аналізу вебсайтів;
- здійснити проєктування програмного забезпечення для аналізу вебсайтів;
- створити програмне забезпечення для аналізу вебсайтів;
- виконати тестування розробленого програмного забезпечення для аналізу вебсайтів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Програмне забезпечення для аналізу вебсайтів

Вебаналітика виступає ключовим інструментом для оцінювання того, як відвідувачі взаємодіють із вебсайтом, дозволяючи отримати глибоке розуміння аудиторії та її поведінки. Завдяки такому аналізу можливо дізнатися характеристики користувачів, включаючи географію, тип пристрою, браузер і час відвідування, а також відстежувати їхні дії на сайті: які сторінки переглядаються, скільки часу користувач проводить на кожній з них і де відбуваються кліки. Це забезпечує основу для адаптації контенту й дизайну під потреби аудиторії та оптимізації шляхів користувачів з урахуванням різних джерел трафіку і типів пристроїв [1].

Аналіз конверсій є важливим аспектом вебаналітики, оскільки він дозволяє відстежувати всі етапи взаємодії користувачів з вебсайтом, визначати ефективність форм, кнопок та навігаційних елементів, а також оцінювати результативність різних закликів до дії через тестування. Завдяки цьому можна виявляти найбільш ефективні сторінки та вдосконалювати користувацькі сценарії для підвищення відсотка досягнення цільових дій [1]

У сфері пошукової оптимізації вебаналітика забезпечує даними про ключові слова, які приваблюють трафік, а також показники популярності сторінок і поведінку користувачів після переходу на сайт. Інформація про показники відмов і швидкість завантаження сторінок дає змогу коригувати стратегію пошукової оптимізації, оптимізувати цільові сторінки та відстежувати конкурентні тенденції в пошукових системах [1]

Вебаналітика також дозволяє визначити контент, який користується найбільшою популярністю серед аудиторії, виявляючи сторінки з високим рівнем переглядів і тривалим часом перебування користувачів. Це дає змогу планувати створення нового контенту на основі реальної взаємодії відвідувачів та оптимізувати календар публікацій [1]

Дані про джерела трафіку дозволяють оцінювати ефективність каналів, через які користувачі потрапляють на сайт, будь то соціальні мережі, пошукові системи, рекламні кампанії або зворотні посилання. Це сприяє більш ефективному розподілу маркетингових ресурсів, зміцненню партнерських відносин і оперативній корекції стратегії залучення користувачів [1]

Серед видів вебаналітики виділяють зовнішню та внутрішню аналітику. Зовнішня аналітика зосереджується на активності користувачів поза межами власного вебресурсу, дозволяючи оцінювати охоплення та ефективність у порівнянні з конкурентами, використовуючи дані з різних онлайн джерел, включаючи соціальні мережі та форуми. Внутрішня аналітика фокусується на подіях, що відбуваються безпосередньо на сайті, аналізуючи популярність контенту та взаємодію відвідувачів. Для цього застосовуються технології аналізу файлів журналів серверів або позначення сторінок тегами, що дає змогу відстежувати показники переглядів, унікальних відвідувачів і переглянуті продукти. Такі методи дозволяють отримати точну інформацію про поведінку користувачів і ефективність вебресурсу загалом [1], [2].

Процес вебаналітики включає послідовну низку етапів, що забезпечують систематичне оцінювання ефективності вебсайту та поведінки його користувачів. Спершу визначаються цілі та очікувані результати діяльності вебресурсу, які можуть охоплювати збільшення продажів, підвищення задоволеності відвідувачів та зміцнення бренду. Ці завдання формулюються як у кількісних, так і якісних параметрах для чіткого відображення бізнес-цілей [1], [2].

На наступному етапі відбувається збір даних, що надходять безпосередньо з вебсайту або через спеціалізовані аналітичні інструменти. Дані формуються на основі запитів протоколу передачі гіпертексту і включають інформацію про мережеві підключення, браузері, IP-адреси та інші технічні характеристики, які дозволяють визначити географію та поведінкові патерни відвідувачів. Інтеграція цих даних із зовнішніми

джерелами надає додаткову інформацію про використання вебресурсів та взаємодію користувачів із контентом [2], [3].

Наступний етап процесу передбачає аналіз зібраної інформації для отримання практичних висновків. На цьому етапі визначаються ключові показники ефективності, такі як кількість унікальних відвідувачів, сеансів користувачів, показник відмов, пошукові запити на сайті та інші метрики, що дозволяють оцінити поведінку користувачів. Ці показники слугують основою для формування стратегій, які відповідають визначеним цілям бізнесу, наприклад, оптимізації контенту на основі популярних пошукових запитів. Подальше формування стратегії передбачає експериментування з різними підходами для визначення найбільш ефективних рішень. Методики, такі як тестування, дозволяють оцінити реакцію аудиторії на різні варіанти контенту шляхом одночасного демонстрування різних версій окремим сегментам користувачів та визначення оптимального варіанту [2], [3].

Дослідження та тестування вебсайту надають цінну інформацію про переваги та очікування відвідувачів, оскільки сайт виступає не лише як інформаційний ресурс, а й як продукт, що представляє компанію. Аналіз взаємодії користувачів з сайтом допомагає покращувати рентабельність інвестицій і підвищувати ефективність конверсійних дій [2], [3].

У контексті пошукової оптимізації вебаналітика забезпечує можливість виявлення сторінок з високою ефективністю та подальшої їх оптимізації для покращення рейтингу в пошукових системах. Відстеження ключових слів дозволяє визначати найбільш результативні пошукові запити, а моніторинг швидкості завантаження та зручності використання допомагає виявляти проблеми продуктивності, що впливають на SEO. Зменшення показника відмов досягається за рахунок вдосконалення контенту на сторінках, з яких користувачі швидко залишають сайт, а аналіз прогалин у контенті дозволяє знаходити нові можливості для задоволення інформаційних потреб аудиторії. Відстеження зворотних посилань і джерел реферального трафіку сприяє оцінці впливу зовнішніх ресурсів на силу SEO та загальну ефективність вебсайту.

Вебаналітика передбачає відстеження ключових показників, що дозволяють оцінити ефективність веб-сайту та поведінку його відвідувачів. Вимірювання метрик забезпечує розуміння того, які сторінки користуються найбільшою популярністю, скільки сеансів користувачі проводять на сайті, і які області потребують поліпшення. Інформація про унікальні перегляди дає змогу визначити реальну зацікавленість аудиторії в конкретному контенті, тоді як аналіз джерел трафіку допомагає зрозуміти, звідки приходять користувачі та які канали приносять найбільшу користь [2], [3].

Важливо також відстежувати конверсії, які показують, наскільки вебсайт ефективно досягає поставлених бізнес-цілей, таких як покупка, підписка або заповнення форми. Показник відмов демонструє, наскільки контент відповідає очікуванням користувачів і впливає на SEO, тоді як коефіцієнт повторних відвідувань відображає лояльність аудиторії та цінність контенту для постійних відвідувачів. Аналіз щомісячних унікальних відвідувачів дозволяє оцінити розширення охоплення, а показники електронної комерції, такі як відмови від кошика, середня вартість замовлення та доходи на користувача, дають уявлення про ефективність продажів і загальну продуктивність вебсайту [2], [3].

Різні типи звітів вебаналітики дозволяють отримати детальне уявлення про взаємодію користувачів із сайтом і ефективність контенту. Теплові карти та відстеження кліків показують, які області сторінки привертають найбільше уваги, а тестування і багатовимірні тести допомагають визначити оптимальні варіанти заголовків, закликів до дії та макетів для підвищення конверсії. Тестування зручності використання дозволяє виявити проблеми навігації та недоліки дизайну, тоді як опитування користувачів надають прямий зворотний зв'язок про задоволеність та якість взаємодії з сайтом. Збір цих даних сприяє ухваленню обґрунтованих рішень щодо поліпшення UX (User eXperience, UX), підвищення залученості та утримання аудиторії, а також ефективного розвитку маркетингових стратегій [2], [3].

Дослідження ключових слів та пошукової оптимізації (SEO, Search Engine Optimization) передбачає вивчення того, які терміни спричиняють трафік на сайт та як контент ранжується у пошукових системах. Це дозволяє визначити, які запити користувачі вводять найчастіше, оптимізувати сторінки для підвищення органічного рейтингу, виявляти прогалини в існуючій стратегії ключових слів та відстежувати зміни популярності термінів залежно від сезонності або тенденцій [2], [3].

Аналіз соціальних мереж концентрується на оцінці ефективності контенту на різних платформах, відстеженні залученості користувачів, поширень, кліків та приросту аудиторії. Це дає можливість зрозуміти, які канали генерують трафік, перевірити ефективність маркетингових кампаній, визначити найбільш резонансний контент та скорегувати час публікацій і налаштування таргетингу [2], [3].

Конкурентний аналіз передбачає вивчення трафіку конкурентів, їхніх стратегій ключових слів, активності у соціальних мережах і профілів зворотних посилань. Це дозволяє порівняти власні показники з конкурентами, виявити упущені можливості та потенційні загрози, відтворити успішні стратегії контенту та виявити прогалини у використанні ключових слів або зворотних посилань [2], [3].

Серед типових проблем з вебаналітикою виділяються помилки у впровадженні коду відстеження, через що збір даних стає неточним або неповним. Часто аналітичні дані накопичуються без чіткої мети, що ускладнює ухвалення рішень. Відсутність налаштованих цілей та подій не дозволяє коректно оцінювати конверсії та дії користувачів. Неправильне розуміння ключових показників, таких як показник відмов або тривалість сеансу, може призвести до хибних висновків [2], [3].

Також поширеними є проблеми з міждоменним відстеженням, відсутність сегментації за типами пристроїв, обмежена інтеграція з CRM, платформами електронної пошти або e-commerce, нерегулярні аудити даних, недотримання вимог конфіденційності та невикористання персоналізованих

панелей керування. Всі ці фактори можуть спотворювати аналітичні висновки, ускладнювати оцінку ефективності маркетингових зусиль і призводити до непродуктивних рішень [3], [4].

Вебаналітика демонструє, як компанії та маркетологи використовують дані для підвищення ефективності своїх ресурсів. Вона дозволяє відстежувати загальний трафік на сайті та визначати, у які періоди відвідувачів найбільше, що допомагає планувати промо-акції та сезонні активності. Аналіз поведінки користувачів дає змогу бачити шляхи навігації по сторінках, тривалість перебування та місця взаємодії, що дозволяє виявляти проблемні сторінки та оптимізувати їхній дизайн [3], [4].

Вебаналітика також оцінює коефіцієнти конверсії, показуючи, скільки відвідувачів досягають бажаної мети, наприклад, завершують покупки або заповнюють форми, і допомагає визначати сегменти користувачів, які потребують додаткової оптимізації. Вона дозволяє аналізувати джерела трафіку, визначаючи, звідки приходять відвідувачі, що допомагає зосереджувати ресурси на найбільш ефективних каналах [4], [5].

Важливою складовою є оцінка SEO, яка відстежує ключові слова, сторінки та пошукові запити, що приводять користувачів, та допомагає підвищувати видимість у пошукових системах. А/В-тестування дає змогу порівняти різні варіанти сторінок та визначити, яка версія забезпечує кращу конверсію. Моніторинг швидкості та продуктивності сайту дозволяє виявляти проблеми завантаження сторінок і покращувати їх за допомогою відповідних технологій [4], [5].

Аналітика також включає вивчення демографічних характеристик та інтересів аудиторії, що дозволяє адаптувати контент і рекламу до цільових груп. Відстеження подій на сайті, таких як кліки на кнопки чи перегляд відео, допомагає оптимізувати взаємодію користувачів із ключовими елементами. Для електронної комерції вебаналітика аналізує поведінку покупців, залишення кошика та дохід за товари, що дозволяє впроваджувати заходи для відновлення втрат продажів [4], [5].

У результаті вебаналітика надає комплексну картину використання сайту, показує, що працює, а що потребує змін, і допомагає приймати обґрунтовані рішення для розвитку підприємств та організацій, від оптимізації контенту та SEO до підвищення конверсій і покращення взаємодії з користувачами [4], [5].

Термін вебаналітика описує процес збору, оцінки та інтерпретації даних для розуміння того, як користувачі взаємодіють із вебсайтами. Основна мета полягає у виділенні практично цінної інформації, яка дозволяє вдосконалювати цифровий досвід та досягати стратегічних цілей. Вебаналітика концентрується не лише на підрахунку відвідувань, а й на оцінці якості трафіку та його поведінкових характеристик [5], [6].

Будь-яка взаємодія користувача, від перегляду сторінок до натискань кнопок, створює дані, які при аналізі формують уявлення про переваги та патерни поведінки. Ці дані функціонують як діагностичний інструмент, виявляючи вузькі місця та недоліки, водночас відкриваючи можливості для стратегічного розвитку. В цьому сенсі вебаналітика нагадує панель приладів у літаку: вона надає критично важливі показники, які дозволяють керувати процесами й досягати потрібних результатів [5], [6].

Сучасна вебаналітика виходить за межі простого підрахунку відвідувачів. Вона оцінює час завантаження сторінок, шляхи навігації та ефективність контенту. Наприклад, власники сайтів на WordPress можуть застосовувати плагіни на кшталт Jetpack Stats для отримання комплексного огляду продуктивності сайту та адаптації стратегії до інтересів аудиторії [5], [6].

Використання вебаналітики дозволяє глибше зрозуміти поведінку користувачів і покращити взаємодію з сайтом. Вона допомагає виявляти проблемні точки в UX/UI, оптимізувати воронки конверсій, підвищувати продуктивність ресурсу та забезпечувати максимальну віддачу від маркетингових кампаній. З її допомогою можна визначати вузькі місця на

шляху користувача, коригувати форми або сторінки продуктів і підвищувати ефективність конверсії [5], [6].

Крім того, вебаналітика дозволяє відстежувати швидкість завантаження та технічну стабільність сайту, допомагаючи своєчасно усувати проблеми, які знижують ефективність та задоволеність користувачів. Вона також сприяє точному оцінюванню маркетингових інвестицій, демонструючи, які кампанії приносять найкращий результат, і забезпечує інструменти для ретаргетингу, підтримуючи повторну взаємодію з відвідувачами, які не завершили конверсію з першого разу [5], [6].

В цілому вебаналітика надає бізнесу цінну інформацію для прийняття обґрунтованих рішень. Вона допомагає підвищити ефективність контенту, поліпшити взаємодію з аудиторією, коригувати маркетингові стратегії та краще управляти цифровими ресурсами, що робить її незамінним інструментом у сучасному онлайн середовищі [6], [7].

Показники вебаналітики охоплюють широкий спектр даних, які допомагають сформувати повну картину роботи сайту, поведінки відвідувачів та його загальної ефективності. Вони дозволяють оцінити як кількість користувачів, що відвідують ресурс, так і глибину їхньої взаємодії з контентом. Серед основних аспектів оцінюється трафік, його джерела, зацікавленість відвідувачів, конверсії та характеристики аудиторії [6], [7].

Трафік відстежується за сеансами та унікальними відвідувачами, що дає уявлення про популярність сайту і охоплення аудиторії. Аналіз переглядів сторінок і тривалості сеансів дозволяє зрозуміти, наскільки контент утримує увагу користувачів, а співвідношення нових і повернених відвідувачів вказує на здатність сайту заохочувати повторні візити [6], [7].

Рівень залученості вимірюється через показник відмов, кількість переглянутих сторінок за сеанс і середній час перебування на сторінці. Високий показник відмов або коротка взаємодія сигналізують про невідповідність очікуванням користувачів або проблеми з дизайном. Водночас

тривала взаємодія і більше переглядів сторінок свідчать про зацікавленість і релевантність контенту [6], [7].

Метрики джерел трафіку і каналів допомагають визначити, звідки приходять користувачі, будь то органічний пошук, соціальні мережі, прямі заходи або реферали, а аналіз ключових слів дає змогу вдосконалювати SEO-стратегію і точніше орієнтуватися на потреби аудиторії [6], [7].

Конверсії оцінюються через коефіцієнти виконання цільових дій, досягнення конкретних цілей і вартість конверсії, що дозволяє розуміти ефективність маркетингових зусиль. У сфері електронної комерції особливу увагу приділяють кількості транзакцій, середній вартості замовлення, показнику покинутих кошиків та продуктивності окремих продуктів, що допомагає оптимізувати продажі і приймати стратегічні рішення щодо асортименту [6], [7].

Аудиторія оцінюється за демографічними характеристиками, інтересами, географією, типом пристроїв і використовуваними браузерами чи операційними системами. Ці дані дозволяють розробникам адаптувати контент, дизайн та маркетингові кампанії відповідно до уподобань і поведінки користувачів [6], [7].

В комплексі всі ці показники створюють глибоке розуміння взаємодії з сайтом, допомагають виявляти слабкі місця, оптимізувати користувацький досвід і приймати обґрунтовані рішення для підвищення ефективності цифрових стратегій [6], [7].

Визначено, що програмні засоби для аналізу вебсайтів підтримують процеси збору, оцінки та інтерпретації даних для розуміння того, як користувачі взаємодіють із вебсайтами. Основна мета використання такого програмного забезпечення полягає у виділенні практично цінної інформації, яка дозволяє вдосконалювати цифровий досвід та досягати стратегічних цілей. Вебаналітика концентрується не лише на підрахунку відвідувань, а й на оцінці якості трафіку та його поведінкових характеристик.

1.2 Аналіз програмного забезпечення для аналізу вебсайтів

Проаналізуємо програмне забезпечення для аналізу вебсайтів [8]-[15].

Програмне забезпечення Chartbeat (рис. 1.1) є спеціалізованим аналітичним рішенням, призначеним для дослідження ефективності вебресурсів із значним обсягом публікацій, зокрема інформаційних порталів і контентно орієнтованих платформ. У процесі використання система Chartbeat забезпечує наочне відображення активності відвідувачів, дозволяючи відстежувати динаміку взаємодії з матеріалами як у поточний момент, так і в ретроспективі. Аналітичні панелі формуються таким чином, щоб надавати цілісне уявлення про поведінку аудиторії та рівень залученості до текстового й мультимедійного наповнення, що сприяє глибшому розумінню результативності опублікованого контенту [8], [13].

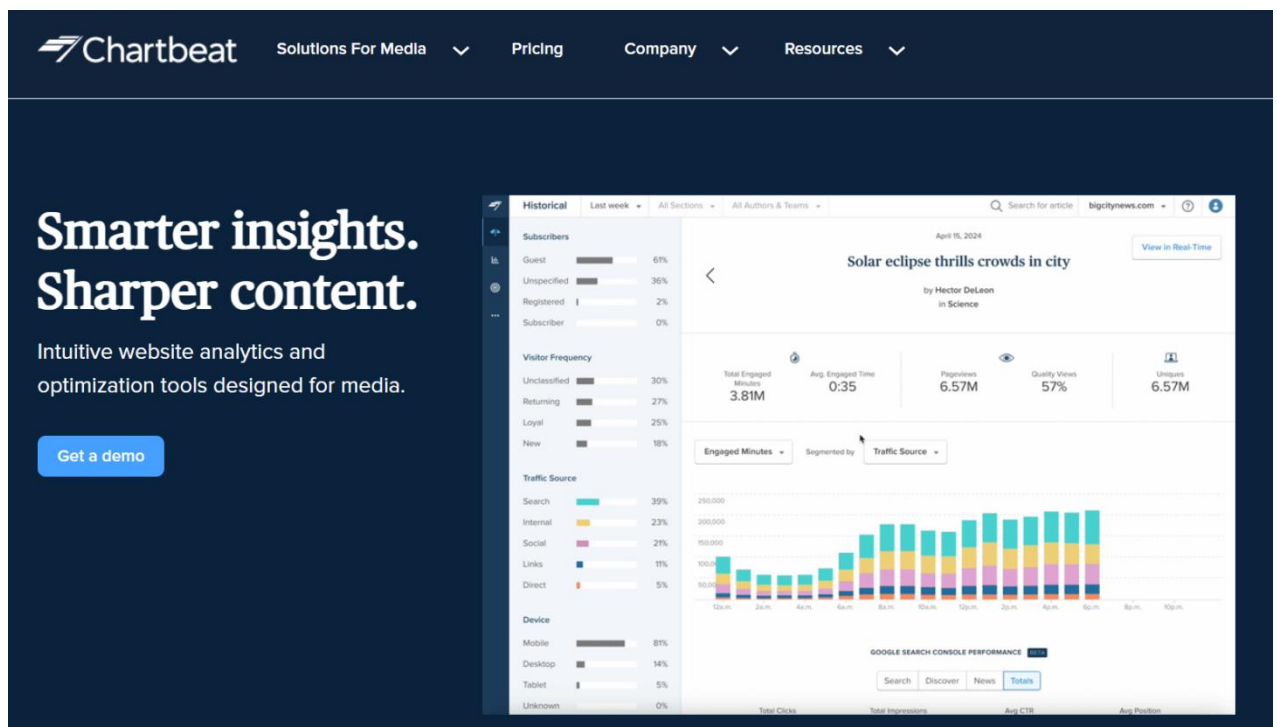


Рисунок 1.1 – Програмне забезпечення Chartbeat [13]

Особливістю Chartbeat є його адаптація до умов переважного використання мобільних пристроїв. У межах платформи реалізовано

підтримку сучасних технологій швидкого доставлення контенту та взаємодії з популярними зовнішніми сервісами, орієнтованими на мобільну аудиторію. Це дозволяє отримувати деталізовані аналітичні дані щодо споживання матеріалів на різних типах пристроїв, а також узагальнювати інформацію з декількох каналів доступу, включно з власними мобільними застосунками організації [8], [13].

Модель використання цього програмного забезпечення передбачає попереднє ознайомлення з його можливостями через демонстраційний доступ, що дає змогу оцінити доцільність впровадження платформи відповідно до потреб конкретного проєкту. Програмне забезпечення Chartbeat орієнтовано насамперед на авторів, редакторів і власників контентних ресурсів, які прагнуть не просто фіксувати показники відвідуваності, а інтерпретувати їх у контексті якості матеріалів та інтересів аудиторії. У результаті використання такого інструменту створюються умови для більш обґрунтованого планування контентної стратегії та розвитку вебресурсів у конкурентному цифровому середовищі [8], [13].

Основними характеристиками програмного забезпечення Chartbeat є такі [8], [13]:

- аналітика в реальному часі: програмне забезпечення Chartbeat забезпечує безперервне відстеження активності відвідувачів і дозволяє оперативно спостерігати за реакцією аудиторії на опублікований контент без затримок та агрегації даних за тривалий період [8], [13];

- орієнтація на контентні ресурси: функціональні можливості програмного забезпечення Chartbeat адаптовані до потреб новинних сайтів, блогів і медіаплатформ, де ключовим є аналіз залученості користувачів до текстових і мультимедійних матеріалів [8], [13];

- візуалізація показників: результати аналізу подаються у вигляді інтерактивних панелей та наочних індикаторів, що спрощує інтерпретацію даних і прийняття управлінських рішень [8], [13];

- підтримка мобільного трафіку: програмне забезпечення Chartbeat приділяє значну увагу аналізу поведінки користувачів на мобільних пристроях, що дозволяє враховувати особливості споживання контенту з різних платформ [8], [13];

- інтеграція з зовнішніми сервісами: програмне забезпечення Chartbeat може взаємодіяти з популярними технологіями поширення контенту та сторонніми платформами, розширюючи можливості збору й узагальнення аналітичної інформації [8], [13];

- фокус на залученості аудиторії: аналіз у програмному забезпечення Chartbeat не обмежується кількісними показниками відвідуваності, а спрямований на оцінювання часу взаємодії, глибини перегляду та інтересу користувачів до матеріалів [8], [13];

- гнучкість налаштувань: користувачеві програмного забезпечення Chartbeat надається можливість адаптувати відображення даних відповідно до власних аналітичних потреб і цілей контентної стратегії [8], [13].

Серед недоліків програмного забезпечення Chartbeat можна відзначити обмежену гнучкість у налаштуванні аналітичних сценаріїв, залежність від якості вхідних даних та необхідність додаткового часу для адаптації користувачів до специфіки інтерфейсу і логіки роботи системи [8], [13].

Програмне забезпечення Hotjar (рис. 1.2) є інтегрованим аналітичним рішенням, призначеним для глибокого дослідження взаємодії користувачів із вебресурсами та отримання якісного зворотного зв'язку. У процесі використання цієї платформи формується цілісне уявлення про поведінкові особливості відвідувачів, що дозволяє не лише фіксувати статистичні показники, а й інтерпретувати дії користувачів у контексті зручності та ефективності вебінтерфейсу. Такий підхід програмного забезпечення Hotjar сприяє прийняттю обґрунтованих рішень, спрямованих на вдосконалення загального користувацького досвіду [9], [10], [14].

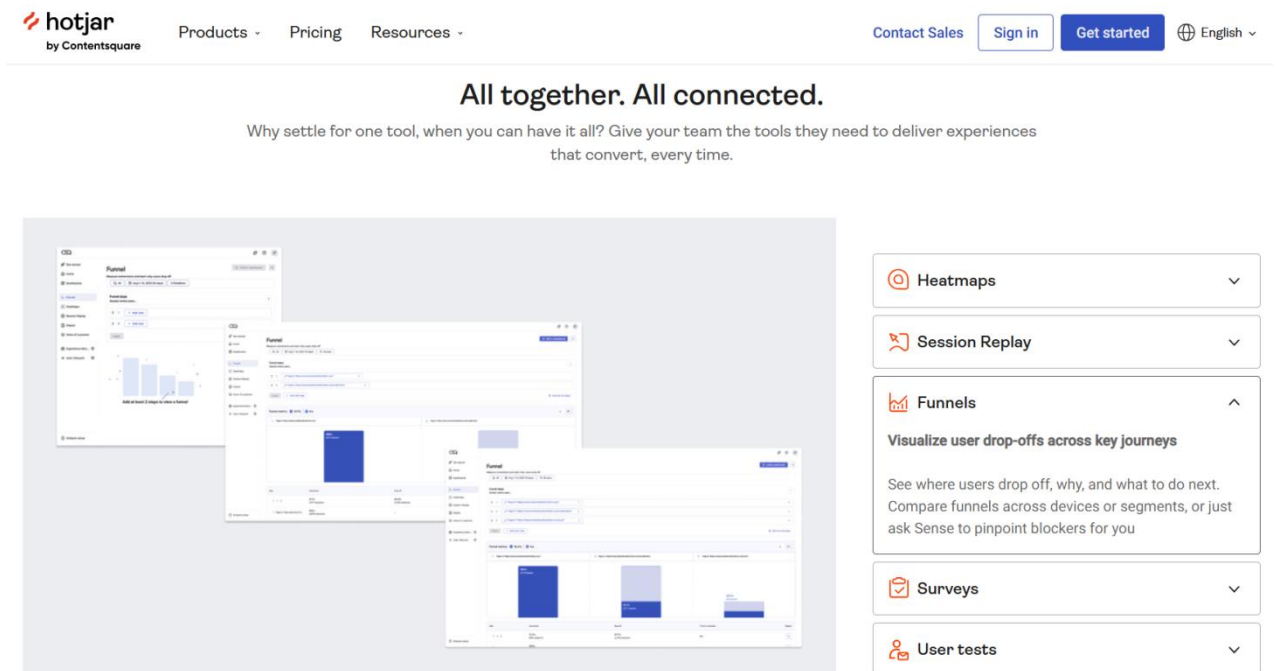


Рисунок 1.2 – Програмне забезпечення Hotjar [14]

Програмне забезпечення Hotjar орієнтоване на виявлення проблемних аспектів у структурі та дизайні вебсайтів, які можуть негативно впливати на сприйняття ресурсу або знижувати результативність взаємодії з ним. За допомогою візуальних і поведінкових інструментів стає можливим визначення зон підвищеного інтересу та елементів, що залишаються поза увагою користувачів. Аналіз рухів курсора, кліків і прокручування сторінок дозволяє оцінити логіку навігації та зрозуміти, наскільки інтерфейс відповідає очікуванням відвідувачів [9], [10], [14].

Суттєвою особливістю програмного забезпечення Hotjar є можливість детального перегляду взаємодії користувачів із сайтом у динаміці. Запис сесій дає змогу відтворювати послідовність дій відвідувачів і виявляти моменти, на яких виникають труднощі або перешкоди. Доповненням до цього є інструменти збору думок і вражень безпосередньо від користувачів, що забезпечує поєднання кількісних показників із якісними оцінками [9], [14].

У межах Hotjar реалізовано механізми аналізу шляхів, які проходять користувачі під час досягнення цільових дій, що дозволяє визначати етапи, на яких відбувається втрата інтересу або припинення взаємодії. Поєднання різних

типів аналітики формує комплексне бачення поведінки аудиторії та створює передумови для оптимізації структури сайту й підвищення його ефективності. Модель використання Hotjar передбачає різні рівні доступу до функціональних можливостей залежно від потреб і масштабу діяльності організації. Такий підхід дозволяє адаптувати програмне забезпечення як для початкового ознайомлення з поведінкою користувачів, так і для більш глибокого аналізу в умовах зростання та розвитку вебпроектів [9], [10], [14].

Програмне забезпечення Hotjar має такі особливості [9], [10], [14]:

- аналіз поведінки користувачів: програмне забезпечення Hotjar дозволяє досліджувати дії відвідувачів на вебсайті шляхом фіксації кліків, переміщень курсора та взаємодії з елементами інтерфейсу, що сприяє глибшому розумінню користувацького досвіду [9], [10], [14];

- теплові карти: програмне забезпечення Hotjar формує візуальні представлення активності користувачів, які відображають найбільш задіяні та малопомітні зони сторінок, допомагаючи оцінити ефективність структури та дизайну вебресурсу [9], [10], [14];

- записи користувацьких сесій: у програмному забезпеченні Hotjar реалізується можливість відтворення індивідуальних сеансів роботи з сайтом, що дозволяє аналізувати послідовність дій відвідувачів і виявляти потенційні проблеми навігації [9], [10], [14];

- збір зворотного зв'язку: програмне забезпечення Hotjar підтримує інтеграцію опитувань і форм для отримання думок користувачів безпосередньо під час взаємодії з сайтом, що поєднує кількісні показники з якісними оцінками [9], [10], [14];

- аналіз шляхів взаємодії: забезпечується відстеження переходів користувачів між сторінками та етапами виконання цільових дій, що дозволяє визначати вузькі місця у процесі взаємодії з вебресурсом [9], [10], [14];

- орієнтація на оптимізацію інтерфейсу: інструменти Hotjar спрямовані на вдосконалення зручності використання вебсайту та підвищення ефективності користувацьких сценаріїв [9], [10], [14];

– поєднання аналітики та якісних даних: програмне забезпечення Hotjar інтегрує поведінковий аналіз із суб'єктивними відгуками, що забезпечує комплексний підхід до оцінювання ефективності вебресурсів [9], [10], [14].

Серед недоліків програмного забезпечення Hotjar можна відзначити обмежені можливості глибокого кількісного аналізу порівняно з класичними системами вебаналітики, залежність якості результатів від обсягу зібраних сесій та необхідність ретельного налаштування інструментів для коректної інтерпретації отриманих даних [9], [10], [14].

Програмне забезпечення Plausible Analytics (рис. 1.3) є інноваційною платформою, яка використовується для аналізу вебсайтів, та де акцент зроблено на простоту сприйняття та швидкий доступ до ключових показників. Інтерфейс програмного забезпечення Plausible Analytics побудовано таким чином, щоб користувач міг одразу оцінити загальний стан вебресурсу без необхідності занурюватися у складну навігацію чи спеціалізовані звіти. Взаємодія з даними здійснюється через зрозумілі елементи керування, що дають змогу змінювати часові проміжки аналізу, спостерігати за активністю відвідувачів у поточному режимі та уточнювати інформацію за різними параметрами. Для більш точного аналізу передбачено можливість вибіркового включення або виключення окремих даних, що підвищує гнучкість роботи з аналітикою. Інформаційна панель програмного забезпечення Plausible Analytics відображає узагальнені відомості про найбільш відвідувані сторінки, типи пристроїв, географію аудиторії, канали надходження трафіку, а також досягнення цільових дій і проходження користувачами визначених сценаріїв. Для поглибленого ознайомлення з показниками у програмному забезпеченні Plausible Analytics реалізовано механізм розкриття додаткових деталей, який дозволяє отримати більш повну картину без перевантаження основного інтерфейсу [11], [15].

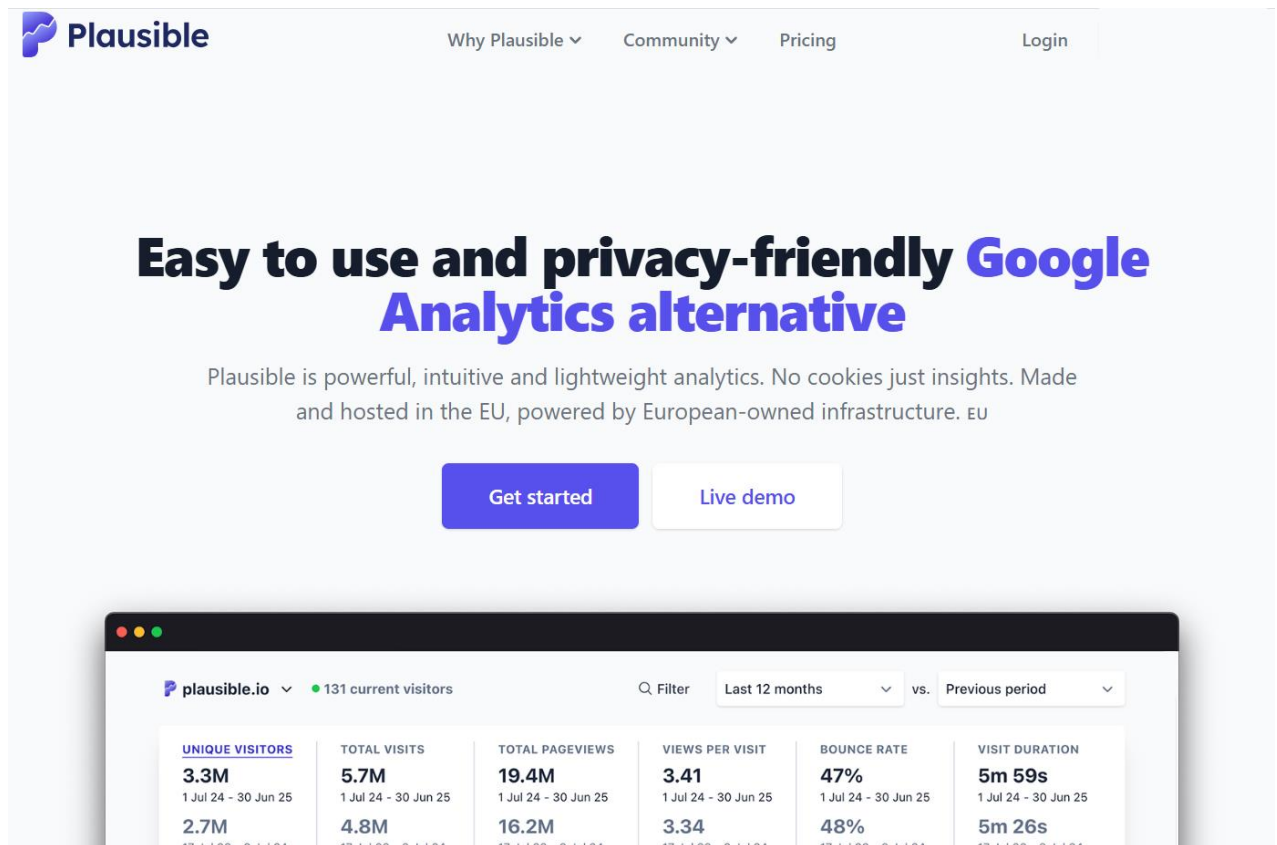


Рисунок 1.3 – Програмне забезпечення Plausible Analytics [15]

Особливістю Plausible Analytics є орієнтація на конфіденційність, оскільки платформа не здійснює збирання персональних даних відвідувачів і не використовує файли cookie. Аналітичний скрипт має малий обсяг, що мінімізує вплив на швидкодію вебсайту, а сегментація аудиторії реалізується без втручання у приватність користувачів. Завдяки цьому програмне забезпечення легко впроваджується та не потребує тривалого навчання персоналу. Водночас функціональність системи зосереджена переважно на базовому аналізі, що обмежує можливості глибокого дослідження поведінки користувачів і не передбачає тісної взаємодії з інструментами для експериментів та оптимізації інтерфейсу [11], [15].

Програмне забезпечення Plausible Analytics має такі особливості [11], [15]:

- мінімалістичний інтерфейс: програмне забезпечення Plausible Analytics має спрощену панель керування без перевантаження меню та

складних структур, що дозволяє швидко орієнтуватися в аналітичних показниках і зосереджуватися на ключових метриках [11], [15];

- аналітика в реальному часі: у програмному забезпеченні Plausible Analytics забезпечується можливість спостереження за поточною активністю відвідувачів вебсайту з оперативним оновленням даних без затримок інформації [11], [15];

- орієнтація на конфіденційність: аналіз відвідуваності за допомогою програмного забезпечення Plausible Analytics здійснюється без збору персональних даних користувачів і без застосування файлів cookie, що відповідає сучасним вимогам захисту приватності [11], [15];

- гнучка фільтрація даних: у програмному забезпеченні Plausible Analytics підтримується відбір і уточнення інформації за різними параметрами з можливістю включення або виключення окремих сегментів для більш точного аналізу [11], [15];

- огляд ключових показників: у програмному забезпеченні Plausible Analytics надається узагальнена інформація щодо популярних сторінок, джерел трафіку, пристроїв, географії аудиторії та досягнення цілей без потреби формування складних звітів [11], [15];

- деталізація показників: у програмному забезпеченні Plausible Analytics реалізовано механізм розкриття додаткової інформації, який дозволяє поглиблювати аналіз окремих метрик без ускладнення основного інтерфейсу [11], [15];

- легкий аналітичний скрипт: програмний код Plausible Analytics для збору статистики має малий обсяг і практично не впливає на швидкість завантаження вебсайту [11], [15];

- швидке впровадження: платформа Plausible Analytics легко інтегрується у вебресурси та не потребує тривалого навчання користувачів завдяки інтуїтивній логіці роботи [11], [15].

Серед недоліків програмного забезпечення Plausible Analytics можна відзначити обмежену глибину поведінкового аналізу користувачів, відсутність

розширених інструментів для дослідження взаємодії з інтерфейсом, а також недостатню підтримку інтеграцій із системами експериментування та оптимізації, що може звужувати можливості комплексної аналітики для складних вебпроектів [11], [15].

Порівняльну характеристику програмного забезпечення для аналізу вебсайтів наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для аналізу вебсайтів

Критерій порівняння	Chartbeat [13]	Hotjar [14]	Plausible Analytics [15]
реальний час аналітики	+	+	+—
поведінковий аналіз користувачів	+—	+	—
конфіденційність даних	+	+—	+
легкість впровадження	+—	+—	+
простота інтерфейсу	+—	+	+
масштабна аналітика та сегментація	+—	+—	+—

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для аналізу вебсайтів. Недоліки програмного забезпечення для аналізу вебсайтів проявляються насамперед у складності інтерпретації отриманих результатів, що зумовлює залежність ефективності використання таких систем від рівня підготовки користувача. Певним обмеженням є також технічна залежність результатів аналізу від коректності налаштування програмного забезпечення та якості вихідних даних. Неправильна інтеграція аналітичних інструментів у структуру вебсайту, використання блокувальників реклами або обмеження доступу до даних з боку браузерів і користувачів можуть знижувати повноту та

достовірність зібраної інформації. Суттєвим недоліком є обмеженість універсальності програмних рішень для аналізу вебсайтів. Більшість інструментів орієнтовані на типові сценарії використання та стандартні метрики, що не завжди відповідає специфіці конкретного вебпроєкту або предметної області. Використання програмного забезпечення для аналізу вебсайтів часто пов'язане зі збиранням інформації про дії користувачів, що може викликати правові та етичні ризики. Тому актуальною є розробка програмного забезпечення для аналізу вебсайтів.

При розробці програмного забезпечення для аналізу вебсайтів необхідно забезпечити такі функціональні вимоги:

- підтримка можливості відображення узагальнених результатів аналізу вебсайтів, зокрема, кількість переглядів сайту, кількість користувачів, кількість пройдених тестів;
- можливість пошуку ключових тем на сайтах;
- можливість відображення базової інформації про вебсайт (назва вебсайту, опис, ключові слова);
- підтримка можливості аналізу вебсайтів на повернення http-статусу;
- підтримка можливості пошуку вебсайтів за ключовим словом.

1.3 Висновки за розділом 1

Визначено, що програмні засоби для аналізу вебсайтів підтримують процеси збору, оцінки та інтерпретації даних для розуміння того, як користувачі взаємодіють із вебсайтами. Основна мета використання такого програмного забезпечення полягає у виділенні практично цінної інформації, яка дозволяє вдосконалювати цифровий досвід та досягати стратегічних цілей. Вебаналітика концентрується не лише на підрахунку відвідувань, а й на оцінці якості трафіку та його поведінкових характеристик. За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для аналізу вебсайтів. Недоліки програмного забезпечення

для аналізу вебсайтів проявляються насамперед у складності інтерпретації отриманих результатів, що зумовлює залежність ефективності використання таких систем від рівня підготовки користувача. Певним обмеженням є також технічна залежність результатів аналізу від коректності налаштування програмного забезпечення та якості вихідних даних. Неправильна інтеграція аналітичних інструментів у структуру вебсайту, використання блокувальників реклами або обмеження доступу до даних з боку браузерів і користувачів можуть знижувати повноту та достовірність зібраної інформації. Суттєвим недоліком є обмеженість універсальності програмних рішень для аналізу вебсайтів. Більшість інструментів орієнтовані на типові сценарії використання та стандартні метрики, що не завжди відповідає специфіці конкретного вебпроєкту або предметної області. Використання програмного забезпечення для аналізу вебсайтів часто пов'язане зі збиранням інформації про дії користувачів, що може викликати правові та етичні ризики. Тому актуальною є розробка програмного забезпечення для аналізу вебсайтів.

Сформульовано функціональні вимоги до програмного забезпечення для аналізу вебсайтів.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Вибір мови програмування

При розробці програмного забезпечення для аналізу вебсайтів було використано мову програмування Java [16], [17].

Мова програмування Java являє собою комплексну технологічну платформу, що сформувалася протягом тривалого періоду еволюції програмної інженерії. Її концептуальні засади ґрунтуються на ідеях суворої типізації, об'єктно-орієнтованого підходу та чітко визначених правил керування виконанням програм, що сприяє підвищенню якості програмного коду та зменшенню кількості помилок на етапах розробки й експлуатації. Однією з характерних рис Java є орієнтація на довготривалу підтримку програмних продуктів. Програмні системи, реалізовані з використанням цієї мови, зазвичай проєктуються з урахуванням подальшого розширення та модифікації, що особливо важливо для корпоративних і наукових рішень. Завдяки зворотній сумісності та стабільності стандартної бібліотеки забезпечується можливість оновлення окремих компонентів без необхідності повного перепроєктування системи [16], [17].

Важливе місце у характеристиці мови програмування Java посідає її роль у формуванні архітектурних підходів до побудови програмного забезпечення. Мова Java широко застосовується для реалізації багаторівневих і модульних архітектур, у яких логіка обробки даних, взаємодія з користувачем і доступ до ресурсів чітко розмежовані. Такий підхід сприяє підвищенню читабельності коду, спрощує тестування та забезпечує кращу керованість складних програмних систем. Окремої уваги заслуговує підтримка стандартизованих підходів до розробки програмного забезпечення, що реалізована у вигляді специфікацій та рекомендацій. Мова програмування Java активно використовується у поєднанні з усталеними шаблонами проєктування, що дозволяє формалізувати процес створення програм і зменшити залежність результату від індивідуального стилю розробника. Це є важливим чинником у

командній розробці, де необхідно забезпечити єдність підходів і прогнозованість результатів [16], [17].

Суттєвим аспектом є також освітнє та наукове значення мови програмування Java. Вона широко використовується у навчальних курсах з програмування, алгоритмів і програмної інженерії, оскільки дозволяє на практиці продемонструвати базові та поглиблені концепції, зокрема інкапсуляцію, спадкування, поліморфізм і абстракцію. Це робить Java зручним інструментом для формування системного мислення у майбутніх фахівців у галузі інформаційних технологій [16], [17].

Не менш важливою є адаптивність мови Java до сучасних тенденцій розвитку програмного забезпечення. Мова активно еволюціонує, зберігаючи при цьому спадковість основних принципів, що дозволяє інтегрувати нові можливості без радикальної зміни парадигми програмування. Це сприяє її використанню у поєднанні з сучасними підходами до розробки, зокрема з мікросервісними архітектурами, хмарними середовищами та розподіленими обчисленнями [16], [17].

Таким чином, мова програмування Java постає як універсальний і концептуально вивірений інструмент, що поєднує теоретичну обґрунтованість із практичною спрямованістю. Її значення виходить за межі окремих програмних проєктів і полягає у формуванні стабільного технологічного підґрунтя для розробки, супроводу та еволюції складних програмних систем у різних галузях застосування [16], [17].

Доцільність вибору мови програмування Java для розробки програмного забезпечення для аналізу вебсайтів обґрунтовується сукупністю технологічних, архітектурних і експлуатаційних чинників, які відповідають сучасним вимогам до аналітичних систем. Зазначена мова програмування широко використовується у створенні масштабованих і надійних програмних рішень, що є критично важливим для систем, призначених для обробки значних обсягів вебданих у режимі реального часу або з накопиченням історичних показників [16], [17].

Важливою перевагою мови програмування Java є її платформонезалежність, яка забезпечується завдяки використанню віртуальної машини. Це дозволяє розгорнути програмне забезпечення для аналізу вебсайтів на різних операційних системах без необхідності внесення змін у програмний код, що спрощує супровід, масштабування та інтеграцію системи в гетерогенне середовище. Така властивість є особливо актуальною для серверних застосунків, які часто функціонують у розподілених інфраструктурах [16], [17].

Суттєвим аргументом на користь Java є наявність розвиненої екосистеми бібліотек і фреймворків, орієнтованих на роботу з мережею, обробку HTTP-запитів, парсинг вебсторінок, аналіз структурованих і неструктурованих даних, а також взаємодію з базами даних. Це дає змогу скоротити час розробки, підвищити надійність реалізації та зосередитися на вирішенні прикладних завдань вебаналізу, а не на створенні базових механізмів з нуля [16], [17].

Не менш важливою є придатність Java до побудови високонавантажених і багатопотокових систем. Мова надає потужні засоби для паралельної обробки даних, керування потоками та синхронізації, що є необхідним для ефективного аналізу великої кількості вебресурсів або одночасної обробки запитів від багатьох користувачів. Це сприяє підвищенню продуктивності та стабільності програмного забезпечення [16], [17].

Крім того, Java характеризується високим рівнем безпеки та підтримкою сучасних механізмів захисту, що має особливе значення для систем, які працюють із мережевими даними та можуть обробляти інформацію, пов'язану з поведінкою користувачів. Наявність вбудованих засобів контролю доступу, перевірки типів і керування пам'яттю знижує ризик виникнення критичних помилок і вразливостей [16], [17].

Таким чином, вибір мови програмування Java для розробки програмного забезпечення для аналізу вебсайтів є доцільним з огляду на її платформонезалежність, масштабованість, продуктивність, безпеку та

розвинену екосистему. Сукупність цих характеристик забезпечує можливість створення ефективних, надійних і довготривало підтримуваних програмних систем, що відповідають сучасним вимогам вебаналітики [16], [17].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для аналізу вебсайтів наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для аналізу вебсайтів

Критерій порівняння мов програмування	Мова програмування		
	Java	JavaScript	PHP
Продуктивність при великих обсягах даних	+	+—	—
Масштабованість серверних застосунків	+	+—	+—
Платформонезалежність	+	+	+—
Розвиненість стандартних бібліотек для мережі та вебаналізу	+	+—	+—
Безпека та контроль виконання	+	+—	+—

Отже, для реалізації програмного забезпечення для аналізу вебсайтів обрано мову програмування Java, яка являє собою універсальний і концептуально вивірений інструмент, що поєднує теоретичну обґрунтованість із практичною спрямованістю. Крім того, мова Java має розвинену екосистему бібліотек і фреймворків, орієнтованих на роботу з мережею, обробку HTTP-запитів, парсинг вебсторінок, аналіз структурованих і неструктурованих даних, а також взаємодію з базами даних, що дає змогу скоротити час розробки, підвищити надійність реалізації та зосередитися на вирішенні прикладних завдань вебаналізу, а не на створенні базових механізмів з нуля.

2.2 Вибір середовища розробки для створення програмного забезпечення для аналізу вебсайтів

В якості середовища розробки для створення програмного забезпечення для аналізу вебсайтів було обрано середовище IntelliJ IDEA [18], [19].

Середовище розробки IntelliJ IDEA є повноцінну інтегровану платформу для інженерії програмного забезпечення, що орієнтована на підвищення якості, надійності та керованості процесу розробки. Його концепція ґрунтується на ідеї глибокого розуміння структури програмного проєкту, що реалізується через постійний аналіз коду, залежностей і конфігурацій без необхідності запуску додаткових утиліт або зовнішніх засобів [18], [19].

Важливою особливістю середовища розробки IntelliJ IDEA є підтримка формування єдиного робочого середовища для розробника, у межах якого поєднуються написання коду, його перевірка, тестування та аналіз якості. Середовище розробки IntelliJ IDEA забезпечує автоматичне застосування правил стилю, виявлення потенційних архітектурних недоліків і підказки щодо оптимізації, що сприяє формуванню структурованого та зрозумілого програмного коду. Це є особливо актуальним для складних проєктів, де важливо підтримувати узгодженість рішень упродовж усього життєвого циклу системи [18], [19].

Суттєвим аспектом є орієнтація IntelliJ IDEA на підтримку колективної розробки. Інструменти аналізу змін, перегляду історії правок і інтеграції з системами контролю версій дозволяють відстежувати еволюцію програмного коду та забезпечувати прозорість процесу розробки. Це створює умови для ефективної взаємодії між учасниками команди та зменшує ризик виникнення конфліктів у великих проєктах [18], [19].

Окремої уваги заслуговує гнучкість середовища розробки IntelliJ IDEA розробки щодо налаштування під індивідуальні потреби користувача або

специфіку проєкту. Середовище розробки IntelliJ IDEA надає широкі можливості конфігурації робочого простору, інструментів аналізу та процесів автоматизації, що дозволяє адаптувати середовище як для навчальних цілей, так і для промислової розробки. Така адаптивність середовища розробки IntelliJ IDEA сприяє підвищенню ефективності роботи та зменшенню когнітивного навантаження на розробника [18], [19].

Значущою рисою IntelliJ IDEA є її здатність підтримувати актуальність програмних рішень у довготривалій перспективі. Регулярне оновлення середовища та орієнтація на сучасні підходи до розробки забезпечують можливість поступового впровадження нових технологій без радикальної зміни інструментарію. Це особливо важливо для проєктів, що розвиваються протягом тривалого часу та потребують стабільного й передбачуваного середовища розробки [18], [19].

Таким чином, IntelliJ IDEA виступає як універсальна інженерна платформа, що поєднує інтелектуальні засоби роботи з кодом, підтримку командної взаємодії та гнучкість налаштувань. Її використання сприяє формуванню системного підходу до розробки програмного забезпечення та створює передумови для підвищення якості й надійності програмних продуктів у різних галузях застосування [18], [19].

Доцільність використання середовища розробки IntelliJ IDEA для створення програмного забезпечення для аналізу вебсайтів зумовлюється його функціональною насиченістю та орієнтацією на розробку складних і масштабованих програмних систем. Середовище розробки IntelliJ IDEA розробки забезпечує комплексну підтримку всіх етапів життєвого циклу програмного забезпечення, починаючи від написання та аналізу програмного коду й завершуючи тестуванням, налагодженням і супроводом готового продукту. Це є особливо важливим для аналітичних систем, які характеризуються значною складністю внутрішньої логіки та багатокомпонентною архітектурою [18], [19].

Однією з ключових переваг IntelliJ IDEA є глибока інтеграція з мовою програмування Java та повна підтримка її стандартів і сучасних можливостей. Середовище розробки забезпечує інтелектуальний аналіз коду в режимі реального часу, що дозволяє своєчасно виявляти синтаксичні, логічні та архітектурні помилки. Це сприяє підвищенню якості програмного забезпечення для аналізу вебсайтів і зменшенню витрат часу на етапі налагодження та виправлення дефектів [18], [19].

Важливим аспектом є також зручність роботи з великими проєктами та складними структурними залежностями. IntelliJ IDEA надає засоби навігації, рефакторингу та візуалізації структури програмного коду, що полегшує розуміння архітектури системи та спрощує її подальший розвиток. Для програмного забезпечення вебаналітичного призначення, яке часто передбачає використання багаторівневих архітектур і значної кількості модулів, така функціональність має принципове значення [18], [19].

Окремої уваги заслуговує підтримка сучасних технологій і фреймворків, що використовуються під час розробки серверних і веборієнтованих застосунків. IntelliJ IDEA забезпечує ефективну інтеграцію з інструментами керування залежностями, системами контролю версій і засобами автоматизації збірки, що сприяє організації керованого та відтворюваного процесу розробки. Це особливо актуально для проєктів, орієнтованих на аналіз вебресурсів у розподілених середовищах [18], [19].

Таким чином, використання середовища розробки IntelliJ IDEA для створення програмного забезпечення для аналізу вебсайтів є доцільним з огляду на його інтелектуальні засоби роботи з кодом, тісну інтеграцію з мовою Java, підтримку масштабних проєктів і сучасних технологій розробки. Сукупність цих характеристик забезпечує підвищення продуктивності праці розробника, якості програмного продукту та ефективності процесу розробки загалом [18], [19].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для аналізу вебсайтів наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для аналізу вебсайтів

Критерій порівняння середовищ розробки	Середовища розробки		
	IntelliJ IDEA	NetBeans	Apache JDeveloper
Інтелектуальний аналіз і підказки коду	+	+—	+—
Підтримка рефакторингу	+	+—	+—
Інтеграція з системами збірки та залежностей	+	+	+—
Продуктивність у складних багатомодульних проєктах	+	+—	—
Зручність налагодження та тестування	+	+	+—

Таким чином, для створення програмного забезпечення для аналізу вебсайтів обрано середовище розробки IntelliJ IDEA, що забезпечує глибоку інтеграцію з мовою програмування Java, повну підтримку її стандартів і сучасних можливостей, а також надає розвинені засоби інтелектуального аналізу коду та ефективну підтримку на всіх етапах життєвого циклу розробки програмного забезпечення.

2.3 Висновки за розділом 2

Для реалізації програмного забезпечення для аналізу вебсайтів обрано мову програмування Java, яка являє собою універсальний і концептуально вивірений інструмент, що поєднує теоретичну обґрунтованість із практичною спрямованістю. Крім того, мова Java має розвинену екосистему бібліотек і фреймворків, орієнтованих на роботу з мережею, обробку HTTP-запитів,

парсинг вебсторінок, аналіз структурованих і неструктурованих даних, а також взаємодію з базами даних, що дає змогу скоротити час розробки, підвищити надійність реалізації та зосередитися на вирішенні прикладних завдань вебаналізу, а не на створенні базових механізмів з нуля.

Для створення програмного забезпечення для аналізу вебсайтів обрано середовище розробки IntelliJ IDEA, що забезпечує глибоку інтеграцію з мовою програмування Java, повну підтримку її стандартів і сучасних можливостей, а також надає розвинені засоби інтелектуального аналізу коду та ефективну підтримку на всіх етапах життєвого циклу розробки програмного забезпечення.

3 ОПИС ПРОГРАМИ

3.1 Структура програмного забезпечення для аналізу вебсайтів

Структуру програмного забезпечення для аналізу вебсайтів у вигляді діаграми класів наведено на рис. 3.1.

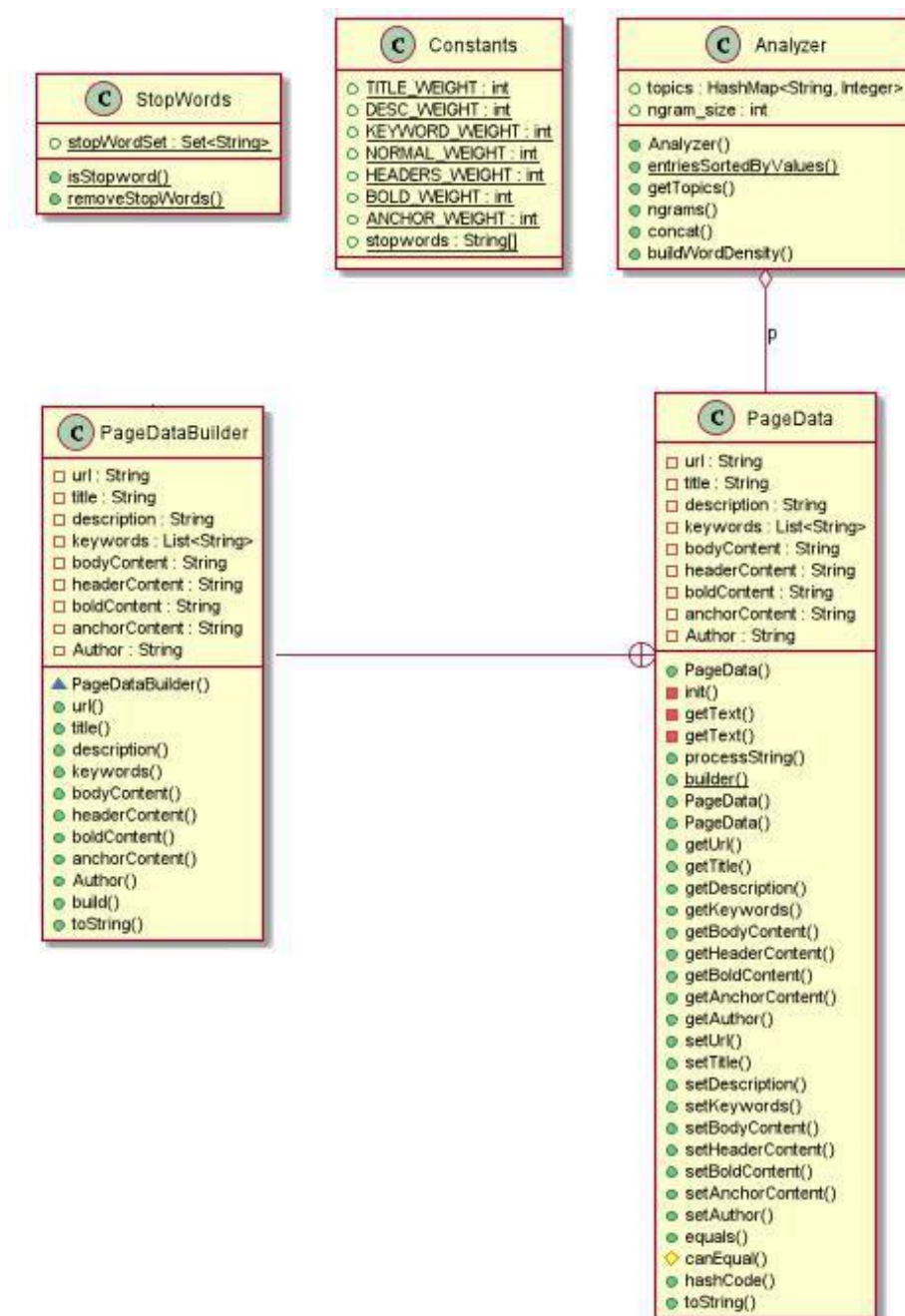


Рисунок 3.1 – Структура програмного забезпечення для аналізу вебсайтів у вигляді діаграми класів

Структура програмного забезпечення для аналізу вебсайтів, подана у вигляді діаграми класів, відображає логічний поділ функціональності системи на взаємопов'язані, але концептуально відокремлені компоненти, кожен з яких відповідає за окремий аспект процесу аналізу вебресурсів. Такий підхід забезпечує зрозумілість архітектури, спрощує супровід програмного коду та створює передумови для подальшого розширення системи без порушення її цілісності.

Центральне місце у структурі займає клас `Analyzer`, який виконує роль основного керуючого компонента системи. Саме в межах цього класу зосереджується логіка ініціалізації процесу аналізу вебсайту, організації взаємодії між іншими класами та узагальнення результатів обробки даних. Клас `Analyzer` відповідає за послідовне виконання етапів аналізу, починаючи з отримання вихідної інформації про вебсторінки та завершуючи формуванням узагальнених аналітичних показників. Його функціональність реалізується без прямого втручання у внутрішню структуру даних, що сприяє збереженню принципів інкапсуляції.

Клас `PageData` призначений для представлення результатів аналізу окремої вебсторінки у вигляді цілісного набору характеристик. У цьому класі зосереджуються дані, що описують зміст, структуру та інші параметри сторінки, необхідні для подальшої обробки або візуалізації. `PageData` виступає як модель предметної області, яка відокремлює логіку зберігання даних від логіки їх отримання та обробки. Такий підхід забезпечує гнучкість при зміні способів аналізу або розширенні набору показників без суттєвих змін у керуючих компонентах системи.

Процес створення об'єктів класу `PageData` реалізується за допомогою класу `PageDataBuilder`, який інкапсулює складну логіку формування структури даних на основі сирової інформації, отриманої під час аналізу вебсторінки. Застосування цього класу дозволяє відокремити етап побудови об'єкта від його використання, що підвищує читабельність коду та зменшує залежність між компонентами. `PageDataBuilder` забезпечує поступове наповнення об'єкта

PageData необхідними значеннями, що є доцільним у випадках, коли дані формуються на основі декількох етапів обробки.

Клас Constraints виконує допоміжну, але важливу роль у структурі програмного забезпечення, забезпечуючи централізоване зберігання незмінних параметрів і налаштувань, які використовуються в різних частинах системи. Наявність окремого класу для таких значень дозволяє уникнути дублювання та спрощує зміну параметрів аналізу без необхідності втручання у логіку основних класів. Це сприяє підвищенню керованості та узгодженості програмного коду.

Клас StopWords призначений для зберігання та обробки набору службових слів, які не мають суттєвого змістового навантаження в процесі аналізу текстового наповнення вебсторінок. Його використання дозволяє підвищити точність результатів аналізу за рахунок виключення лексичних елементів, що можуть спотворювати статистичні або семантичні показники. Інтеграція цього класу в загальну структуру системи забезпечує чітке розмежування між логікою обробки тексту та загальним механізмом аналізу вебресурсів.

Отже, представлена діаграма класів відображає структурно впорядковану та логічно узгоджену архітектуру програмного забезпечення для аналізу вебсайтів, у якій чітко визначені ролі кожного класу та характер їх взаємодії. Така структура забезпечує модульність, зручність супроводу та можливість подальшого розширення функціональності системи, що є важливою передумовою для ефективного розвитку програмного продукту.

Діаграму пакетів програмного забезпечення для аналізу вебсайтів наведено на рис. 3.2. Діаграма пакетів програмного забезпечення для аналізу вебсайтів використовується для відображення логічної організації системи на рівні укрупнених структурних одиниць та для демонстрації залежностей між ними. Така діаграма є узагальненим поданням моделі, у якому акцент робиться не на окремих класах, а на групах елементів, об'єднаних за функціональним призначенням. У цьому контексті діаграма пакетів може

розглядатися як спрощений різновид діаграми класів, що дозволяє зосередитися на архітектурних зв'язках і загальній структурі програмного забезпечення.

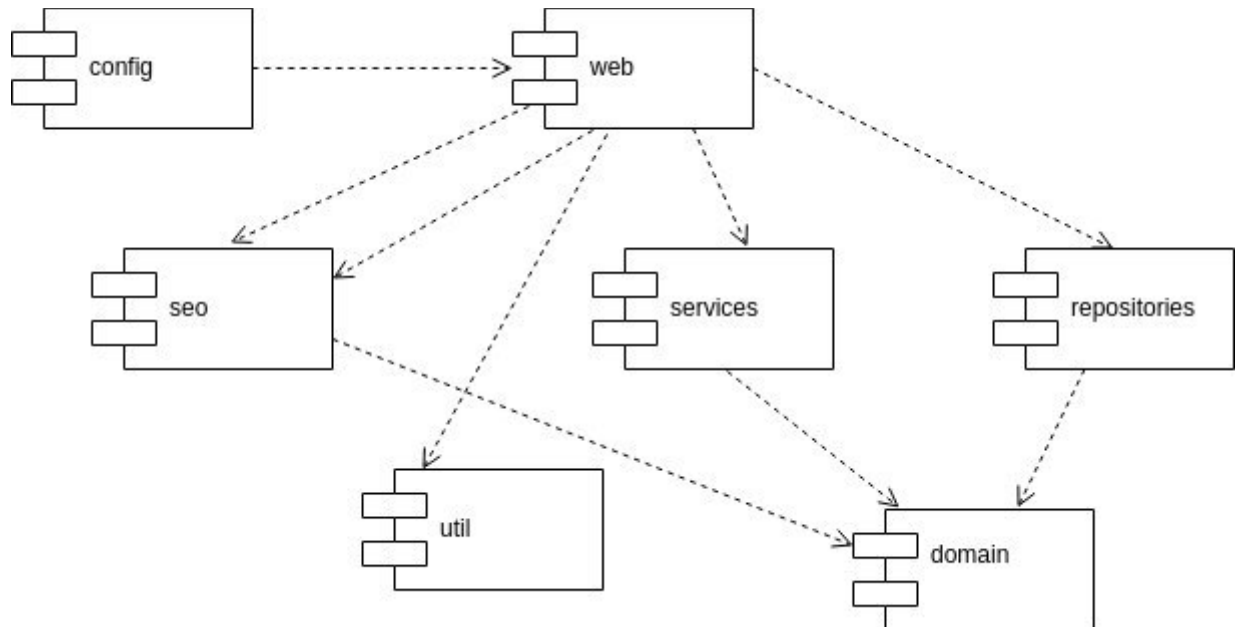


Рисунок 3.2 – Діаграма пакетів програмного забезпечення для аналізу вебсайтів

Пакет у моделі програмного забезпечення виступає як контейнер для інших елементів, забезпечуючи їх логічне групування та впорядкування. Класи та інші елементи, що входять до складу пакету, перебувають у його власності, тобто їх життєвий цикл безпосередньо пов'язаний з існуванням відповідного пакету. У разі видалення пакету всі елементи, що до нього належать, також вилучаються з моделі, що підкреслює роль пакету як цілісної структурної одиниці. Такий підхід сприяє підтриманню цілісності архітектури та спрощує керування складними програмними системами.

Діаграма пакетів програмного забезпечення для аналізу вебсайтів відображає поділ реалізованої системи на декілька логічних рівнів, кожен з яких відповідає за окрему групу функцій. У процесі розробки вебсервісу було створено значну кількість класів, які з міркувань зручності супроводу та розвитку були згруповані у відповідні пакети. Така організація дозволила

відокремити конфігураційні механізми від бізнес-логіки, логіку доступу до даних від аналітичних алгоритмів, а серверну частину від клієнтського інтерфейсу.

Пакет конфігурації об'єднує компоненти, що відповідають за налаштування параметрів функціонування вебсервісу, зокрема механізми безпеки, автентифікації, шифрування та локалізації. Пакет предметної області містить класи, які відображають сутності системи та логічно пов'язані з таблицями бази даних, забезпечуючи узгодженість між об'єктною моделлю та сховищем даних. Доступ до цих сутностей реалізується через окремий пакет репозиторіїв, у межах якого інкапсулюється логіка отримання та фільтрації даних відповідно до запитів інших компонентів системи.

Аналітична складова програмного забезпечення зосереджена у пакеті, призначеному для аналізу вебсторінок і їх пошукової оптимізації. У ньому реалізуються алгоритми обробки HTML-коду, аналізу текстового наповнення, структури сайту та перевірки коректності URL-адрес. Сервісний пакет забезпечує реалізацію бізнес-логіки, пов'язаної з автентифікацією користувачів і керуванням інформацією про них у контексті роботи вебсервісу. Допоміжні функції, такі як обробка помилок і валідація вхідних даних, винесені в окремий утилітарний пакет, що дозволяє повторно використовувати ці механізми в різних частинах системи.

Взаємодія із зовнішнім середовищем реалізується через пакет вебкомпонентів, у якому зосереджені контролери, що обробляють запити клієнтів і повертають результати аналізу для подальшого відображення. Клієнтська частина застосунку представлена кореневим пакетом вебзастосунку, який містить ресурси, доступні користувачеві безпосередньо, а також компоненти представлення, доступ до яких здійснюється опосередковано через серверну логіку.

Таким чином, діаграма пакетів наочно демонструє архітектурну організацію програмного забезпечення для аналізу вебсайтів, підкреслюючи чіткий розподіл відповідальності між компонентами та залежності між ними.

Такий підхід забезпечує структурованість, зручність супроводу й масштабування системи та створює надійну основу для її подальшого розвитку.

3.2 Функціонування програмного забезпечення для аналізу вебсайтів

Функціонування програмного забезпечення для аналізу вебсайтів подамо за допомогою схеми, зображеної на рис. 3.3.

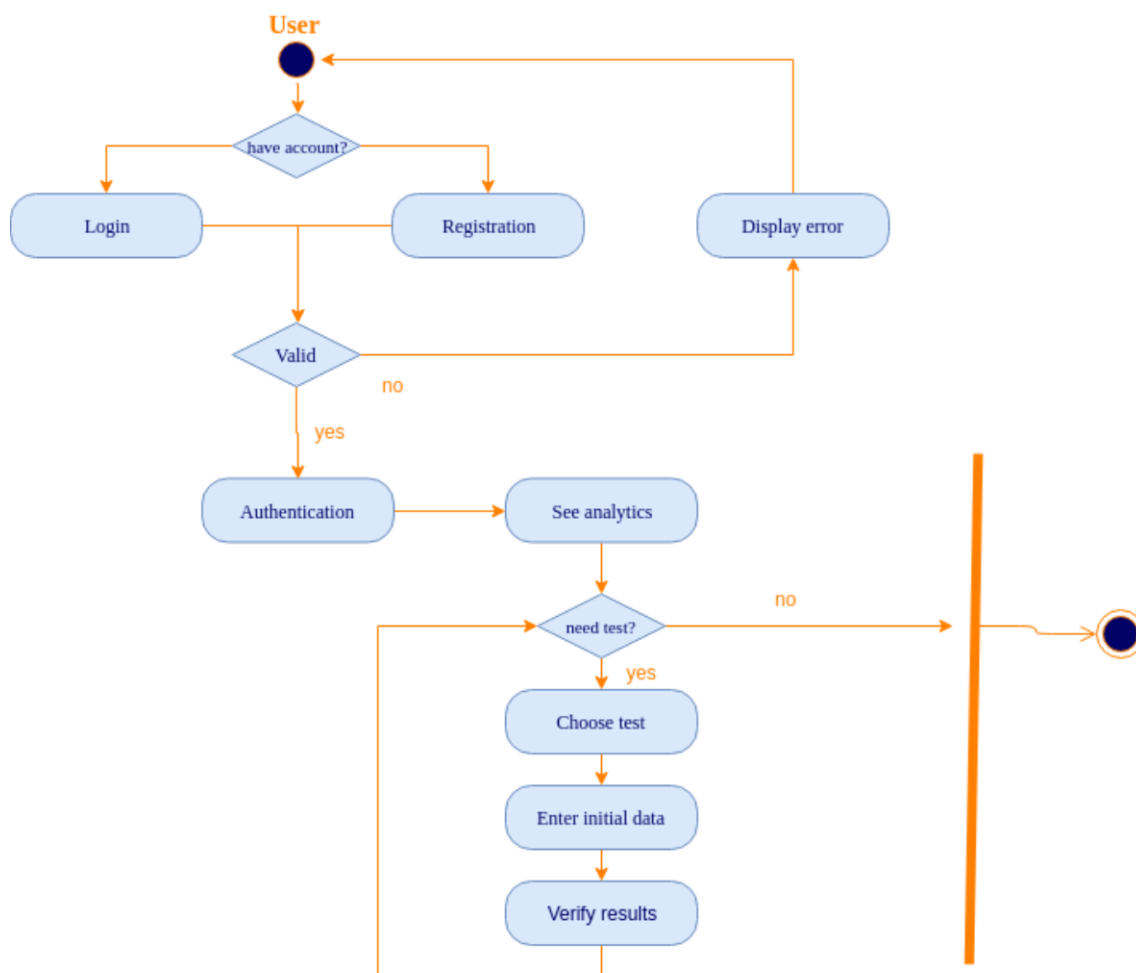


Рисунок 3.3 – Схема функціонування програмного забезпечення для аналізу вебсайтів

Функціонування програмного забезпечення для аналізу вебсайтів, представлене у вигляді діаграми діяльності, відображає послідовний та логічно впорядкований процес взаємодії користувача з системою від моменту її запуску до завершення роботи. Дана схема акцентує увагу на основних етапах обробки запитів і прийняття рішень, що забезпечують коректність доступу, достовірність результатів аналізу та гнучкість використання програмного продукту.

Початковою точкою процесу є ініціалізація роботи користувача із програмним забезпеченням. На цьому етапі система фіксує звернення до функціоналу аналізу та переводить керування до механізму перевірки наявності облікового запису. Вибіркова умова, реалізована у вигляді логічного розгалуження, дозволяє визначити подальший сценарій взаємодії. У випадку, якщо користувач уже має обліковий запис, передбачається перехід до процедури входу, під час якої здійснюється введення ідентифікаційних даних. За відсутності облікового запису ініціюється процес реєстрації, спрямований на створення нового профілю користувача та формування базових даних для подальшої автентифікації.

Після завершення етапу входу або реєстрації система переходить до перевірки коректності введеної інформації. На цьому кроці здійснюється контроль повноти, формату та логічної узгодженості даних, що є необхідною умовою безпечної та стабільної роботи програмного забезпечення. У разі успішного проходження перевірки виконується процедура автентифікації, яка підтверджує право користувача на доступ до функціональних можливостей системи. Цей етап забезпечує захист від несанкціонованого використання та формує контекст користувацької сесії.

Після успішної автентифікації користувачеві надається доступ до основного функціоналу, пов'язаного з аналізом вебсайтів. На цьому етапі відображається модуль аналізу, який дозволяє ініціювати дослідження вебресурсу та отримати відповідні аналітичні результати. Подальший перебіг процесу залежить від потреб користувача, що відображено у вигляді

наступного логічного розгалуження, пов'язаного з необхідністю проведення додаткового тестування.

У випадку, якщо проведення тестів не є потрібним, робота із системою завершується, що відповідає виходу з програми. Якщо ж користувач виявляє потребу у виконанні тестів, система переходить до етапу вибору відповідного типу тестування. Після цього здійснюється введення вхідних даних, необхідних для коректного виконання обраного тесту. Цей етап передбачає активну взаємодію користувача з програмним забезпеченням і визначає параметри подальшого аналізу.

Завершальним етапом функціонування є верифікація результатів, під час якої система перевіряє коректність отриманих даних та їх відповідність заданим умовам і очікуваним критеріям. Даний етап забезпечує надійність і достовірність результатів аналізу, що є критично важливим для прийняття подальших рішень на основі отриманої інформації.

Таким чином, діаграма діяльності наочно демонструє послідовність і логіку функціонування програмного забезпечення для аналізу вебсайтів, підкреслюючи орієнтацію системи на безпеку, зручність користування та гнучкість у виконанні аналітичних завдань.

3.3 Розробка бази даних програмного забезпечення для аналізу вебсайтів

База даних програмного забезпечення для аналізу вебсайтів побудована з урахуванням функціональних потреб системи, вимог до зберігання користувацької інформації та необхідності накопичення результатів аналітичної обробки. Її структура відображає логіку взаємодії користувачів із програмним забезпеченням, процеси автентифікації та тестування, а також механізми збереження історичних даних для подальшого аналізу й узагальнення результатів.

Схему бази даних програмного забезпечення для аналізу вебсайтів наведено на рис. 3.4.

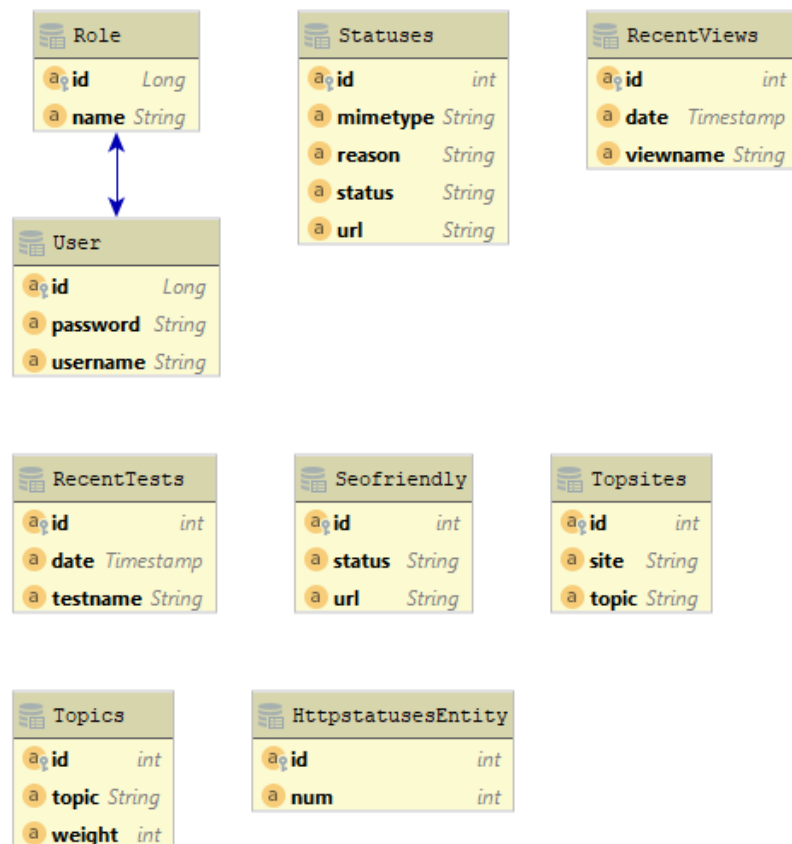


Рисунок 3.4 – Схема бази даних програмного забезпечення для аналізу вебсайтів

Основою механізму керування доступом до системи є взаємодія сутностей, що відповідають за користувачів і їх ролі. Зв'язок між цими сутностями реалізовано таким чином, щоб забезпечити гнучке призначення прав доступу та підтримку різних сценаріїв використання системи. Кожен користувач може мати декілька ролей, а кожна роль може бути пов'язана з кількома користувачами, що створює умови для реалізації складної моделі авторизації. Такий підхід дозволяє розмежовувати функціональні можливості системи залежно від типу користувача та спрощує подальше розширення політик безпеки.

Окрему групу в структурі бази даних становлять сутності, призначені для тимчасового зберігання інформації, що формується під час виконання тестів аналізу вебсайтів. Дані, пов'язані з показниками пошукової оптимізації, популярності ресурсів, тематичної спрямованості та параметрів HTTP-взаємодії, існують у межах конкретної сесії користувача. Такі сутності створюються на момент запуску тестування та використовуються для обробки і верифікації результатів, після чого можуть бути видалені або перезаписані. Це дозволяє зменшити навантаження на базу даних і забезпечити актуальність інформації, яка має значення лише в контексті поточного аналізу.

Поряд із тимчасовими даними в базі даних передбачено зберігання накопичуваної інформації, що формується в процесі тривалої експлуатації системи. Такі сутності призначені для фіксації результатів виконаних тестів, переглядів вебресурсів та змін їх станів у часі. Збереження подібних даних дозволяє відстежувати динаміку показників, аналізувати поведінку користувачів і формувати статистичні звіти. Накопичуваний характер цих сутностей забезпечує можливість ретроспективного аналізу та підвищує аналітичну цінність програмного забезпечення.

Загалом база даних програмного забезпечення для аналізу вебсайтів поєднує механізми оперативного зберігання даних, необхідних для виконання поточних тестів, із засобами довготривалого накопичення інформації для аналітики та звітності. Така організація забезпечує баланс між продуктивністю системи та глибиною аналітичних можливостей, створюючи надійну основу для стабільної роботи програмного забезпечення та його подальшого розвитку.

3.4 Проєктування інтерфейсу програмного забезпечення для аналізу вебсайтів

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для аналізу вебсайтів виконано з урахуванням вимог

технічного завдання. При розробленні програмного забезпечення для аналізу вебсайтів враховані функціональні вимоги до програми:

- підтримка можливості відображення узагальнених результатів аналізу вебсайтів, зокрема, кількість переглядів сайту, кількість користувачів, кількість пройдених тестів;
- можливість пошуку ключових тем на сайтах;
- можливість відображення базової інформації про вебсайт (назва вебсайту, опис, ключові слова);
- підтримка можливості аналізу вебсайтів на повернення http-статусу;
- підтримка можливості пошуку вебсайтів за ключовим словом.

Інтерфейс головної сторінки програмного забезпечення для аналізу вебсайтів наведено на рис. 3.5.

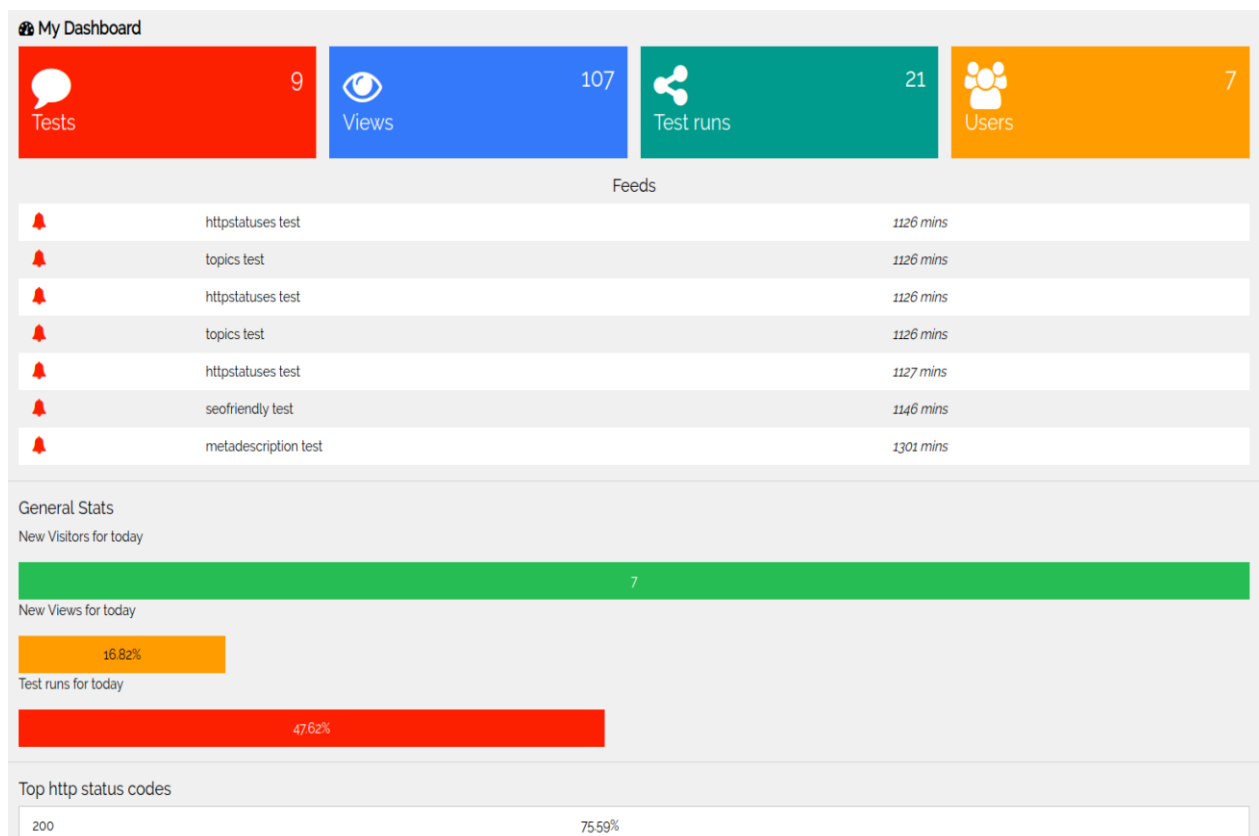


Рисунок 3.5 – Інтерфейс головної сторінки програмного забезпечення для аналізу вебсайтів

Сторінку з прикладом базової інформації про вебсайт наведено на рис. 3.6.

The screenshot shows a web form titled "Get main seo info". At the top, there is a text input field containing the URL "https://korrespondent.net/". Below this is a large grey rectangular area with a "Submit Information" button in the center. Underneath the button, the form displays the following information:

Title:
Новини - останні новини України та світу сьогодні

Description:
 Новини сьогодні - новини України сьогодні. Головні новини в Україні та світі онлайн, всі надзвичайні новини дня - політика, економіка, новини спорту та шоу-бізнесу

Author:
 Korrespondent.net

Keywords:
 новини, новини україни, новини сьогодні, останні новини, україна новини, останні новини в Україні, новини світу, новини України сьогодні, новини онлайн, новини дня

Рисунок 3.6 – Сторінка з прикладом базової інформації про вебсайт

Сторінку з результатами аналізу вебсайтів на повернення http-статусу наведено на рис. 3.7.

The screenshot shows a web form titled "Check SEO friendly". At the top, there is a text input field containing three URLs: "https://github.com/", "https://eurosport.com", and "https://flirt.com/fddffd". Below this is a large grey rectangular area with a "Submit Information" button in the center. Underneath the button, the form displays the results of the check in a table. The table has two columns: "Url" and "Status". The results are as follows:

Url	Status
https://github.com/	Success
https://eurosport.com	Success
https://flirt.com/fddffd	Success

At the bottom left of the table, there is a "Show 10 entries" dropdown menu. At the bottom right, there is a "Search:" input field.

Рисунок 3.7 – Сторінка з результатами аналізу вебсайтів на повернення http-статусу

Сторінку з результатами пошуку ключових тем на сайтах наведено на рис. 3.8.

Top 10 topics for site	
https://github.com/	
Submit Information	
Show 10 entries	Search
Topic name	Weight/Relevance
github	81
open source	36
search github jump	27
code	27
enterprise	24

Рисунок 3.8 – Сторінка з результатами пошуку ключових тем на сайтах

3.5 Висновки за розділом 3

Розроблено програмне забезпечення для аналізу вебсайтів.

Запропоновано структуру програмного забезпечення для аналізу вебсайтів у вигляді діаграми класів, що відображає структурно впорядковану та логічно узгоджену архітектуру програми для аналізу вебсайтів, у якій чітко визначені ролі кожного класу та характер їх взаємодії. Така структура забезпечує модульність, зручність супроводу та можливість подальшого розширення функціональності системи, що є важливою передумовою для ефективного розвитку програмного продукту.

Розроблено діаграму пакетів програмного забезпечення для аналізу вебсайтів, яка використовується для відображення логічної організації системи на рівні укрупнених структурних одиниць та для демонстрації залежностей між ними. Така діаграма є узагальненим поданням моделі, у

якому акцент робиться не на окремих класах, а на групах елементів, об'єднаних за функціональним призначенням.

Описано функціонування програмного забезпечення для аналізу вебсайтів, представлене у вигляді діаграми діяльності, відображає послідовний та логічно впорядкований процес взаємодії користувача з системою від моменту її запуску до завершення роботи. Дана схема акцентує увагу на основних етапах обробки запитів і прийняття рішень, що забезпечують коректність доступу, достовірність результатів аналізу та гнучкість використання програмного продукту.

Розроблено базу даних програмного забезпечення для аналізу вебсайтів, яка побудована з урахуванням функціональних потреб системи, вимог до зберігання користувацької інформації та необхідності накопичення результатів аналітичної обробки. Її структура відображає логіку взаємодії користувачів із програмним забезпеченням, процеси автентифікації та тестування, а також механізми збереження історичних даних для подальшого аналізу й узагальнення результатів.

Описано особливості реалізації програмного забезпечення для аналізу вебсайтів. Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням для аналізу вебсайтів.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Програма призначена для аналізу вебсайтів. Програма написана на мові Java, яка являє собою універсальний і концептуально вивірений інструмент, що поєднує теоретичну обґрунтованість із практичною спрямованістю. В якості середовища розробки для створення програмного забезпечення для аналізу вебсайтів обрано IntelliJ IDEA, що забезпечує глибоку інтеграцію з мовою програмування Java, повну підтримку її стандартів і сучасних можливостей, а також надає розвинені засоби інтелектуального аналізу коду та ефективну підтримку на всіх етапах життєвого циклу розробки програмного забезпечення.

Програмне забезпечення для аналізу вебсайтів забезпечує виконання таких функцій:

- підтримка можливості відображення узагальнених результатів аналізу вебсайтів, зокрема, кількість переглядів сайту, кількість користувачів, кількість пройдених тестів;
- можливість пошуку ключових тем на сайтах;
- можливість відображення базової інформації про вебсайт (назва вебсайту, опис, ключові слова);
- підтримка можливості аналізу вебсайтів на повернення http-статусу;
- підтримка можливості пошуку вебсайтів за ключовим словом.

Для роботи програмного забезпечення для аналізу вебсайтів потрібні такі технічні та програмні ресурси: персональний комп'ютер або сервер із процесором архітектури x86-64 з тактовою частотою не нижче двох гігагерців, що дозволяє ефективно виконувати обробку вебданих, мережеві запити та аналітичні обчислення. Обсяг оперативної пам'яті має становити щонайменше чотири гігабайти, оскільки під час аналізу вебсайтів здійснюється паралельна обробка сторінок, зберігання проміжних результатів і робота з об'єктами

аналітичних моделей. Для підвищення продуктивності та стабільності роботи системи доцільним є використання восьми гігабайтів оперативної пам'яті.

Зберігання програмного коду, журналів роботи, конфігураційних файлів і результатів аналізу потребує наявності вільного дискового простору обсягом не менше десяти гігабайтів, при цьому використання твердотілого накопичувача є бажаним для зменшення часу доступу до даних. Обов'язковою умовою є наявність стабільного мережевого з'єднання з доступом до мережі Інтернет, оскільки програмне забезпечення здійснює аналіз зовнішніх вебресурсів і виконує HTTP-запити до віддалених серверів.

З програмної точки зору функціонування системи передбачає використання сучасної операційної системи сімейства Windows, Linux або macOS, яка підтримує віртуальну машину Java. Для запуску програми необхідна встановлена платформа Java Development Kit відповідної версії, що забезпечує виконання байт-коду, роботу з мережевими протоколами та використання стандартних бібліотек.

Функціональні характеристики розробленого програмного забезпечення для аналізу вебсайтів наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для аналізу вебсайтів наведено у додатку Б.

4.2 Характеристики програми для аналізу вебсайтів

Програмне забезпечення для аналізу вебсайтів характеризується орієнтацією на комплексну обробку інформації, що надходить із мережевих ресурсів, та здатністю формувати узагальнені аналітичні результати на основі великого обсягу різнорідних даних. У процесі функціонування забезпечується поєднання технічного аналізу вебсторінок із дослідженням їх змістового наповнення, що дозволяє отримувати цілісне уявлення про стан і якість вебресурсу. Такий підхід сприяє виявленню як функціональних, так і семантичних особливостей досліджуваних сайтів.

Важливою характеристикою програмного забезпечення є його модульна побудова, яка передбачає чітке розмежування між компонентами збору даних, їх обробки та представлення результатів. Завдяки цьому забезпечується гнучкість системи та можливість її адаптації до різних сценаріїв використання без порушення загальної логіки роботи. Взаємодія між модулями реалізується таким чином, щоб зберігалася цілісність даних і підтримувалася стабільність функціонування навіть за умов зростання навантаження.

Програмне забезпечення для аналізу вебсайтів також вирізняється здатністю працювати в інтерактивному режимі, забезпечуючи користувачеві оперативний доступ до результатів аналізу та можливість повторного виконання досліджень із різними параметрами. У процесі роботи враховується необхідність контролю коректності вхідних даних і перевірки отриманих результатів, що підвищує достовірність аналітичних висновків. Особлива увага приділяється зручності взаємодії з користувачем, що досягається завдяки логічно впорядкованому інтерфейсу та зрозумілій подачі інформації.

Суттєвою рисою такого програмного забезпечення є орієнтація на безпеку та надійність. Реалізуються механізми автентифікації та контролю доступу, а також забезпечується захист даних, що зберігаються та обробляються в межах системи. Це дозволяє використовувати програмний продукт у середовищах, де важливими є збереження конфіденційності та цілісності інформації.

Загалом програмне забезпечення для аналізу вебсайтів характеризується поєднанням аналітичної глибини, гнучкості архітектури та орієнтації на практичне застосування. Такі властивості забезпечують можливість використання системи для вирішення широкого кола завдань, пов'язаних з оцінюванням якості, структури та ефективності вебресурсів у різних умовах експлуатації.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для запуску програмного забезпечення для аналізу вебсайтів необхідно запустити відповідний java-файл.

4.3.2 Вхідні й вихідні дані

Вхідними даними до програмного забезпечення для аналізу вебсайтів є інформація, що вводиться користувачем у процесі взаємодії із системою та визначає параметри виконання аналізу, адреси вебресурсів, що підлягають дослідженню, а також додаткові налаштування, які впливають на глибину й характер аналізу.

Вихідними даними програмного забезпечення для аналізу вебсайтів є результати, сформовані внаслідок обробки отриманих вхідних відомостей та аналізу структури й вмісту вебресурсів.

4.3.3 Повідомлення

Користувач програмного забезпечення для аналізу вебсайтів може отримати такі повідомлення: повідомлення інформаційного характеру, що підтверджують успішне виконання окремих етапів роботи системи та завершення аналізу, повідомлення попереджувального змісту, які вказують на можливі неточності у введених даних або обмеження під час виконання аналізу, а також повідомлення про помилки, що інформують про неможливість виконання певних дій у разі некоректних параметрів чи проблем доступу. Такі повідомлення спрямовані на забезпечення зворотного зв'язку з користувачем і підвищення зрозумілості процесу роботи з програмним забезпеченням.

4.4 Виконання програмного забезпечення для аналізу вебсайтів

Після запуску відображається головна сторінка з основною інформацією про вебсайт, що аналізується (рис. 4.1). Тут користувачу надаються узагальнені результати аналізу вебсайтів, зокрема, кількість переглядів сайту, кількість користувачів, кількість пройдених тестів.

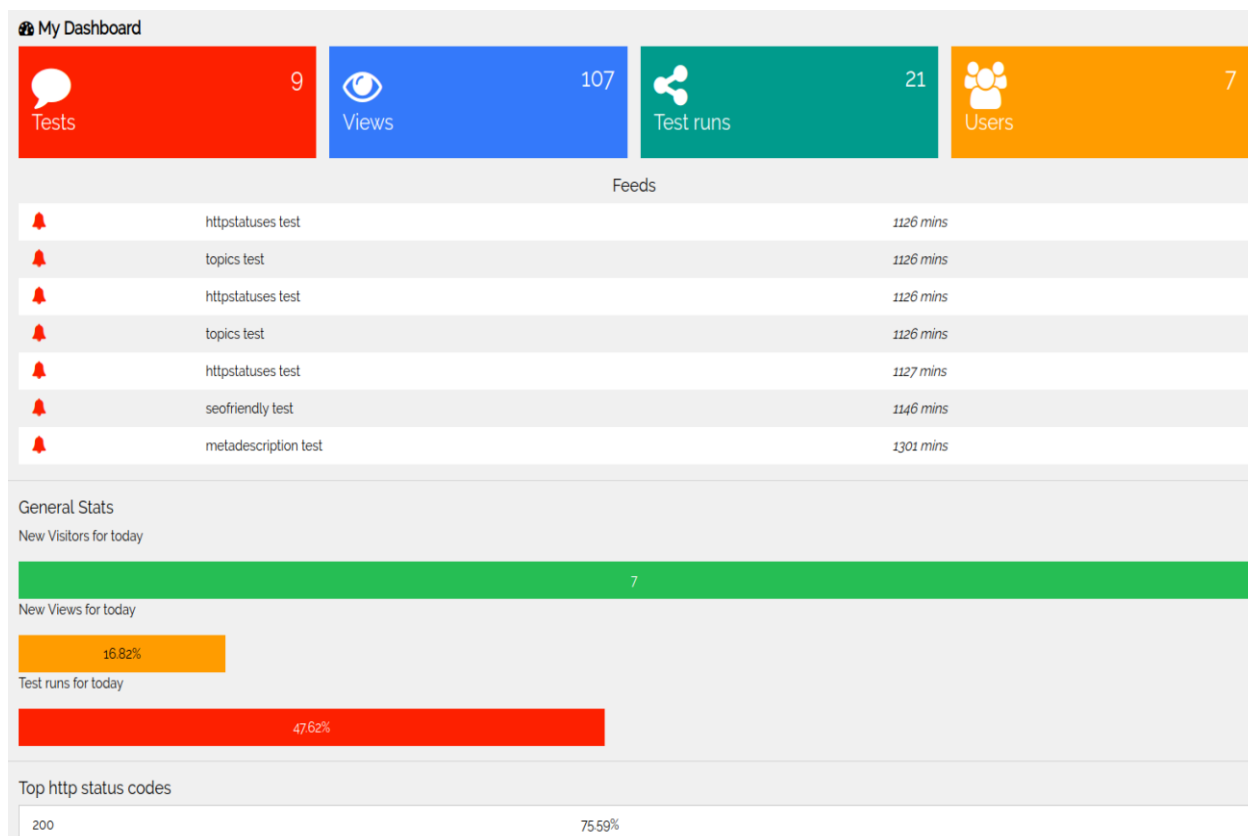


Рисунок 4.1 – Інтерфейс головної сторінки
з основною інформацією про вебсайт

При переході за відповідним посиланням відображається базова інформація про вебсайт, зокрема, його назва, опис, ключові слова (рис. 4.2).

Get main seo info

https://korrespondent.net/

Submit Information

Title:
Новини - останні новини України та світу сьогодні

Description:
Новини сьогодні - новини України сьогодні. Головні новини в Україні та світі онлайн, всі надзвичайні новини дня - політика, економіка, новини спорту та шоу-бізнесу

Author:
Korrespondent.net

Keywords:
новини, новини україни, новини сьогодні, останні новини, україна новини, останні новини в Україні, новини світу, новини України сьогодні, новини онлайн, новини дня

Рисунок 4.2 – Базова інформація про вебсайт

Для аналізу вебсайтів на повернення http-статусу (рис. 4.3) необхідно у відповідному полі ввести посилання на ці сайти, натиснути кнопку Submit information та отримати результати аналізу у відповідній таблиці.

Check SEO friendly

https://github.com/
https://eurosport.com
https://flirt.com/fddffd

Submit Information

Show 10 entries
Search:

Url	Status
https://github.com/	Success
https://eurosport.com	Success
https://flirt.com/fddffd	Success

Рисунок 4.3 – Аналіз вебсайтів на повернення http-статусу

Для пошуку ключових тем на сайтах (рис. 4.4) необхідно у відповідному полі ввести посилання на ці сайти, натиснути кнопку Submit information та отримати результати пошуку у відповідній таблиці.

Top 10 topics for site

[Submit Information](#)

Show entries Search:

Topic name	Weight/Relevance
github	81
open source	36
search github jump	27
code	27
enterprise	24

Рисунок 4.4 – Аналіз пошуку ключових тем на сайтах

Для пошуку вебсайтів за ключовим словом його необхідно ввести у рядку пошуку, натиснути кнопку та отримати результати у вигляді таблиці (рис. 4.5), де подано перелік вебсайтів та назви сторінок, що відповідають шуканому ключовому слову.

Top sites for keyword

[Submit Information](#)

Show entries Search:

Site	Topic name
https://blog.iteducenter.ua/sysadmin-basics/utility-df-i-du-v-linux/	Утилиты df и du в Linux IT Education Center Blog
http://termia.com.ua/ua/produkcija/maslonapovneni-elektro radiatori-seriji-df/	Маслонаповнені електрорадіатори серії DF
https://dfstore.com/ru/sets	Комплекты Спортивной Одежды для Фитнеса Интернет Магази ...
http://manpages.ubuntu.com/manpages/precise/ru/man1/df.1.html	Ubuntu Manpage: df - отчёт об использовании дискового ...
https://groupdf.com/uk/	Group DF – Group DF
https://uk.wikipedia.org/wiki/Df	df — Вікіпедія
https://ru.wikipedia.org/wiki/Df	df — Википедия
https://dfstore.com/ru/	Designed For Fitness: DFSTORE.COM
https://dfstore.com/	Designed For Fitness: DFSTORE.COM

Рисунок 4.5 – Аналіз пошуку вебсайтів за ключовим словом

4.5 Тестування програмного забезпечення для аналізу вебсайтів

Виконано тестування програмного забезпечення для аналізу вебсайтів.

Виявлено, що програма для аналізу вебсайтів функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу аналізу вебсайтів.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою процесу аналізу вебсайтів.

4.6 Висновки за розділом 4

Описано програмне забезпечення для аналізу вебсайтів. Виконано тестування розробленого програмного забезпечення для аналізу вебсайтів. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу аналізу вебсайтів.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для аналізу вебсайтів.

Визначено, що програмні засоби для аналізу вебсайтів підтримують процеси збору, оцінки та інтерпретації даних для розуміння того, як користувачі взаємодіють із вебсайтами. Основна мета використання такого програмного забезпечення полягає у виділенні практично цінної інформації, яка дозволяє вдосконалювати цифровий досвід та досягати стратегічних цілей. Вебаналітика концентрується не лише на підрахунку відвідувань, а й на оцінці якості трафіку та його поведінкових характеристик.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для аналізу вебсайтів. Недоліки програмного забезпечення для аналізу вебсайтів проявляються насамперед у складності інтерпретації отриманих результатів, що зумовлює залежність ефективності використання таких систем від рівня підготовки користувача. Певним обмеженням є також технічна залежність результатів аналізу від коректності налаштування програмного забезпечення та якості вихідних даних. Неправильна інтеграція аналітичних інструментів у структуру вебсайту, використання блокувальників реклами або обмеження доступу до даних з боку браузерів і користувачів можуть знижувати повноту та достовірність зібраної інформації. Суттєвим недоліком є обмеженість універсальності програмних рішень для аналізу вебсайтів. Більшість інструментів орієнтовані на типові сценарії використання та стандартні метрики, що не завжди відповідає специфіці конкретного вебпроекту або предметної області. Використання програмного забезпечення для аналізу вебсайтів часто пов'язане зі збиранням інформації про дії користувачів, що може викликати правові та етичні ризики. Тому актуальною є розробка програмного забезпечення для аналізу вебсайтів.

Сформульовано функціональні вимоги до програмного забезпечення для аналізу вебсайтів.

Для реалізації програмного забезпечення для аналізу вебсайтів обрано мову програмування Java, яка являє собою універсальний і концептуально вивірений інструмент, що поєднує теоретичну обґрунтованість із практичною спрямованістю. Крім того, мова Java має розвинену екосистему бібліотек і фреймворків, орієнтованих на роботу з мережею, обробку HTTP-запитів, парсинг вебсторінок, аналіз структурованих і неструктурованих даних, а також взаємодію з базами даних, що дає змогу скоротити час розробки, підвищити надійність реалізації та зосередитися на вирішенні прикладних завдань вебаналізу, а не на створенні базових механізмів з нуля.

Для створення програмного забезпечення для аналізу вебсайтів обрано середовище розробки IntelliJ IDEA, що забезпечує глибоку інтеграцію з мовою програмування Java, повну підтримку її стандартів і сучасних можливостей, а також надає розвинені засоби інтелектуального аналізу коду та ефективну підтримку на всіх етапах життєвого циклу розробки програмного забезпечення.

Розроблено програмне забезпечення для аналізу вебсайтів.

Запропоновано структуру програмного забезпечення для аналізу вебсайтів у вигляді діаграми класів, що відображає структурно впорядковану та логічно узгоджену архітектуру програми для аналізу вебсайтів, у якій чітко визначені ролі кожного класу та характер їх взаємодії. Така структура забезпечує модульність, зручність супроводу та можливість подальшого розширення функціональності системи, що є важливою передумовою для ефективного розвитку програмного продукту.

Розроблено діаграму пакетів програмного забезпечення для аналізу вебсайтів, яка використовується для відображення логічної організації системи на рівні укрупнених структурних одиниць та для демонстрації залежностей між ними. Така діаграма є узагальненим поданням моделі, у

якому акцент робиться не на окремих класах, а на групах елементів, об'єднаних за функціональним призначенням.

Описано функціонування програмного забезпечення для аналізу вебсайтів, представлене у вигляді діаграми діяльності, відображає послідовний та логічно впорядкований процес взаємодії користувача з системою від моменту її запуску до завершення роботи. Дана схема акцентує увагу на основних етапах обробки запитів і прийняття рішень, що забезпечують коректність доступу, достовірність результатів аналізу та гнучкість використання програмного продукту.

Розроблено базу даних програмного забезпечення для аналізу вебсайтів, яка побудована з урахуванням функціональних потреб системи, вимог до зберігання користувацької інформації та необхідності накопичення результатів аналітичної обробки. Її структура відображає логіку взаємодії користувачів із програмним забезпеченням, процеси автентифікації та тестування, а також механізми збереження історичних даних для подальшого аналізу й узагальнення результатів.

Описано особливості реалізації програмного забезпечення для аналізу вебсайтів. Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням для аналізу вебсайтів.

Виконано тестування розробленого програмного забезпечення для аналізу вебсайтів. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу аналізу вебсайтів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. What is Web Analytics? Definition, Metrics and Examples. URL: <https://www.theknowledgeacademy.com/blog/what-is-web-analytics/> (date of access: 14.04.2026).
2. Web Analytics Guide: Importance, Purpose, Benefits, & Common Metrics. URL: <https://jetpack.com/resources/web-analytics/> (date of access: 14.04.2026).
3. A Beginner's Guide to Web Analytics. URL: <https://inkbotdesign.com/web-analytics/> (date of access: 14.04.2026).
4. What is Web Analytics. URL: https://www.tutorialspoint.com/web_analytics/web_analytics_introduction.htm (date of access: 14.04.2026).
5. What is Web Analytics? Definition, Metrics & Best Practices. URL: <https://uxcam.com/blog/what-is-web-analytics/> (date of access: 14.04.2026).
6. What is Web Analytics. URL: <https://engaiodigital.com/what-is-web-analytics/> (date of access: 14.04.2026).
7. Web Analytics Tools: Best Solutions to Track and Optimize Performance. URL: <https://growthnatives.com/blogs/web-analytics/web-analytics-tools-techniques-and-best-practices/> (date of access: 14.04.2026).
8. Best Web Analytics Tools to Use. URL: <https://www.wix.com/blog/best-website-analytics-tools> (date of access: 14.04.2026).
9. Best Website Analytics Tools. URL: <https://analytify.io/best-website-analytics-tools/> (date of access: 14.04.2026).
10. The 15 Best Web Analytics Tools. URL: <https://uxcam.com/blog/web-analytics-tools/> (date of access: 14.04.2026).
11. Best Web Analytics Tools in 2026 Rated by Industry Professionals. URL: <https://vwo.com/blog/web-analytics-tools/> (date of access: 14.04.2026).

12. Best Website Analytics Tools for Visitor Stats. URL: <https://thecxlead.com/tools/best-website-analytics-tools/> (date of access: 14.04.2026).

13. Chartbeat. URL: <https://chartbeat.com/> (date of access: 14.04.2026).

14. Hotjar. URL: <https://www.hotjar.com/> (date of access: 14.04.2026).

15. Plausible Analytics. URL: <https://plausible.io/> (date of access: 14.04.2026).

16. The Destination for Java Developers. URL: <https://dev.java/> (date of access: 14.04.2026).

17. Java Environment Setup. URL: https://www.tutorialspoint.com/java/java_environment_setup.htm (date of access: 14.04.2026).

18. What is IntelliJ IDEA - A Comprehensive Guide. URL: <https://www.simplilearn.com/what-is-intellij-idea-article> (date of access: 14.04.2026).

19. IntelliJ IDEA Features overview. URL: <https://www.jetbrains.com/idea/features/> (date of access: 14.04.2026).

ДОДАТОК А
Технічне завдання

Вступ

Програмне забезпечення може використовуватися для аналізу вебсайтів.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмне забезпечення для аналізу вебсайтів», затверджене наказом Національного університету «Запорізька політехніка» № 139 від 7 квітня 2026 р.

A.2 Призначення розробки

Програмний продукт призначений для аналізу вебсайтів.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Програмне забезпечення для аналізу вебсайтів забезпечує виконання таких функцій:

- підтримка можливості відображення узагальнених результатів аналізу вебсайтів, зокрема, кількість переглядів сайту, кількість користувачів, кількість пройдених тестів;
- можливість пошуку ключових тем на сайтах;
- можливість відображення базової інформації про вебсайт (назва вебсайту, опис, ключові слова);
- підтримка можливості аналізу вебсайтів на повернення http-статусу;
- підтримка можливості пошуку вебсайтів за ключовим словом.

А.3.2 Вимоги до інтерфейсу програми

Інтерфейс програмного забезпечення для аналізу вебсайтів повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

А.3.3 Вимоги до надійності

Програмне забезпечення для аналізу вебсайтів повинно забезпечити надійне функціонування.

А.3.4 Умови експлуатації

Для експлуатації програмного забезпечення для аналізу вебсайтів необхідна наявність персонального комп'ютера.

А.3.5 Вимоги до складу та параметрів технічний засобів

Для роботи програмного забезпечення для аналізу вебсайтів потрібні такі технічні та програмні ресурси: персональний комп'ютер або сервер із процесором архітектури x86-64 з тактовою частотою не нижче двох гігагерців, що дозволяє ефективно виконувати обробку вебданих, мережеві запити та аналітичні обчислення. Обсяг оперативної пам'яті має становити щонайменше чотири гігабайти, оскільки під час аналізу вебсайтів здійснюється паралельна обробка сторінок, зберігання проміжних результатів і робота з об'єктами аналітичних моделей. Для підвищення продуктивності та стабільності роботи системи доцільним є використання восьми гігабайтів оперативної пам'яті.

Зберігання програмного коду, журналів роботи, конфігураційних файлів і результатів аналізу потребує наявності вільного дискового простору обсягом не менше десяти гігабайтів, при цьому використання твердотілого накопичувача є бажаним для зменшення часу доступу до даних. Обов'язковою умовою є наявність стабільного мережевого з'єднання з доступом до мережі Інтернет, оскільки програмне забезпечення здійснює аналіз зовнішніх вебресурсів і виконує HTTP-запити до віддалених серверів.

А.3.6 Вимоги до маркування і пакування

Програма для аналізу вебсайтів може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для аналізу вебсайтів».

А.4 Вхідні дані до роботи

Вхідними даними до програмного забезпечення для аналізу вебсайтів є інформація, що вводиться користувачем у процесі взаємодії із системою та визначає параметри виконання аналізу, адреси вебресурсів, що підлягають дослідженню, а також додаткові налаштування, які впливають на глибину й характер аналізу.

Вихідними даними програмного забезпечення для аналізу вебсайтів є результати, сформовані внаслідок обробки отриманих вхідних відомостей та аналізу структури й вмісту вебресурсів.

ДОДАТОК Б
Фрагмент тексту програми


```

public class Nlzrr {
    public MpgDt nlzPg(String rl) {
        MpgDt dt = new MpgDt();
        dt.stRl(rl);
        String cnt = "";
        try {
            URL r = new URL(rl);
            BufferedReader rdr = new BufferedReader(new
InputStreamReader(r.openStream()));
            String ln;
            while ((ln = rdr.readLine()) != null) {
                cnt = cnt + ln;
            }
            rdr.close();
        } catch (Exception x) {
            cnt = cnt + "";
        }
        dt.stCnt(cnt);
        int lng = cnt.length();
        dt.stLng(lng);
        int wrdCnt = 0;
        boolean nWrd = true;
        for (int i = 0; i < cnt.length(); i++) {
            char c = cnt.charAt(i);
            if (Character.isWhitespace(c)) {
                if (!nWrd) {
                    wrdCnt = wrdCnt + 1;
                }
                nWrd = true;
            } else {
                nWrd = false;
            }
        }
        if (!nWrd) {
            wrdCnt = wrdCnt + 1;
        }
        dt.stWrdCnt(wrdCnt);
        int tgCnt = 0;
        for (int i = 0; i < cnt.length(); i++) {
            if (cnt.charAt(i) == '<') {

```

```

        tgCnt = tgCnt + 1;
    }
}
dt.stTgCnt(tgCnt);
return dt;
}
}

```

```

class MpgDt {
    private String rl;
    private String cnt;
    private int lng;
    private int wrdCnt;
    private int tgCnt;

    public void stRl(String v) {
        rl = v;
    }
}

```

```

class DtBldr {
    public Map<String, Object> bld(MpgDt dt) {
        Map<String, Object> mp = new HashMap<>();
        mp.put("rl", dt.gtRl());
        mp.put("lng", dt.gtLng());
        mp.put("wrд", dt.gtWrdCnt());
        mp.put("tg", dt.gtTgCnt());
        String s = dt.gtCnt();
        int ltr = 0;
        for (int i = 0; i < s.length(); i++) {
            char c = s.charAt(i);
            if (Character.isLetter(c)) {
                ltr = ltr + 1;
            }
        }
        mp.put("ltr", ltr);
        return mp;
    }
}
}

```

```

class WbCntrlr {
    private final Nlzrr nlzrr;
}

```

```

private final DtBldr dtBldr;

public WbCntrlr(Nlzrr n, DtBldr d) {
    nlzrr = n;
    dtBldr = d;
}

@RequestMapping(value = "/nlz", method = RequestMethod.POST)
@ResponseBody
public Map<String, Object> prcss(HttpServletRequestRequest rq) {
    String rl = rq.getParameter("rl");
    if (rl == null) {
        rl = "";
    }
    MpgDt dt = nlzrr.nlzPg(rl);
    Map<String, Object> rs = dtBldr.bld(dt);
    rs.put("stts", "k");
    return rs;
}
}

class TstSrv {
    public Map<String, Object> rnTst(String rl) {
        Map<String, Object> mp = new HashMap<>();
        String rs = "";
        for (int i = 0; i < rl.length(); i++) {
            char c = rl.charAt(i);
            if (Character.isLetter(c)) {
                rs = rs + c;
            }
        }
        mp.put("flt", rs);
        int hsh = rs.hashCode();
        mp.put("hsh", hsh);
        boolean vld = hsh % 2 == 0;
        mp.put("vld", vld);
        return mp;
    }
}

class StpWrd {

```

```

private Set<String> st;

public StpWrd() {
    st = new HashSet<>();
    st.add("th");
    st.add("nd");
    st.add("fr");
    st.add("tht");
    st.add("thrs");
}

public boolean hs(String w) {
    return st.contains(w);
}
}

class TxtSrv {
    private final StpWrd stp;

    public TxtSrv(StpWrd s) {
        stp = s;
    }

    public Map<String, Integer> prcssTxt(String t) {
        Map<String, Integer> mp = new HashMap<>();
        String[] prts = t.split("\\s+");
        for (int i = 0; i < prts.length; i++) {
            String w = prts[i].toLowerCase();
            if (!stp.hs(w)) {
                Integer c = mp.get(w);
                if (c == null) {
                    mp.put(w, 1);
                } else {
                    mp.put(w, c + 1);
                }
            }
        }
        return mp;
    }
}

```

```

class TstDbSrv {
    private final TstRsltRepo repo;

    @Autowired
    public TstDbSrv(TstRsltRepo r) {
        repo = r;
    }

    public TstRslt svTstRslt(String rl, boolean vld, int scr) {
        TstRslt tr = new TstRslt();
        tr.stRl(rl);
        tr.stVld(vld);
        tr.stScr(scr);
        return repo.save(tr);
    }

    public List<TstRslt> gtAllTstRslt() {
        return repo.findAll();
    }

    public Optional<TstRslt> gtTstById(Long id) {
        return repo.findById(id);
    }
}

@Service
class LgnSrv {
    private Map<String, String> usrs;

    public LgnSrv() {
        usrs = new HashMap<>();
    }

    public boolean rgrUsr(String nm, String psw) {
        if (usrs.containsKey(nm)) return false;
        usrs.put(nm, psw);
        return true;
    }

    public boolean lgnUsr(String nm, String psw) {
        return psw.equals(usrs.get(nm));
    }
}

```

```

    }

    public boolean chngPsw(String nm, String oldPsw, String
newPsw) {
        if (lgnUsr(nm, oldPsw)) {
            usrs.put(nm, newPsw);
            return true;
        }
        return false;
    }
}

```

```

@Service
class SttsSrv {
    private Map<String, String> sts;

    public SttsSrv() {
        sts = new HashMap<>();
    }

    public void updtStts(String rl, String st) {
        sts.put(rl, st);
    }

    public String gtStts(String rl) {
        return sts.getOrDefault(rl, "nd");
    }
}

```

```

@Service
class TxtHstSrv {
    private List<String> hst;

    public TxtHstSrv() {
        hst = new ArrayList<>();
    }

    public void addTxt(String txt) {
        hst.add(txt);
    }
}

```

```

    public List<String> gtHst() {
        return new ArrayList<>(hst);
    }

    public void clrHst() {
        hst.clear();
    }
}

class WbApiCntrlr {

    private final TstDbSrv tstSrv;
    private final LgnSrv lgnSrv;
    private final SttsSrv sttsSrv;
    private final TxtHstSrv txtHstSrv;

    public WbApiCntrlr(TstDbSrv t, LgnSrv l, SttsSrv s, TxtHstSrv
th) {
        tstSrv = t;
        lgnSrv = l;
        sttsSrv = s;
        txtHstSrv = th;
    }

    public Map<String, Object> rgrUsr(@RequestParam String nm,
@RequestParam String psw) {
        Map<String, Object> mp = new HashMap<>();
        boolean rs = lgnSrv.rgrUsr(nm, psw);
        mp.put("stts", rs ? "ok" : "fail");
        return mp;
    }

    public Map<String, Object> lgnUsr(@RequestParam String nm,
@RequestParam String psw) {
        Map<String, Object> mp = new HashMap<>();
        boolean rs = lgnSrv.lgnUsr(nm, psw);
        mp.put("stts", rs ? "ok" : "fail");
        return mp;
    }
}

```

```

    public Map<String, Object> chngPsw(@RequestParam String nm,
    @RequestParam String oldPsw, @RequestParam String newPsw) {
        Map<String, Object> mp = new HashMap<>();
        boolean rs = lgnSrv.chngPsw(nm, oldPsw, newPsw);
        mp.put("stts", rs ? "ok" : "fail");
        return mp;
    }

    public Map<String, Object> svTst(@RequestParam String rl,
    @RequestParam boolean vld, @RequestParam int scr) {
        Map<String, Object> mp = new HashMap<>();
        TstRslt tr = tstSrv.svTstRslt(rl, vld, scr);
        mp.put("id", tr.getId());
        mp.put("stts", "ok");
        return mp;
    }

    public List<TstRslt> gtAllTst() {
        return tstSrv.gtAllTstRslt();
    }

    public Map<String, String> gtStts(@RequestParam String rl) {
        Map<String, String> mp = new HashMap<>();
        String st = sttsSrv.gtStts(rl);
        mp.put("rl", rl);
        mp.put("stts", st);
        return mp;
    }

    public Map<String, Object> updtStts(@RequestParam String rl,
    @RequestParam String st) {
        sttsSrv.updtStts(rl, st);
        Map<String, Object> mp = new HashMap<>();
        mp.put("stts", "ok");
        return mp;
    }

    @PostMapping("/addtxt")
    public Map<String, Object> addTxt(@RequestParam String txt) {
        txtHstSrv.addTxt(txt);
        Map<String, Object> mp = new HashMap<>();

```



```
        mp.put("stts", "ok");
        return mp;
    }

    public List<String> gtHst() {
        return txtHstSrv.gtHst();
    }

    public Map<String, Object> clrHst() {
        txtHstSrv.clrHst();
        Map<String, Object> mp = new HashMap<>();
        mp.put("stts", "ok");
        return mp;
    }
}
```

ДОДАТОК В
Слайди презентації

Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів

Програмне забезпечення для аналізу вебсайтів

Виконав: Кіріл МАТИЦІН

ст. групи КНТ-113сп

Керівник: Тетяна КОЛПАКОВА

к.т.н., доцент НУ «Запорізька політехніка»

Рисунок В.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження – процес розробки програмного забезпечення для аналізу вебсайтів.

Предмет дослідження – програмні засоби для аналізу вебсайтів.

Метою роботи є розробка програмного забезпечення для аналізу вебсайтів для скорочення витрат на ручний аудит вебсайтів.

2

Рисунок В.2 – Слайд 2

Завдання роботи

Для досягнення поставленої мети у кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для аналізу вебсайтів;
- здійснити проектування програмного забезпечення для аналізу вебсайтів;
- створити програмне забезпечення для аналізу вебсайтів;
- виконати тестування розробленого програмного забезпечення для аналізу вебсайтів.

3

Рисунок В.3 – Слайд 3

Порівняння ПЗ для аналізу вебсайтів

Критерій порівняння	Chartbeat	Hotjar	Plausible Analytics
реальний час аналітики	+	+	+—
поведінковий аналіз користувачів	+—	+	—
конфіденційність даних	+	+—	+
легкість впровадження	+—	+—	+
простота інтерфейсу	+—	+	+
масштабна аналітика та сегментація	+—	+—	+—

4

Рисунок В.4 – Слайд 4

Вибір мови програмування

Критерій порівняння мов програмування	Мова програмування		
	Java	JavaScript	PHP
Продуктивність при великих обсягах даних	+	+—	—
Масштабованість серверних застосунків	+	+—	+—
Платформонезалежність	+	+	+—
Розвиненість стандартних бібліотек для мережі та вебаналізу	+	+—	+—
Безпека та контроль виконання	+	+—	+—

5

Рисунок В.5 – Слайд 5

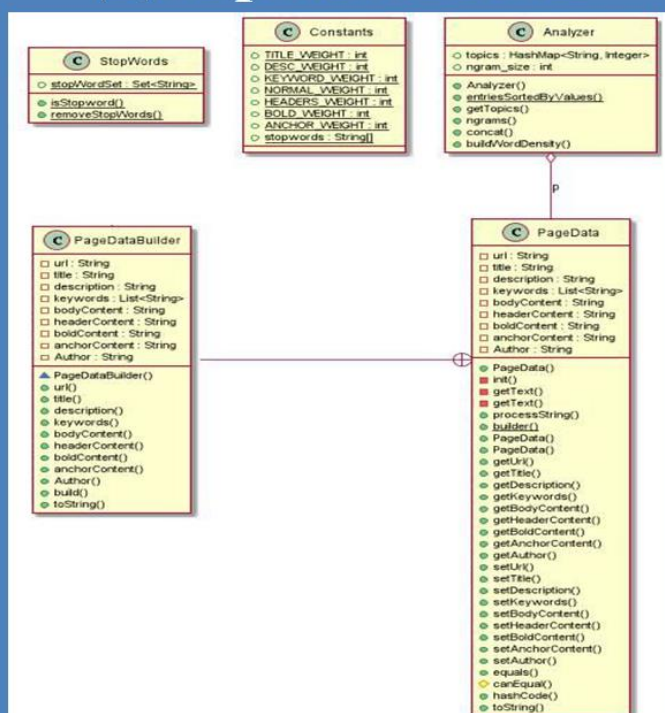
Вибір середовища розробки

Критерій порівняння середовищ розробки	Середовища розробки		
	IntelliJ IDEA	NetBeans	Apache JDeveloper
Інтелектуальний аналіз і підказки коду	+	+—	+—
Підтримка рефакторингу	+	+—	+—
Інтеграція з системами збірки та залежностей	+	+	+—
Продуктивність у складних багатомодульних проектах	+	+—	—
Зручність налагодження та тестування	+	+	+—

6

Рисунок В.6 – Слайд 6

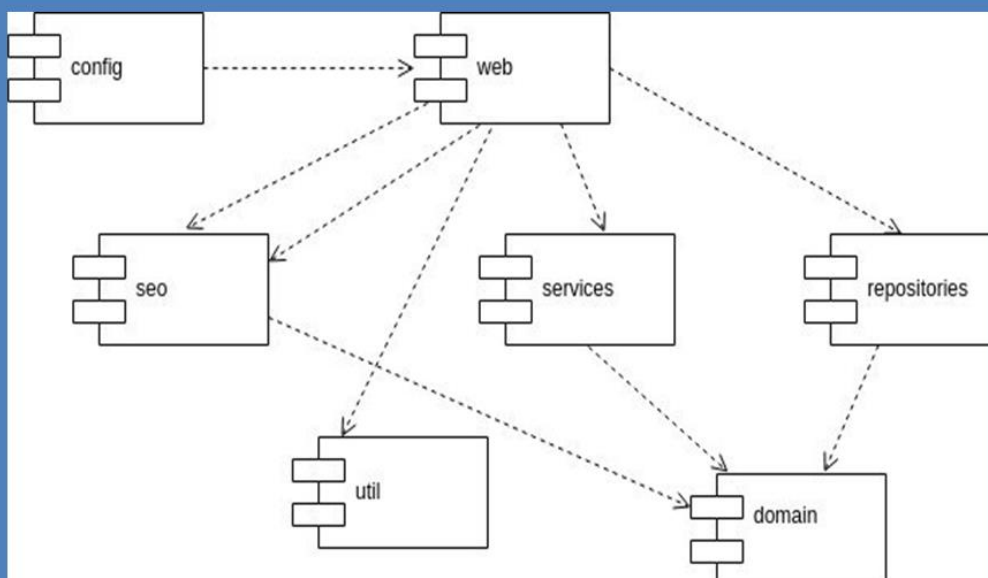
Діаграма класів



7

Рисунок В.7 – Слайд 7

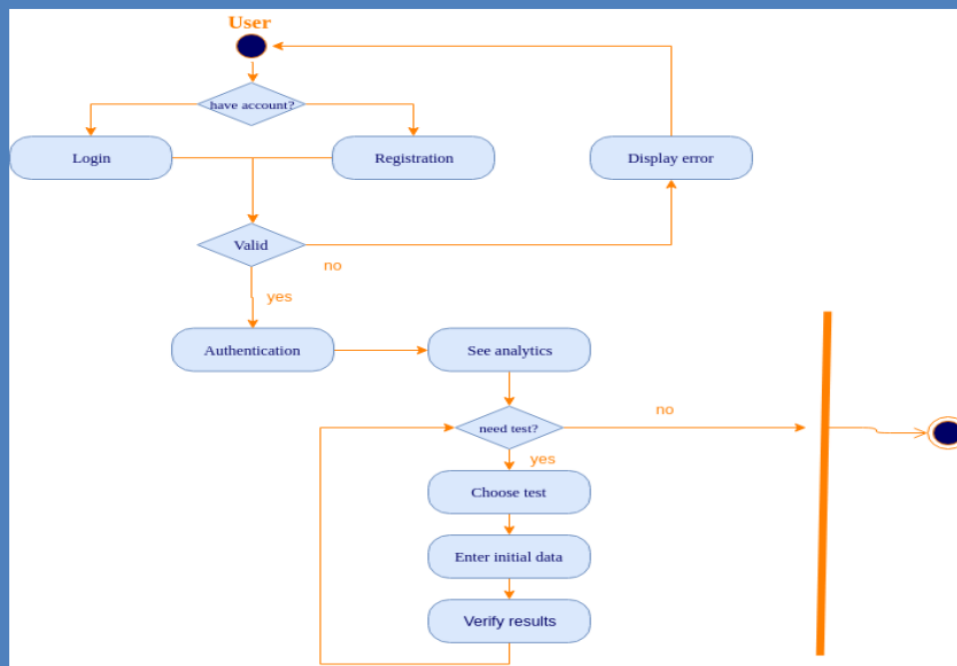
Діаграма пакетів



8

Рисунок В.8 – Слайд 8

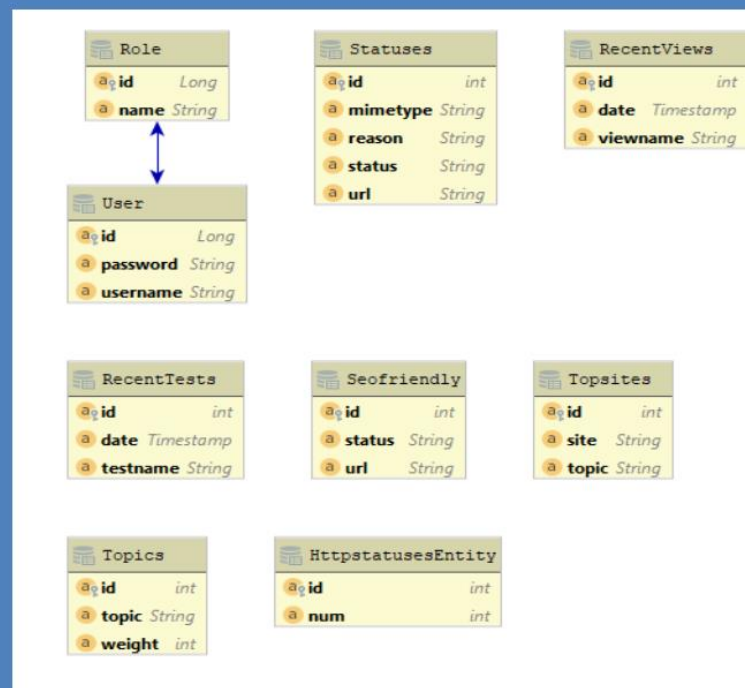
Схема функціонування програми



9

Рисунок В.9 – Слайд 9

Схема бази даних



10

Рисунок В.10 – Слайд 10

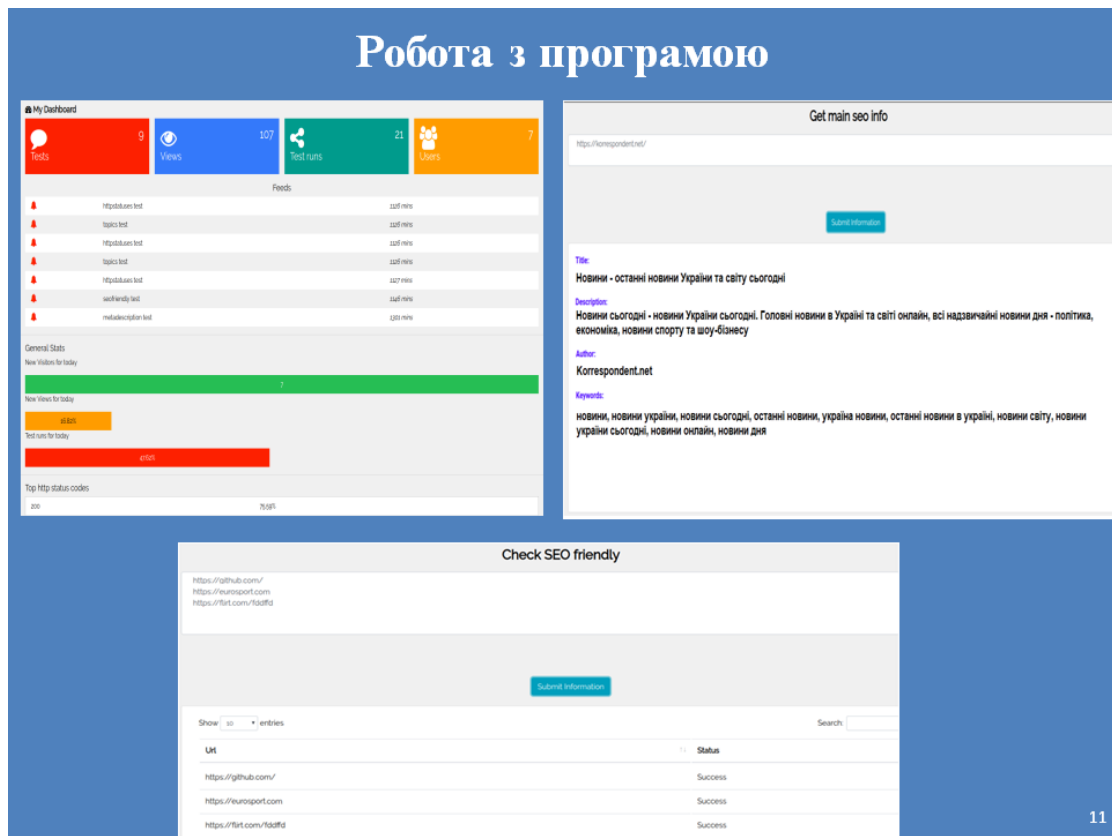


Рисунок В.11 – Слайд 11

Висновки

В результаті виконання кваліфікаційної роботи бакалавра було проаналізовано та досліджено процес розробки програмного забезпечення для аналізу вебсайтів.

Розроблено програмне забезпечення для аналізу вебсайтів. Для реалізації програмного забезпечення для аналізу вебсайтів обрано мову програмування Java та середовище розробки IntelliJ IDEA. Запропоновано структуру програмного забезпечення для аналізу вебсайтів. Розроблено базу даних програмного забезпечення для аналізу вебсайтів.

Результати перевірки розробленого програмного забезпечення підтвердили його працездатність.

Усі завдання кваліфікаційної роботи бакалавра повністю виконано.

Рисунок В.12 – Слайд 12