МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

ДОПУЩЕНО ДО ЗАХИСТУ
Завідувач кафедри інтелектуальних
інформаційних систем, д-р техн. наук, професор
_____Ю. П. Кондратенко
«____»_____2023 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

«ІНТЕЛЕКТУАЛЬНА СИСТЕМА ЗБОРУ ДАНИХ ТА АНАЛІЗУ ПОВЕДІНКИ КОРИСТУВАЧА У WEB - СЕРЕДОВИЩІ»

Спеціальність 122 «Комп'ютерні науки»

122 – БКР – 401.22020101

Виконала студентка 4-го курсу, групи 401
_____ К. О. Бурім
« » червня 2023 р.

Керівник: д-р техн. наук, доцент
_____О. В. Козлов
« » червня 2023 р.

Миколаїв – 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет ім. Петра Могили
Факультет комп'ютерних наук
Кафедра інтелектуальних інформаційних систем

Рівень вищої освіти **бакалавр**
Спеціальність **122 «Комп'ютерні науки»**
(шифр і назва)
Галузь знань **12 «Інформаційні технології»**
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інтелектуальних інформаційних систем, д-р техн. наук, проф.
_____ Ю. П. Кондратенко
«____» _____20___р.

З А В Д А Н Н Я

на виконання кваліфікаційної роботи

Видано студентці групи 401 факультету комп'ютерних наук Бурім Катерині Олександрівні.

1. Тема кваліфікаційної роботи «Інтелектуальна система збору даних та аналізу поведінки користувача у web-середовищі».

Керівник роботи Козлов Олексій Валерійович, д-р техн. наук, доцент.

Затв. наказом Ректора ЧНУ ім. Петра Могили від «___» _____ 20___ р. № ____

2. Строк представлення кваліфікаційної роботи студентом «___» _____ 20__ р.

3. Вхідні (початкові) дані до роботи: методи та технології збору даних та аналізу поведінки користувача у вебсередовищі.

Очікуваний результат: інтелектуальна система збору даних та аналізу поведінки користувача у web-середовищі.

4. Перелік питань, що підлягають розробці (зміст пояснювальної записки):

- аналіз наявних аналогів;
- аналіз метрик необхідних для збору даних та аналізу поведінки користувача у вебсередовищі;

- вибір інструментів для реалізації системи;
- проєктування інтелектуальної системи;
- опис програмної реалізації.

5. Перелік графічного матеріалу: презентація.

6. Завдання до спеціальної частини: «Аналіз робочого місця розробника програмного забезпечення з точки зору техніки безпеки».

7. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис
Спеціальна частина з охорони праці	Алексєєва А. О., канд. техн. наук, доцент	

Керівник роботи д-р техн. наук, доцент Козлов О. В.
(наук. ступінь, вчене звання, прізвище та ініціали)

_____
(підпис)

Завдання прийнято до виконання Бурім К. О.
(прізвище та ініціали)

_____
(підпис)

Дата видачі завдання « 23 » листопада 2022 р.

КАЛЕНДАРНИЙ ПЛАН
виконання кваліфікаційної роботи

Тема: Інтелектуальна система збору даних та аналізу поведінки користувача у web-середовищі.

№	Найменування роботи	Початок	Закінчення	Примітки
1	Подання заявки та затвердження теми БКР	27.10.2022	27.10.2022	Виконано
2	Отримання завдання на виконання БКР	22.11.2022	22.11.2022	Виконано
3	Складання календарного плану роботи на весь період виконання БКР	06.12.2022	09.12.2022	Виконано
4	Отримання завдання на переддипломну практику	18.04.2023	18.04.2023	Виконано
5	Проходження переддипломної практики, збір та аналіз матеріалів до БКР	01.05.2023	14.05.2023	Виконано
6	Розробка звіту з переддипломної практики	15.05.2023	17.05.2023	Виконано
7	Виконання БКР: аналіз наявних аналогів, аналіз метрик необхідних для збору даних та аналізу, вибір інструментів для реалізації системи, проєктування інтелектуальної системи, опис програмної реалізації	15.05.2023	19.06.2023	Виконано
8	Попередній захист БКР на засіданні комісії кафедри	29.05.2023	30.05.2023	Виконано
9	Доробка та остаточне оформлення БКР	02.06.2023	19.06.2023	Виконано
10	Подання БКР рецензенту	15.06.2023	17.06.2023	Виконано

11	Подання БКР, її електронної копії та інших документів (відгуку, рецензії) до захисту	19.06.2023	22.06.2023	Виконано
12	Захист БКР перед екзаменаційною комісією (ЕК)	26.06.2023	29.06.2023	Виконано

Розробила студентка Бурім К. О. _____
(прізвище, ім'я, по батькові студента) (*підпис*)

Керівник роботи д-р техн. наук, доцент Козлов О. В. _____
(*посада, прізвище, ім'я, по батькові)* *(підпис)*

« 9 » грудня 2022 р.

АНОТАЦІЯ

бакалаврської кваліфікаційної роботи студентки групи 401 ЧНУ ім. Петра Могили

Бурім Катерини Олександрівни

Тема: «Інтелектуальна система збору даних та аналізу поведінки користувача у web-середовищі»

Кваліфікаційна робота бакалавра присвячена створенню інтелектуальної системи збору даних та аналізу поведінки користувача у вебсередовищі. Дана робота є актуальною через зростання і розвиток обсягу даних, які збираються від користувачів у мережі величезними темпами. Інтелектуальні системи збору даних та аналізу поведінки користувачів можуть забезпечити більш ефективний та точний аналіз поведінки користувачів, що дасть можливість точніше визначити їхні потреби та вимоги.

Об’єкт дослідження – процеси збору даних та інтелектуального аналізу поведінки користувачів у вебсередовищі.

Предметом дослідження – технології та програмні засоби автоматизованого збору даних та аналізу поведінки користувача у вебсередовищі.

Мета дослідження – підвищення ефективності процесів збору, аналізу та використання даних про поведінку користувачів у вебсередовищі шляхом розробки інтелектуального алгоритму.

У першому розділі було розглянуто наявні аналоги та публікацій та актуальність предметної сфери. У другому розділі продемонстровано методи для розв’язання задачі та технології розробки системи. У третьому розділі спроєктовано інтелектуальну систему та описано програмну реалізацію.

В результаті виконаної роботи було реалізовано інтелектуальну систему збору та аналізу даних поведінки користувача у вебсередовищі.

БКР викладена на 64 с. (без додатків), містить 3 розділи, 33 рис., 2 табл., 1 додаток, 18 джерел в переліку посилань.

***Ключові слова**: аналіз вебсайтів, аналітика, збір даних, аналіз даних, вебкористувач.*

ABSTRACT

of the Bachelor's Thesis

by student of group 401 in Petro Mohyla Black Sea National University

Kateryna Burim

"An intelligent system of data collection and analysis of user behavior in the web environment"

The bachelor's qualification work is devoted to the creation of an intelligent system of data collection and analysis of user behavior in the web environment. This work is relevant due to the growth and development of the amount of data that is collected from users in the network at a huge rate. Intelligent data collection and user behavior analysis systems can provide a more efficient and accurate analysis of user behavior, which will enable more accurate determination of their needs and requirements. Also, the collection and analysis of user behavior can be used to identify problems in the web environment and correct them in time.

The object of research is the processes of data collection and intellectual analysis of user behavior in the web environment.

The subject of the research is technologies and software tools for automated data collection and analysis of user behavior in the web environment.

The purpose of the research is to improve the efficiency of the processes of collecting, analyzing and using data on user behavior in the web environment through the development of intelligent algorithms.

In the first chapter, existing analogues and publications and the relevance of the subject area were considered. The second chapter demonstrates the methods for solving the problem and the system development technology. In the third chapter, the intelligent system is designed and the software implementation is described.

As a result of the work performed, an intelligent system of data collection and analysis of user behavior in the web environment was implemented.

The Bachelor's Thesis contains 64 p. (without appendices), contains 3 chapters, 31 figures, 2 tables, 1 appendix, 18 sources in the list of references.

Keywords: *website analysis, analytics, data collection, data analysis, web user*.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

HTTP – Hyper Text Transfer Protocol

KPI – Key Performance Indicators

URL – Uniform Resource Locator

JSON – JavaScript Object Notation

XML – Extensible Markup Language

DNT – Do Not Track

IE – Internet Explorer

VPN – Virtual Private Network

ПЗ – програмне забезпечення

ВСТУП

Щодня кількість користувачів Інтернету зростає, а також збільшується кількість даних, які збираються про їх поведінку. Згідно зі звітом [1] "Digital 2022 Global Overview" від We Are Social та Hootsuite, станом на січень 2022 року, на планеті існує більше ніж 6 мільярдів активних користувачів Інтернету. Це є причиною зростання кількості даних, які збираються щодо поведінки користувачів у мережі.

Згідно зі звітом [2] "Data Never Sleeps 8.0" від Domo, за 2021 рік щодня в Інтернеті:

– було створено 4,5 мільйона відео на YouTube;

– було опубліковано 500 мільйонів твітів на Twitter;

– було здійснено понад 4,7 мільярда пошукових запитів на Google;

– Facebook зареєстрував понад 1,9 мільярда активних користувачів.

Ці статистичні дані підтверджують, що обсяг даних, які збираються від користувачів у мережі, зростає і розвивається величезними темпами. Інтелектуальні системи збору даних та аналізу поведінки користувачів можуть забезпечити більш ефективний та точний аналіз поведінки користувачів, що дасть можливість точніше визначити їхні потреби та вимоги. Також, збір та аналіз поведінки користувачів може бути використаний для виявлення проблем в роботі вебсередовища та їх вчасного виправлення. Наприклад, якщо виявити, що користувачі не використовують певні функції вебсайту, це може бути сигналом для того, що їх потрібно поліпшити або переробити.

Отже, інтелектуальна система збору даних та аналізу поведінки користувачів у вебсередовищі є дуже актуальною та може бути корисною для розробки більш ефективних та орієнтованих вебсайтів та додатків.

Метою дослідження є підвищення ефективності процесів збору, аналізу та використання даних про поведінку користувачів у вебсередовищі шляхом розробки інтелектуального алгоритму.

Для досягнення визначеної мети необхідно вирішити такі завдання:

– провести аналіз в потребі дослідження поведінки користувача у вебсередовищі;

– провести аналіз технічних можливостей та інструментів для збору даних користувачів у вебсередовищі;

– визначити параметри та метрики, які необхідно збирати для аналізу поведінки користувачів;

– розробити алгоритм для аналізу поведінки користувачів, використовуючи зібрані дані;

– провести експериментальну перевірку розробленої системи.

Об'єктом дослідження є процеси збору даних та інтелектуального аналізу поведінки користувачів у вебсередовищі.

Предметом дослідження є технології та програмні засоби автоматизованого збору даних та аналізу поведінки користувача у вебсередовищі.

Аналізуючи сучасний стан проблеми, можна зазначити, що існують певні розробки у галузі збору та аналізу даних користувачів у вебсередовищі, але більшість з них не є достатньо ефективними або мають обмеження у використанні.

Провідні фірми у даній галузі включають: Google, Facebook, Amazon, Microsoft, IBM, Oracle, SAS та інші. Вчені та спеціалісти, які працюють у даній галузі, активно досліджують і розробляють нові методи та алгоритми для аналізу поведінки користувачів у вебсередовищі. Провідні фірми та вчені у даній галузі, активно працюють над розробкою та удосконаленням інтелектуальних систем збору даних та аналізу поведінки користувача у вебсередовищі, що свідчить про потребу у розв’язанні цієї проблеми.

Результати дослідження можуть знайти застосування в різних галузях, пов'язаних з вебресурсами та інтернет-технологіями. Наприклад, вони можуть бути використані в маркетингових дослідженнях та аналізі вебтрафіку для поліпшення конверсії сайту, розробці інтернет-реклами та збільшення прибутку компаній, що працюють в онлайн-бізнесі. Крім того, інтелектуальна система збору даних та

аналізу поведінки користувача може бути корисною для оптимізації роботи вебсервісів та покращення користувацького досвіду в інтернеті. У сфері наукових досліджень інтелектуальна система може бути використана для аналізу поведінки користувачів на вебсайтах, що дозволить покращити розуміння їхньої поведінки та реакції на різні вебресурси.

1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ. ОГЛЯД НАЯВНИХ РІШЕНЬ. ПОСТАНОВКА ЗАДАЧІ

1.1 Розкриття об’єкта і предмету дослідження

Аналітика вебсайтів описує процес збору, вимірювання, аналізу та звітності даних, що стосуються вебсайтів та їх використання. Від початку створення Інтернету, аналітика вебсайтів розвивалась із простого протоколу передачі гіпертексту до складних наборів даних для відстеження та аналізу. Зараз існує безліч інструментів, платформ та робочих місць для дослідження аналітики вебсайтів та електронної комерції. Прогнозувалося, що у 2014 році ринок аналітики вебсайтів досягне 1 млрд доларів з щорічним темпом зростання понад 15% [3].

Технології аналітики вебсайтів можуть бути класифіковані як аналітика вебсайтів на сайті та за його межами. На сайті аналітика дозволяє збирати дані про кількість відвідувань, час відвідування та інші метрики, які відображають пряму взаємодію з сайтом. За межами сайту аналітика включає дані з інших джерел, таких як опитування, звіти про ринок та публічна інформація. Такий підхід дозволяє використовувати широкий спектр даних для аналізу та прийняття рішень.

За історичними даними, після запуску World Wide Web у 1993 році та створення першого популярного браузера Mosaic, для відстеження вебзапитів використовувались log files журналів. У 1993 році компанія WebTrends з міста Портленд, штат Орегон, стала однією з перших, яка використовувала дані з журналів вебсервера для проведення аналізу вебсайту. У тому ж році, WebTrends створив перше програмне забезпечення для аналізу комерційних вебсайтів. У 1995 році, доктор Стівен Тернер створив безкоштовне програмне забезпечення для аналізу файлів журналів, яке отримало назву Analog. В 1996 році, компанія WebSideStory запропонувала лічильник хітів як службу для вебсайтів, які були відображені на банері.

Однак, журнали вебсерверів мають певні обмеження у типах зібраних даних. Наприклад, вони не можуть надати інформацію про розміри екрана відвідувачів,

взаємодію користувачів із елементами сторінки, події для миші, такі як натискання та навігація і т.д. Зараз, завдяки сучасним технологіям, використання тегів сторінок дозволяє подолати ці обмеження.

Основна мета аналітики вебсайтів полягає в зборі та аналізі даних про вебтрафік та використання різноманітних моделей. Розмірна модель є найпоширенішим методом для вивчення цих даних [4]. За цією моделлю існує два основних типи даних: факти, які є вимірювальними даними та описують факти з різних аспектів і рівнів, та габаритні дані, які є значно складнішими. Дані фактів в основному стосуються кількості та часу використання, наприклад, переглядів сторінок та кількості дій користувача, таких як клацання мишею. Різні показники розраховуються на основі цих основних вимірів та розмірів.

Існує декілька основних типів параметрів, які включають різноманітні аспекти, такі як час, вміст, розташування та інформація про користувача клієнта. Інформація про користувача може включати дані про операційну систему, тип браузера, розмір екрана та інше, а також дані про сеанс.

Ці дані можуть бути зібрані з різних джерел, які можна умовно розділити на наступні 4 типи:

– пряма інформація, отримана з запиту HTTP;
– дані, які передаються на рівні додатків у межах HTTP-запитів;
– мережеві дані та дані сервера, що генеруються відповідно до HTTP-запитів;
– зовнішні дані.

Дослідження розкриває значний взаємозв’язок між часом завантаження сторінки і ймовірністю конверсії користувача [5]. Аналітика вебсайтів допомагає вирішити цю проблему, використовуючи показники завантаження сторінки, такі як середній час завантаження через вебпереглядач та географічне розташування, для вимірювання ефективності. Аналіз ефективності в режимі реального часу дозволяє активно виявляти, досліджувати та діагностувати проблеми продуктивності. Застосування поліпшень може варіюватися від простої оптимізації зображень до

зміни термінів придатності, вказаних у заголовках HTTP, для того, щоб спонукати браузери використовувати кешований контент вебсайту. Використання техніки "карти тепла" може допомогти виявити помилки на вебсайті, такі як натискання кнопок або зображень без посилань. Ті самі підходи можуть бути використані розробниками вебдодатків та ігор для додавання або зміни функціональності програмного забезпечення.

Аналітика вебсайтів – процес збору, аналізу та інтерпретації даних про вебсайти з метою оптимізації їх ефективності. Вона включає в себе вимірювання даних, аналіз цих даних для виявлення тенденцій та можливостей, а також впровадження оптимізаційних заходів для покращення вебсайту. KPI – метрика, яка використовується для оцінки успішності вебсайту. Прикладами KPI можуть бути кількість відвідувачів, переглянутих сторінок, часу перебування на сайті та конверсії [6]. Аналітика вебсайтів є потужним інструментом для зрозуміння та оптимізації вебсайтів. Вона допомагає збирати та аналізувати дані для виявлення проблем та можливостей для покращення ефективності вебсайту.

HTTP не має здатності надавати інформацію про сеанс, оскільки воно само по собі не є суб'єктом. Тому дані про сеанс керуються на рівні програми. Зазвичай дані сеансу передаються як параметри URL або у вигляді файлів cookie. Ці дані є важливими для розрахунку різних показників, таких як кількість відвідувань, час на сайті, кількість переглянутих сторінок на відвідування та інші.

Дані з додатків зазвичай включаються в HTTP-запити і можуть бути збережені на трьох різних рівнях. По-перше, ці дані можна додати до URL-адреси як параметри. Серверні програми можуть аналізувати ці параметри для отримання додаткової інформації. Наприклад, Google використовує спеціальні URL-адреси у своїх пошукових результатах, щоб перенаправити користувачів на цільові сторінки з додатковою інформацією. По-друге, дані додатків можуть бути передані як HTTP-заголовок файлу cookie. Файли cookie - це невеликі текстові файли, які зазвичай зберігають профіль користувача та інформацію про його активність. Тип даних, які можна зберігати в файлах cookie, залежить від клієнтського програмного

забезпечення та його налаштувань [7]. По-третє, дані додатків також можуть бути включені в тіло HTTP-запиту при використанні методу "POST", який часто використовується для передачі даних форм. Таким чином, дані про сеанс та дані з додатків включаються в HTTP-запити на різних рівнях і можуть бути використані для отримання додаткової інформації та аналізу її на сервері.

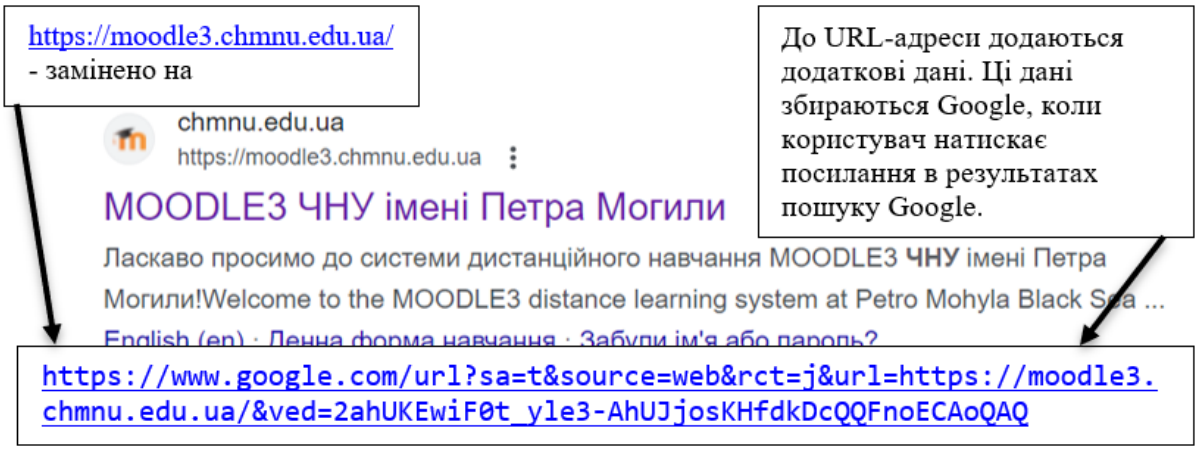

Рисунок 1.1 – Зібрані URL-адреси Google у результатах пошуку

Сторонні постачальники послуг, такі як Google Analytics і Open Web Analytics, використовують метод відстеження, відомий як JavaScript-відстеження. Журнал вебсерверів менш шкідливий і не потребує модифікації [8]. В порівнянні з методом реєстрації вебсервера, маркування сторінок має багато переваг [9]. Основна ідея полягає в тому, щоб вбудувати JavaScript-код на вебсторінки, який буде збирати і передавати дані про взаємодію користувачів з вебсайтом. Сторонній постачальник послуг, такий як Google Analytics, надає унікальний JavaScript-код, який потрібно вставити на всі сторінки вебсайту. Коли користувач відвідує вебсайт, вбудований JavaScript-код виконується в його браузері. Цей код збирає різноманітні дані про взаємодію користувача з вебсторінками, наприклад, переглянуті сторінки, кліки на посилання, дії на формах тощо. Зібрані дані форматуються в певну структуру, наприклад, JSON або XML, і передаються на сервер стороннього постачальника послуг. Для цього використовується HTTP-запит, який включає дані в тілі запиту. Іноді дані можуть бути передані

безпосередньо через URL-параметри. Сервер стороннього постачальника послуг отримує дані і виконує їх обробку. Це може включати збереження даних у базі даних, агрегування і аналіз даних для отримання різноманітних метрик і відомостей про поведінку користувачів. Після обробки даних сторонні постачальники послуг зазвичай надають розширені звіти і аналітичні інструменти. Ці звіти можуть включати інформацію про кількість відвідувачів, джерела трафіку, популярність сторінок, час перебування на сторінках, конверсії та багато іншого. Додатково, сторонні постачальники послуг можуть надати можливості налаштування цілей і трекінгу подій. Це дозволяє вам визначити конкретні дії, які ви хочете відстежувати, такі як натискання на кнопки, заповнення форм або завершення покупок. Ці дані допомагають зрозуміти ефективність вебсайту і вплинути на його поліпшення.

Таким чином, метод JavaScript-відстеження, що використовується сторонніми постачальниками послуг, дозволяє збирати, передавати, обробляти і аналізувати дані про взаємодію користувачів з вебсайтом. Цей метод допомагає вебвласникам здійснювати моніторинг та аналізувати ефективність своїх вебсайтів, а також зробити дальші поліпшення на основі отриманих даних.

Зацікавленість відвідувачів є однією з найбільш важливих метрик, оскільки вона вказує на рівень залученості і сприйняття вебсайту користувачами. Чим більше сторінок відвідує користувач, тим більша його зацікавленість. Аналізуючи середнє число переглянутих сторінок на відвідувача, можна отримати загальну картину зацікавленості на вебсайті. Довгий час перебування на сторінках може свідчити про активну взаємодію та зацікавленість відвідувачів. Аналіз тривалості перебування на сторінках може дати уявлення про те, як добре контент привертає увагу користувачів. Взаємодія з елементами вебсторінки, натискання на посилання, заповнення форм, перегляд відео, коментування тощо, чим більше активності в цих взаємодіях, тим більше зацікавленість відвідувачів. Конверсії і досягнення цілей можуть слугувати ознакою високої зацікавленості відвідувачів. Наприклад, якщо велика кількість відвідувачів реєструється на сайті або здійснює покупки, це може

свідчити про їх зацікавленість і залученість. Взаємодія зі змістом оцінюється шляхом аналізу коментарів, оцінок, відгуків або рейтингів, які залишають відвідувачі на вебсайті. Позитивна взаємодія і високі оцінки можуть вказувати на зацікавленість і задоволення користувачів зі змісту, що сприяє залученості [10].

Розглянемо схему, яка відображає модель онлайн поведінки споживачів (рис. 1.2) [11]. У цій моделі особливості користувача включають характеристики кожного відвідувача вебсайту, такі як рівень освіти, інтереси, рівень довіри тощо. Модель також враховує структури середовища, такі як культура, вплив засобів масової інформації та соціального середовища. Інший фактор, що впливає на поведінку користувача, - це довіра до продукту та його характеристик. У цьому випадку рекламні характеристики включають особливості вебсайту, такі як його легкість використання та безпека онлайн-платежів. Комерційні характеристики включають якість електронного сервісу, рівень безпеки та надійності, а також репутацію бренду.

Рисунок 1.2 – Модель онлайн поведінки споживачів

1.2 Огляд та аналіз наявних аналогів та публікацій

Аналіз Clickstream є важливим технічним рішенням для розуміння поведінки користувачів на вебсайтах або в додатках. Він включає в себе збір, обробку та аналіз даних про послідовність кліків користувача під час їх взаємодії з вебсторінками або додатками [12]. Збір даних може бути досягнуто за допомогою вбудованих інструментів аналітики вебсайтів, які відстежують події, такі як кліки, переходи, заповнення форм і т. д. Програми збору даних, такі як Google Analytics, можуть надати багато корисних метрик та звітів. Після збору Clickstream даних, їх можна обробити та зберегти для подальшого аналізу. Це може включати зберігання даних в базі даних, використання стрімінгових систем для обробки в реальному часі або використання хмарних платформ для масштабованої обробки. Візуалізація Clickstream даних допомагає відображати взаємодію користувачів з вебсайтом або додатком у зрозумілій формі. Це дозволяє виявити патерни, тренди та незвичайну активність. Застосування аналітичних методів, таких як класифікація, кластеризація, асоціативні правила та прогнозування, можуть допомогти розкрити важливу інформацію з Clickstream даних. Наприклад, аналіз Clickstream може розкрити найчастіші шляхи навігації користувачів, популярні сторінки або продукти, які привертають найбільше уваги, або незвичайні поведінкові шаблони, які можуть вказувати на шахрайську або аномальну активність.

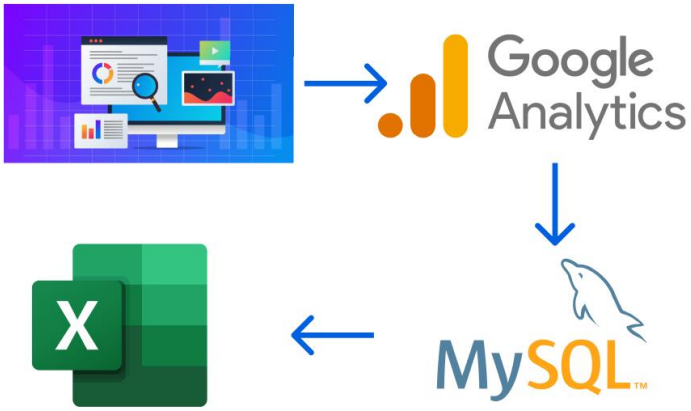

Рисунок 1.3 – Схема процесу збору, агрегації і візуалізації даних

Аналітика вебсайтів насправді залежить від використання файлів cookie для збору та передачі даних на сервер. Файли cookie – це невеликі текові файли, які зберігаються на комп'ютері користувача під час взаємодії з вебсайтом. Вони виконують кілька функцій, включаючи зберігання інформації про користувача, стан сесії, налаштування вебсайту та інші дані, які допомагають забезпечити персоналізований та зручний досвід відвідувачів. У контексті аналітики вебсайтів, файли cookie використовуються для збору інформації про поведінку користувача на вебсайті. Коли користувач переходить на вебсайт, вебаналітичний код, який зазвичай вбудовується на сторінках сайту, створює та встановлює унікальний ідентифікатор у вигляді файлу cookie на комп'ютері користувача. Цей ідентифікатор використовується для збору історії взаємодії користувача зі сторінками вебсайту.

Якщо файли cookie заблоковані на боці клієнта, це загрожує відсутністю частини інформації, що може впливати на точність збору вебтрафіку. Для захисту конфіденційності користувачі є кілька методів керування налаштуваннями клієнтських програм. Усі основні сучасні браузери надають прості засоби для видалення файлів cookie та блокування сторонніх файлів cookie та скриптів. Крім того, користувачі можуть скористатись приватним режимом перегляду, який доступний у браузерах, таких як Google Chrome (режим Incognito), Internet Explorer (режим InPrivate) та Firefox (приватний браузер).

Для стандартизації приватності та контролю за відстежуванням, W3C рекомендує використовувати заголовок HTTP DNT. Заголовок DNT є налаштуванням, яке користувач може встановити в своєму браузері. Якщо DNT установлено в true, то сервер та клієнтський скрипт можуть зчитувати це і не відстежувати користувача. Однак це не зобов'язує вебсайти дотримуватись цього налаштування, і провайдер послуг може вирішити не враховувати вибір користувачів. Наприклад, коли Microsoft встановила значення DNT в «true» за замовчуванням у браузері Internet Explorer 10, Yahoo оголосила, що вони не будуть враховувати налаштування DNT в IE 10 [13].

Web2.0 приніс значні зміни в галузь аналітики вебсайтів, перетворивши її підхід та можливості. Він відкрив можливості для активної участі користувачів, включаючи коментування, рейтинги, відгуки та спільне створення вмісту. Це призвело до збільшення кількості даних, що можна аналізувати, і створило потребу у нових методах та інструментах аналітики вебсайтів для виявлення та розуміння впливу користувацької взаємодії на вебсайти, а не лише на основі HTTP-запитів.

Мобільний Інтернет став не менш впливовим за останні п’ять років [14], але для вимірювання мобільного вебдоступу існую свою проблеми [15]. Наявність різних мобільних пристроїв з різними розмірами екранів, операційними системами та браузерами ускладнює вимірювання мобільного трафіку. Вимірювальні інструменти повинні враховувати цю різноманітність для точного визначення характеристик мобільного вебдоступу. Іноді складно відрізнити мобільні пристрої від настільних комп'ютерів або інших пристроїв, особливо при наявності проксі-серверів або VPN. Вимірювальні системи повинні мати ефективні методи виявлення мобільних пристроїв для точного вимірювання мобільного вебтрафіку. Мобільні пристрої часто працюють в змінних умовах мережі, таких як слабий сигнал, переривання з'єднання або обмежена пропускна здатність. Це може впливати на швидкість завантаження сторінок, вимірювання часу перебування та інші метрики, що стосуються мобільного вебдоступу.

Деякі експерти та фахівці з аналітики вебсайтів перейшли від використання терміну "web analytics" до терміну "digital analytics". Це сталося відображенням змін в сфері технологій та вимог до аналізу даних. Одна з причин переходу до терміну "digital analytics" полягає в тому, що вебсайти не є єдиними каналами взаємодії з користувачами. З появою соціальних медіа, мобільних додатків, електронної пошти та інших цифрових каналів, аналітика розширилася на вимірювання та аналіз активності користувачів не тільки на вебсайтах, але й в усьому цифровому просторі. Таким чином, термін "digital analytics" відображає ширшиймспектр джерел даних та охоплює аналіз активності на всіх цифрових платформах [16]. Крім того, термін "digital analytics" відображає перехід від

простого вимірювання вебтрафіку до глибшого аналізу даних. Сучасні аналітичні інструменти дозволяють вимірювати не лише кількість відвідувань та переглядів сторінок, але й проводити сегментацію аудиторії, аналізувати поведінку користувачів, вимірювати ефективність маркетингових кампаній та багато іншого. Термін "digital analytics" відображає цей ширший спектр можливостей аналізу даних в цифровому.

Аналітика вебсайтів – це процес збору, вимірювання, аналізу та інтерпретації даних про активність користувачів на вебсайтах та інших цифрових платформах. Вона допомагає розуміти поведінку користувачів, оцінювати ефективність вебстратегій та приймати обґрунтовані рішення для поліпшення досвіду користувачів та досягнення бізнес-цілей. Web2.0 приніс значні зміни у сферу аналітики вебсайтів, перетворивши її підхід та можливості. Він забезпечив активну участь користувачів через коментування, рейтинги, спільне створення вмісту та соціальні медіа. Це призвело до збільшення обсягу даних, що можна аналізувати, і створило потребу у нових методах та інструментах аналітики вебсайтів. середовищі.

1.3 Постановка задачі

В даній роботі необхідно розробити систему аналітики вебсайтів, яка дозволить відстежувати активність користувачів на вебсторінці. Система буде збирати дані про відвідувачів, такі як ім'я користувача, час входу, час виходу, тривалість перебування на сторінці, кількість кліків та країну користувача за допомогою API GeoIP. Отримані дані будуть зберігатися в базі даних для подальшого аналізу та використання.

Технічні пропозиції:

– використання мови програмування PHP для реалізації серверної частини;

– використання бази даних MySQL для зберігання інформації про активність користувачів;

– використання API GeoIP (наприклад, IP-API.com) для визначення країни користувача за його IP-адресою;

– вебінтерфейс для відображення активності користувачів, їх часу перебування на сторінці та кількості кліків.

Технічне завдання:

– реалізувати серверну частину на PHP для зберігання та обробки даних активності користувачів;

– використовувати базу даних MySQL для зберігання інформації про активність користувачів;

– використати API GeoIP для отримання країни користувача на основі його IP-адреси;

– розробити вебінтерфейс для відображення активності користувачів, часу перебування на сторінці та кількості кліків;

– забезпечити безпеку даних, використовуючи захисні механізми, такі як захист від SQL-ін'єкцій і XSS-атак;

– забезпечити можливість розширення системи, додавання нових функцій та покращень в майбутньому.

Специфікація проєкту ПЗ:

1) функціональні вимоги:

– збір та збереження інформації про активність користувачів, включаючи ім'я користувача, час входу, час виходу, тривалість перебування на сторінці та кількість кліків;

– використання API GeoIP для визначення країни користувача на основі його IP-адреси;

– збереження країни користувача разом з іншою інформацією про активність у базі даних;

– відображення активності користувачів у вебінтерфейсі, включаючи час перебування на сторінці та кількість кліків.

2) нефункціональні вимоги:

– забезпечити захист від SQL-ін'єкцій і XSS-атак шляхом належної обробки та екранування вхідних даних перед їх використанням у SQL-запитах та вебсторінках;

– забезпечити оптимальну продуктивність системи, мінімізуючи запізнення в збереженні даних та відображенні активності користувачів;

– розробити систему з урахуванням можливості додавання нових функцій та покращень в майбутньому, таких як відстежування інших параметрів активності або аналітичні звіти.

Висновки до розділу 1

В результаті написання першого розділу було розкрито об’єкт і предмет дослідження. Проведено аналіз технології аналітики вебсайтів, визначено її основну мету, типи параметрів. Розглянуто схему, яка відображає модель онлайн поведінки споживачів.

Було розглянуто важливі аспекти аналізу Clickstream, файли cookie, вплив Web2.0 та мобільного Інтернету, а також перехід від терміну "web analytics" до "digital analytics".

Було сформульовано основну мету бакалаврської кваліфікаційної роботи, а саме – розробку системи аналітики вебсайтів, яка здатна відстежувати активність користувачів на вебсторінці. Було визначено, що система буде збирати різні дані про відвідувачів, включаючи їх ім'я, час входу та виходу, тривалість перебування на сторінці, кількість кліків, а також країну користувача. Отримані дані будуть зберігатися в базі даних для подальшого аналізу та використання.

Технічні пропозиції були надані з метою реалізації системи. Зокрема, запропоновано використовувати мову програмування PHP для реалізації серверної частини, базу даних MySQL для зберігання інформації про активність користувачів, API GeoIP для визначення країни користувача та вебінтерфейс для відображення активності користувачів.

Технічне завдання визначає основні вимоги до розробки системи. Необхідно реалізувати серверну частину на PHP для зберігання та обробки даних активності користувачів, використовувати базу даних MySQL для зберігання інформації, розробити вебінтерфейс для відображення активності користувачів.

Специфікація проєкту ПЗ визначає функціональні та нефункціональні вимоги до системи. Функціональні вимоги охоплюють збір та збереження інформації про активність користувачів, визначення країни користувача, відображення активності у вебінтерфейсі. Нефункціональні вимоги стосуються безпеки даних, продуктивності системи та можливості розширення функціональності в майбутньому.

Усі ці визначені елементи допоможуть у подальшій розробці та реалізації системи аналітики вебсайтів, що дозволить відстежувати та аналізувати активність користувачів з метою поліпшення вебсайту та користувацького досвіду.

2 МОДЕЛІ, МЕТОДИ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ЗБОРУ ДАНИХ ТА АНАЛІЗУ ПОВЕДІНКИ КОРИСТУВАЧА У ВЕБСЕРЕДОВИЩІ

2.1 Методи для розв’язання задачі

Методи збору даних про інтернет-користувачів являють собою різні техніки та інструменти, які використовуються для отримання інформації про поведінку, переваги та характеристики користувачів під час їх взаємодії з інтернет-сервісами та вебсайтами. Ці методи дозволяють збирати дані, аналізувати їх і використовувати отриману інформацію для різних цілей, таких як персоналізація контенту, поліпшення досвіду користувача і оптимізація маркетингових стратегій.

Аналітика вебсайту є одним з найпоширеніших та найефективніших методів збору даних про інтернет-користувачів. Вона включає використання спеціальних інструментів та платформ для збору, аналізу та звітності про поведінку користувачів на вебсайті. Ці інструменти забезпечують цінну інформацію про трафік, демографічні характеристики, використання функцій сайту та інші метрики, які можуть бути використані для прийняття стратегічних рішень та оптимізації вебсайту.

Аналітика надає метрики, такі як кількість відвідувань, унікальні відвідувачі, сеанси та конверсії, які допомагають оцінити, наскільки успішно вебсайт приваблює та утримує користувачів. Це дозволяє маркетологам та власникам вебсайтів вимірювати ефективність своїх маркетингових кампаній та оптимізувати вебсайт для досягнення поставлених цілей. Аналітика вебсайтів надає інформацію про те, як користувачі взаємодіють із вебсайтом, на що вони звертають увагу, які сторінки відвідують та які дії роблять. Це дозволяє зрозуміти, як користувачі використовують вебсайт, і визначити області, які потребують покращення або оптимізації. Аналітика вебсайтів надає об'єктивні дані, на основі яких можна приймати рішення. Вона допомагає ідентифікувати проблеми, визначити успішні стратегії та провести A/B тестування для оптимізації вебсайту. Маркетологи

можуть використовувати ці дані для прийняття обґрунтованих рішень, які допоможуть покращити користувальницький досвід та досягти бізнес-цілей.

Існують різноманітні види доступної інформації про користувача, що є метриками, які демонструють перевагу аналітики вебсайтів. Показники можуть бути доволі примітивними, але вони можуть допомогти аналізу вебтрафіку та поліпшенню вебсайту. Згідно з даними Panalysis, ці метрики відносяться до однієї з чотирьох категорій: використання сайту, відвідувачі сайту, аналіз контенту сайту, і якості. У таблиці 2.1 наведено типи метрик, які є в цих категоріях.

Таблиця 2.1 – Класифікація метрик

Використання сайту	**Відвідувачі**	**Аналіз контенту сайту**	**Якість**
– кількість відвідувачів і сеансів; – скільки людей неодноразово відвідують сайт; – географічна інформація; – пошукова діяльність.	– які вебсайти спрямовують відвідувачів на ваш сайт; – пошукові терміни, за якими люди знайшли ваш сайт; – скільки людей розміщують закладки на сайті.	– найпопулярніші сторінки входу; – найпопулярніші сторінки; – найпопулярніші сторінки для сеансів перегляду однієї сторінки; – топ сторінок виходу; – топ шляхів по сайту; – ефективність ключового контенту.	– зламані сторінки або помилки сервера; – реакція відвідувача на помилки.

Тип і загальний відсоток метрик на пряму залежать від постачальників вебаналітики, група загальних метрик все ще має місце бути. У таблиці 2.2 наведено 8 найпоширеніших типів метрик, які вимірюють, хто є відвідувачем вебсайту, та його дії під час сесії, пов'язавши кожен із цих метрик з відповідною категорією.

Таблиця 2.2 – Характеристика загальних метрик

Метрика	Опис	Категорія
Тип відвідувача	Хто має доступ до вебсайту (повторний, унікальний тощо)	Використання сайту
Тривалість відвідування	Загальна кількість часу, який відвідувач проводить на вебсайті	Використання сайту
Демографія та системна статистика	Фізичне місцеперебування та інформація системи, яка використовується для доступу до Вебсайту	Використання сайту
Інформація про внутрішній пошук	Інформація про ключові слова та сторінки результатів, які переглядаються за допомогою пошукової системи, вбудованої у Вебсайт	Використання сайту
Шлях відвідувача	Маршрут, який відвідувач використовує для навігації вебсайтом	Аналіз контенту сайту
Найпопулярніші сторінки	Сторінки, які отримують найбільше трафіку	Аналіз контенту сайту
URL-адреса переходу та аналіз ключових слів	Які сайти спрямовують трафік на Вебсайт і які ключові слова використовують відвідувачі, щоб знайти Вебсайт	Відвідувачі
Помилки	Будь-які помилки, які виникли під час спроби отримати сторінку	Якість

Дана метрика є відношенням відсотка звернень, отриманих сайтом, до кількості осіб, які відвідали його.

Є два типи користувачів: ті, які вже відвідували сторінку, і ті, хто ні. Ця різниця складається з повторних та нових користувачів. Система має відстежувати кожного користувача, як унікального відвідувача. Унікальним є лише один

відвідувач, однак це зазвичай не так (рис. 2.1). Однак, цілком можливо щоб декілька користувачів відвідували сайт з одного пристрою. Більша частина аналітичних систем довіряють cookie файлам відстеження відвідувачів, однак, коли користувач їх вимикає та очищує кеш, він буде вважатися новим відвідувачем сайту. Тому краще покладатися на унікальні відвідування або сеанси.

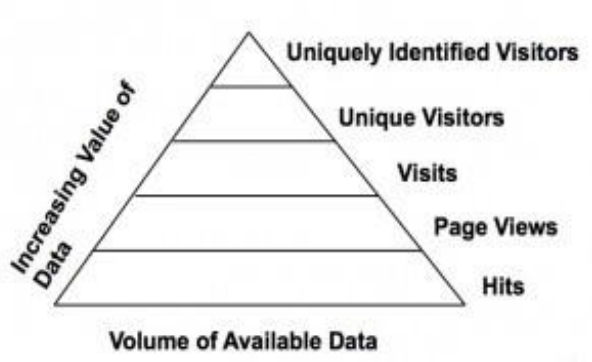

Рисунок 2.1 – Значення метрик

Збір даних про користувачів та їхню поведінку має вирішальне значення для ефективного керування вебсайтом. Вебсайти, які збирають та аналізують дані, мають значну перевагу перед тими, які цього не роблять.

Збір даних дозволяє отримати інформацію про те, що користувачі шукають, які проблеми намагаються вирішити, які продукти чи послуги їм цікаві. Це допомагає власникам вебсайтів налаштувати свої пропозиції та контент, щоб відповідати потребам користувачів. Зібрані дані про користувачів можуть бути використані для персоналізації власного досвіду. Наприклад, на основі переваг та попередньої поведінки користувачів можна запропонувати рекомендації, релевантний контент або спеціальні пропозиції, які збільшать задоволеність та ймовірність конверсії. Зібрані дані допомагають ідентифікувати слабкі місця та проблемні області на вебсайті. Аналіз поведінки користувачів дозволяє виявити причини відмов, низьку конверсію або високий відтік. Це дає можливість внести необхідні зміни та покращити користувальницький досвід, що в кінцевому підсумку призводить до збільшення конверсії та успіху бізнесу. Дані про користувачів та їхню поведінку можуть бути використані для прогнозування

майбутніх тенденцій та планування маркетингових стратегій. Аналітика вебсайтів дозволяє передбачити потенційний попит, ідентифікувати нові можливості та вжити заходів для досягнення конкурентної переваги.

В аналітиці вебсайтів існує кілька методів збору даних, які допомагають збирати інформацію щодо поведінки користувачів:

– код відстеження ґрунтується на впровадженні спеціального коду відстеження на вебсайт. Код відстеження, що зазвичай розміщується на кожній сторінці вебсайту, збирає дані про дії користувачів, такі як відвідування сторінок, натискання на посилання, заповнення форм та інші взаємодії. Найпопулярнішим інструментом для відстеження коду є Google Analytics;

– коли користувач запитує вебсторінку, сервер вебсайту записує інформацію про запит до журналу сервера. Ця інформація включає IP-адресу користувача, дату та час запиту, запитану сторінку та інші дані. Журнали сервера дозволяють отримати загальну інформацію про відвідування, але вони можуть бути менш докладними, ніж дані, зібрані за допомогою коду відстеження;

– анкети та опитування включають проведення анкет або опитувань серед користувачів, щоб отримати інформацію про їх переваги, потреби та думки. Анкети можуть містити питання про вік, поле, освіту, інтереси та інші параметри, які можуть бути корисні для сегментації та аналізу аудиторії;

– тестування інтерфейсу користувача дозволяє збирати дані про взаємодію користувачів з вебсайтом в реальному часі. Це може включати проведення A/B-тестування, де дві або більше версії вебсторінки порівнюються, щоб визначити, яка з них працює краще в термінах конверсії або залучення користувача.

В аналітиці вебсайтів існує ряд основних метрик, які використовуються для вимірювання та аналізу продуктивності та ефективності вебсайту:

– кількість відвідувань показує загальну кількість відвідувань вебсайту протягом певного періоду часу. Вона дає уявлення про популярність сайту та загальний обсяг трафіку, який він приваблює. Висока кількість відвідувань може свідчити про хорошу видимість та привабливість вебсайту;

– унікальні відвідувачі відображає кількість унікальних користувачів, які відвідали вебсайт. Вона дозволяє визначити скільки різних людей зайшло на сайт у заданий період часу. Унікальні відвідувачі допомагають оцінити, наскільки широко вебсайт досягає своєї аудиторії;

– сеанс є періодом, протягом якого користувач активно взаємодіє з вебсайтом. Метрика сеансів показує загальну кількість сеансів на вебсайті. Вона може дати уявлення про те, як довго користувачі залишаються на сайті та наскільки глибоко вони взаємодіють із його контентом;

– відмови відображає відсоток користувачів, які залишають вебсайт після перегляду лише однієї сторінки, не виконавши жодних додаткових дій. Відмовлення можуть вказувати на проблеми із залученням та утриманням користувачів. Чим нижчий відсоток відмов, тим ефективнішим вважається вебсайт у залученні та утриманні своїх відвідувачів.

Згідно з дослідженням Google, високі показники відмов можуть залежати від категорії сайту (рис. 2.2), що може вказувати на низьку якість контенту, поганий інтерфейс користувача або проблеми із завантаженням сторінки. Це підкреслює важливість зниження показників відмов для поліпшення досвіду користувача і підвищення задоволеності користувачів.

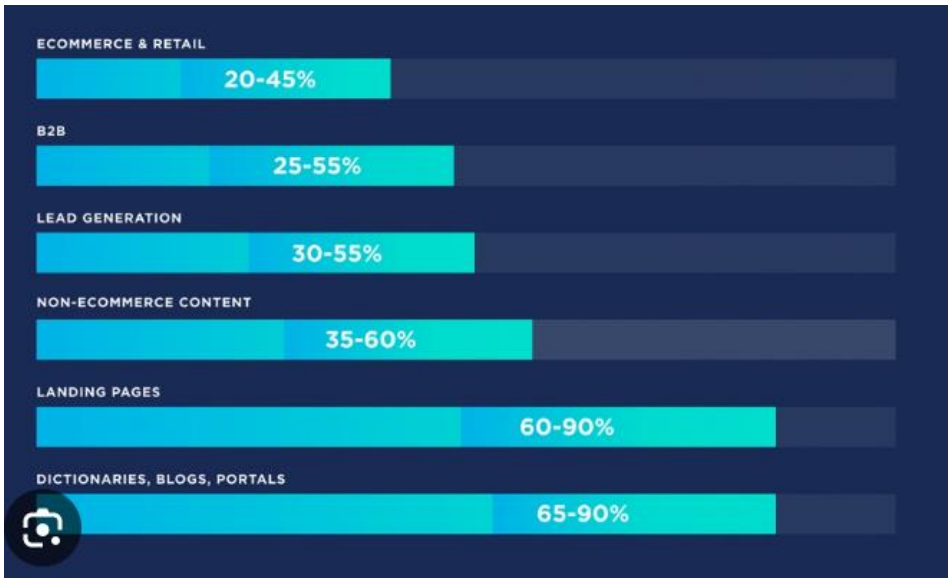

Рисунок 2.2 – Середній показник відмов в залежності від категорії сайту

Процес аналізу даних в аналітиці вебсайтів включає збирання, обробку, візуалізацію та інтерпретацію даних з метою отримання цінної інформації про продуктивність та поведінку користувачів. Ось основні етапи процесу аналізу даних:

– спочатку необхідно зібрати дані про відвідування, дії користувачів та інші метрики за допомогою інструментів аналітики вебсайту, таких як Google Analytics. Ці дані можуть включати інформацію про кількість відвідувань, джерела трафіку, час перебування на сайті та інші показники;

– після збору даних вони мають бути оброблені та структуровані для зручного аналізу. Це може включати фільтрацію та видалення некоректних чи небажаних даних, агрегацію даних для отримання загальних показників та створення зведених таблиць чи графіків для наочного подання інформації;

– одним із ключових аспектів аналізу даних в аналітиці вебсайтів є їхня візуалізація. Візуалізація дозволяє подати дані у зрозумілій та наочній формі, використовуючи графіки, діаграми та інші візуальні елементи. Це допомагає візуально уявити тренди, патерни та взаємозв'язки між різними метриками;

– остаточний етап аналізу даних полягає у їх інтерпретації. Інтерпретація даних вимагає розуміння контексту та цілей вебсайту. Аналітик повинен проводити порівняльний аналіз метрик, звертати увагу на зміни та тренди, а також шукати причинно-наслідкові зв'язки. Важливо звернути увагу на основні висновки, які можна зробити на основі даних, та використовувати їх для прийняття рішень щодо оптимізації вебсайту.

Методи збору даних про інтернет-користувачів, зокрема аналітика вебсайту, є важливими і ефективними інструментами для отримання інформації про поведінку та характеристики користувачів в онлайн-середовищі. Ці дані дозволяють власникам вебсайтів та маркетологам аналізувати та оптимізувати свої стратегії, покращувати користувальницький досвід та досягати бізнес-цілей. Збір і аналіз даних про користувачів надає об'єктивну інформацію, на основі якої можна приймати обґрунтовані рішення щодо вебсайту, персоналізації контенту та

маркетингових кампаній. Крім того, дані про користувачів можуть бути використані для прогнозування майбутніх тенденцій та планування стратегій, що допомагає досягти конкурентної переваги. Загалом, збір і аналіз даних про користувачів вебсайту є незамінним інструментом для ефективного керування та розвитку онлайн-присутності.

2.2 Технології розробки системи

У 2023 році найпопулярнішою мовою залишається JavaScript – 19% розробників пишуть нею комерційні проєкти. Далі – Java (14%) і Python (13%). Остання вперше серед найпопулярніших. З мінімальним відривом за Python йдуть C# і TypeScript (рис 2.3).

Рисунок 2.3 – Середній показник відмов в залежності від категорії сайту

Розглянемо детальніше декілька мов програмування, які найкраще підходять для розробки інтелектуальної системи збору та аналізу даних поведінки користувача у вебсередовищі.

2.2.1 JavaScript

JavaScript є однією з ключових технологій для розробки інтелектуальних систем збирання та аналізу даних користувача у вебсередовищі. Він надає розробникам потужні інструменти та можливості для створення інтерактивних та динамічних вебдодатків, які можуть збирати та аналізувати дані користувача на основі його дій та поведінки. Ось деякі з переваг та недоліків JavaScript у контексті розробки такої системи:

Переваги JavaScript:

– JavaScript є однією з найпопулярніших мов програмування та широко підтримується всіма сучасними веббраузерами. Це означає, що розробники можуть створювати інтелектуальні системи збору та аналізу даних, які працюватимуть на більшості пристроїв та платформ;

– JavaScript легко вбудовується в HTML-код вебсторінки, що дозволяє створювати інтерактивні елементи та керувати поведінкою сторінки. Це робить JavaScript ідеальним інструментом для збирання даних користувача на вебсторінках;

– JavaScript підтримує асинхронне програмування, що дозволяє ефективно обробляти великі обсяги даних без блокування інтерфейсу користувача. Це особливо корисно при роботі із запитами на сервер та обробці даних у реальному часі;

– існує безліч бібліотек та фреймворків JavaScript, таких як React, Angular та Vue.js, які надають готові інструменти та компоненти для розробки складних вебдодатків. Вони можуть значно спростити розробку та підвищити продуктивність системи.

Недоліки JavaScript:

– JavaScript виконується на стороні клієнта, що може створювати вразливості безпеки, такі як міжсайтова підробка запиту (CSRF) та ін'єкції коду. Розробники повинні вживати заходів для захисту даних та безпеки при розробці інтелектуальних систем збору та аналізу даних;

– використання JavaScript для збирання та аналізу даних користувача вимагає підтримки JavaScript з боку браузера користувача. Якщо користувач відключив JavaScript або використовує стару версію браузера, функціональність системи може бути обмежена або повністю недоступна;

– JavaScript має деякі обмеження при роботі з даними, особливо при обробці великих обсягів інформації. Деякі складні аналітичні завдання можуть вимагати використання інших мов програмування чи інструментів.

Моделі та методи для розробки системи збору та аналізу даних користувача:

– JavaScript дозволяє відстежувати різні події, такі як клацання, наведення курсора, введення даних та інші дії користувача. Ці події можуть бути використані для збору даних щодо поведінки користувача;

– AJAX (асинхронні JavaScript та XML) та вебсокети дозволяють обмінюватися даними між вебсторінкою та сервером без перезавантаження сторінки. Це дозволяє передавати дані користувача на сервер для аналізу та отримувати обернені дані для оновлення інтерфейсу в реальному часі;

– JavaScript має набір бібліотек, таких як TensorFlow.js та Brain.js, які надають можливості машинного навчання та аналізу даних у браузері. Ці інструменти дозволяють розробникам реалізувати алгоритми навчання, класифікації та прогнозування, щоб аналізувати дані користувача;

– JavaScript підтримує різні методи зберігання даних, такі як локальне сховище (localStorage) та бази даних IndexedDB. Це дозволяє зберігати та аналізувати дані користувача на клієнтській стороні;

– JavaScript має безліч бібліотек для візуалізації даних, таких як D3.js та Chart.js. Ці бібліотеки дозволяють створювати графіки, діаграми та інші візуальні уявлення даних користувача.

Розробка інтелектуальної системи збору та аналізу даних користувача у вебсередовищі вимагає грамотного використання JavaScript у поєднанні з іншими технологіями та методами аналізу даних. Важливо враховувати як переваги, так і недоліки JavaScript під час проєктування та розробки такої системи.

2.2.2 Java

Java є потужною технологією для розробки інтелектуальних систем збирання та аналізу даних користувача у вебсередовищі. Вона пропонує розробникам широкий набір інструментів та можливостей для створення надійних та масштабованих вебдодатків, здатних збирати та аналізувати дані користувача на

основі його взаємодії та поведінки. Ось деякі з переваг та недоліків Java у контексті розробки такої системи:

Переваги Java:

– Java є платформонезалежною мовою програмування, що означає, що програми, написані на Java, можуть працювати на різних операційних системах та апаратних платформах. Це забезпечує універсальність та доступність системи для широкого кола користувачів;

– Java має вбудовані механізми безпеки, які допомагають запобігати вразливості та захищати дані користувача. Віртуальна машина Java (JVM) контролює виконання програмного коду та запобігає доступу до ресурсів системи без відповідних дозволів;

– Java має безліч інструментів і фреймворків для розробки масштабованих вебзастосунків. Це дозволяє розробляти системи, здатні обробляти великі обсяги даних та підтримувати високу продуктивність при одночасній роботі з великою кількістю користувачів;

– Java має велику стандартну бібліотеку класів (Java API), яка надає різні інструменти та функції для обробки даних, мережевої взаємодії, автентифікації та інших завдань. Це значно спрощує розробку системи збору та аналізу даних користувача.

Недоліки Java:

– Java є відносно складною мовою програмування, що вимагає від розробників глибокого розуміння та досвіду. Це може виявитися недоліком для програмістів-початківців або команд з обмеженими ресурсами;

– розробка Java може вимагати великих обчислювальних ресурсів і використання багатошарової архітектури. Це може спричинити вищі витрати на обладнання та розгортання системи;

– на відміну від JavaScript, Java вимагає встановлення Java Runtime Environment (JRE) на стороні клієнта для коректної роботи. Деякі браузери можуть

обмежувати або не підтримувати використання аплетів Java на вебсторінках, що може обмежити доступність системи для користувачів.

Моделі та методи для розробки системи збору та аналізу даних користувача на Java:

– Java Servlets та JavaServer Pages (JSP) надають можливості для створення динамічних вебсторінок та обробки запитів від користувачів. За допомогою цих технологій можна збирати дані від користувача та обробляти їх на сервері;

– Java Enterprise Edition (Java EE) надає широкий набір специфікацій та API для розробки масштабованих та надійних вебдодатків. Існують також популярні фреймворки, такі як Spring та Hibernate, які спрощують розробку та інтеграцію системи з іншими компонентами;

– Java має багатий набір інструментів для роботи з базами даних, таких як JDBC (Java Database Connectivity) та JPA (Java Persistence API). Це дозволяє зберігати та аналізувати дані користувача в базах даних;

– Java надає бібліотеки, такі як Weka та Apache Mahout, які містять реалізації різних алгоритмів машинного навчання та аналізу даних. Ці інструменти можна використовувати для обробки та аналізу даних користувача.

Розробка інтелектуальної системи збору та аналізу даних користувача у вебсередовищі на Java вимагає глибокого розуміння мови та використання відповідних інструментів та фреймворків. Враховуючи переваги та недоліки Java, розробники можуть створити надійну та ефективну систему для збирання та аналізу даних користувача.

2.2.3 Python

Python є потужною мовою програмування, що широко використовується для розробки інтелектуальних систем збору та аналізу даних користувача у вебсередовищі. Python пропонує простий синтаксис, велику бібліотеку та безліч інструментів, які роблять його популярним вибором для створення таких систем.

Ось деякі з переваг та недоліків Python у контексті розробки системи збору та аналізу даних користувача:

Переваги Python:

– Python має простий і зрозумілий синтаксис, який робить код читаним і дозволяє розробникам швидко писати та підтримувати програми. Це особливо важливо для систем збору та аналізу даних, де ефективність та швидкість розробки відіграють важливу роль;

– Python має велику стандартну бібліотеку, яка включає модулі для роботи з різними типами даних, мережевою взаємодією, веброзробкою та аналізом даних. Крім того, існує безліч сторонніх бібліотек та фреймворків, таких як NumPy, Pandas, Scikit-learn та TensorFlow, які надають потужні інструменти для роботи з даними;

– Python є популярним вибором для розробки систем машинного навчання та аналізу даних. Завдяки бібліотекам, таким як TensorFlow, PyTorch та Scikit-learn, розробники можуть використовувати різні моделі машинного навчання, алгоритми класифікації та кластеризації для аналізу даних користувача;

– Python має безліч фреймворків, таких як Django та Flask, які спрощують розробку вебдодатків. З їх допомогою можна створювати інтерфейси користувача, обробляти запити від клієнтів, а також збирати та аналізувати дані користувача у вебсередовищі.

Недоліки Python:

– Python є інтерпретованою мовою програмування, що означає, що вона може бути повільнішою у виконанні порівняно з мовами, що компілюються. Це може бути недоліком у разі обробки великих обсягів даних або вимог до високої продуктивності;

– Python має обмежені можливості для паралельного виконання коду через глобальне блокування інтерпретатора (Global Interpreter Lock, GIL). Це може зашкодити продуктивності системи у разі використання багато поточності для обробки даних.

Моделі та методи для розробки системи збору та аналізу даних користувача на Python:

– за допомогою бібліотек, таких як Beautiful Soup або Scrapy, можна збирати дані з вебсторінок, витягувати інформацію та зберігати її для аналізу;

– бібліотеки, такі як Pandas, NumPy та Scikit-learn, надають потужні інструменти для обробки та аналізу даних, а також для побудови моделей машинного навчання для класифікації, кластеризації, рекомендацій та прогнозування;

– Python має бібліотеки, такі як NLTK та SpaCy, які дозволяють аналізувати та обробляти текстові дані, включаючи розбір синтаксису, вилучення ключових слів та сутностей, а також класифікацію тексту;

– Python надає модулі для роботи з різними базами даних, такими як SQLite, MySQL та PostgreSQL. Це дозволяє зберігати та витягувати дані користувача з баз даних для подальшого аналізу.

Python пропонує широкий спектр інструментів та можливостей для розробки інтелектуальної системи збору та аналізу даних користувача у вебсередовищі. Завдяки простоті використання та багатій екосистемі Python, розробники можуть створити ефективну та гнучку систему для збирання та аналізу даних користувача.

2.2.4 C#

C# (C Sharp) є потужною та гнучкою мовою програмування, розробленою Microsoft, і надає широкий спектр інструментів та можливостей для створення інтелектуальних систем збирання та аналізу даних користувача у вебсередовищі. Ось деякі з переваг та недоліків використання C# для розробки системи:

Переваги:

– C# забезпечує доступ до Visual Studio, однієї з найпопулярніших IDE для розробки програм. Visual Studio надає багатий набір інструментів і функцій, таких як налагоджувач, інтелектуальне завершення коду та інструменти для створення

інтерфейсу користувача, спрощуючи розробку та налагодження інтелектуальної системи;

– C# є однією з найпопулярніших мов програмування, і в ній існує велика спільнота розробників, які готові надати підтримку та допомогу. Існує також велика кількість документації, навчальних матеріалів та онлайн-ресурсів, що полегшують вивчення мови та її застосування для розробки інтелектуальних систем;

– C# розроблявся для роботи з .NET-платформою, яка надає потужні бібліотеки та інструменти для розробки вебдодатків. Використання C# у поєднанні з .NET дозволяє легко обробляти дані, встановлювати з'єднання з базами даних та створювати вебсервіси;

– C# має вбудовані механізми безпеки, такі як перевірка типів під час компіляції, управління пам'яттю та механізми обробки винятків. Це допомагає знизити можливість помилок та забезпечити безпеку даних користувачів.

Недоліки:

– C# є мовою, специфічною для платформи Windows та .NET. Це обмежує його застосування в інших операційних системах та платформах, що може обмежити аудиторію системи;

– хоча C# має велику спільноту розробників та велику документацію, вивчення мови та її застосування потребує певного часу та зусиль. Якщо команда розробників не знайома з C# і .NET, може знадобитися час на освоєння мови та платформи.

Моделі та методи для розробки інтелектуальної системи збору та аналізу даних користувача у вебсередовищі з використанням C# можуть включати:

– використання вебскрапінгу (web scraping) для отримання даних із вебсторінок та API для взаємодії з вебсервісами та додатками;

– застосування алгоритмів обробки даних, таких як фільтрація, агрегація, перетворення та статистичний аналіз для очищення та підготовки даних для подальшого аналізу;

– використання алгоритмів машинного навчання, таких як класифікація, кластеризація та регресія, для аналізу даних користувача та виявлення патернів, тенденцій чи аномалій;

– розробка вебінтерфейсів та дашбордів для відображення та візуалізації результатів аналізу даних, щоб користувачі могли отримувати інформацію у зручному та зрозумілому вигляді;

– реалізація механізмів шифрування, автентифікації та авторизації для забезпечення безпеки та конфіденційності даних користувачів;

– враховуючи, що вебсередовище може мати великий потік даних, важливо розробити систему, здатну масштабуватись та обробляти великі обсяги даних ефективно.

2.2.5 PHP

PHP є широко використовуваною мовою програмування, спеціально призначеною для розробки вебдодатків. Він також може бути використаний для розробки інтелектуальних систем збирання та аналізу даних користувача у вебсередовищі. Ось деякі з переваг та недоліків використання PHP для цієї мети:

Переваги:

– PHP є однією з найпопулярніших мов програмування для веброзробки. Він має велику спільноту розробників, що забезпечує доступ до безлічі ресурсів, бібліотек та розширень. Також існує безліч форумів та онлайн-спільнот, де можна отримати підтримку та допомогу;

– PHP має простий та зрозумілий синтаксис, заснований на мові C, що робить його відносно легким для вивчення та використання. Це дозволяє швидко створювати прототипи та розробляти вебдодатки;

– PHP має широкий набір вбудованих функцій і бібліотек для роботи з різними аспектами веброзробки. Існує безліч розширень та фреймворків, таких як Laravel та Symfony, які надають додаткові інструменти для прискорення розробки та обробки даних;

– PHP надає широкі можливості для роботи з різними системами управління базами даних (СУБД), такими як MySQL, PostgreSQL, Oracle та інші. Це дозволяє ефективно зберігати та витягувати дані користувача для подальшого аналізу.

Недоліки:

– PHP є мовою програмування, що інтерпретується, що може позначитися на продуктивності при обробці великих обсягів даних. Однак з використанням оптимізації та кешування можна впоратися з цією проблемою;

– деякі архітектурні рішення в PHP можуть ускладнювати масштабування вебзастосунків при великому навантаженні. Оптимізація коду та правильне використання кешування можуть допомогти впоратися з цим недоліком.

Моделі та методи, які можна застосувати для розробки інтелектуальної системи збору та аналізу даних користувача у вебсередовищі з використанням PHP, включають:

– використання HTTP-запитів та API для отримання даних із вебсторінок, вебслужб або соціальних мереж. Також можна застосувати техніки парсингу (наприклад, з використанням бібліотеки Simple HTML DOM) для отримання потрібної інформації;

– розробка функцій та алгоритмів для очищення, перетворення та агрегації даних. Наприклад, фільтрація небажаних даних, перетворення форматів даних або агрегація даних для отримання загальної статистики;

– застосування алгоритмів машинного навчання (наприклад, з використанням бібліотеки PHP-ML) для класифікації, кластеризації або передбачення даних користувача;

– використання бібліотек та інструментів, таких як Chart.js або Google Charts, для створення графіків, діаграм та візуальних звітів для надання результатів аналізу даних;

– реалізація механізмів шифрування, зберігання безпечних паролів, автентифікації та авторизації для забезпечення безпеки та конфіденційності даних користувачів.

2.3 Вибір мови програмування

PHP – це інтерпретована мова програмування, спеціально розроблена для створення динамічних вебсторінок та взаємодії з базами даних. Він має простий та інтуїтивно зрозумілий синтаксис, що робить його доступним для широкого кола розробників. PHP широко підтримується вебсерверами та працює на більшості операційних систем, що робить його дуже популярним вибором для веброзробки.

PHP володіє широким набором функцій та модулів, що робить його ідеальним інструментом для створення систем відстеження поведінки та збору даних користувача у вебсередовищі.

Для розробки системи відстеження поведінки та збору даних користувача на PHP доступно кілька модулів та бібліотек:

– для отримання інформації про країну, з якої користувач відвідує ваш вебсайт, можна використовувати модуль GeoIP. GeoIP дозволяє визначити географічне розташування користувача на основі його IP-адреси. Існують різні бібліотеки, такі як MaxMind GeoIP та GeoIP2, які надають API для роботи з геолокаційними даними;

– для збору інформації про браузер та операційну систему користувача ви можете використовувати модуль браузера. Цей модуль дозволяє отримувати інформацію про заголовки HTTP-запитів, які надсилає браузер на сервер. Наприклад, ви можете отримати інформацію про тип браузера, версію, мову, функції, що підтримуються і т. д. Стандартна бібліотека PHP надає функції, такі як *get_browser(),* які дозволяють отримати ці дані;

– для відстеження дій користувача на вебсайті можна використовувати функції сесій. Сесії в PHP дозволяють зберігати дані між запитами та надають унікальний ідентифікатор сесії для кожного користувача. Ви можете

використовувати цей ідентифікатор для запису даних про дії користувача до бази даних або файлу. Функції, такі як session_start(), *$_SESSION* та *session_write_close()*, допоможуть вам керувати сесіями в PHP;

– PHP також надає функціональність для збору даних із форм, заповнених користувачами на вашому вебсайті. При надсиланні форми дані можуть бути передані на сервер, і ви можете отримати доступ до них за допомогою змінної $_POST, якщо методом передачі даних був обраний POST, або $_GET, якщо був обраний метод GET. Ви можете зберегти ці дані в базі даних або обробити їх відповідно до ваших вимог;

– для збереження зібраних даних вам може знадобитися використовувати базу даних. У PHP можна використовувати різні розширення для роботи з різними типами баз даних, такими як MySQL, PostgreSQL або MongoDB. Наприклад, для роботи з MySQL можна використовувати розширення PDO (PHP Data Objects) або mysqli. Ці розширення надають набір функцій для підключення до бази даних, виконання запитів та обробки даних.

PHP забезпечує різні можливості для зчитування даних щодо поведінки користувачів на вебсайті. Наприклад, він може отримувати інформацію про відвідані сторінки, час перебування на сторінці, взаємодію з елементами інтерфейсу, відправлення форм та інші дії користувача. Ці дані можуть бути зібрані за допомогою різних технік, таких як обробка GET і запитів POST, аналіз HTTP заголовків, читання файлів журналів сервера і т.д.

Отримані дані щодо поведінки користувача можуть бути збережені та організовані для подальшого використання. І тут на допомогу приходить MySQL. MySQL – це вільна реляційна база даних, яка пропонує ефективне зберігання та вилучення даних. Вона працює на багатьох платформах та підтримує безліч функцій для роботи з даними.

У контексті систем відстеження поведінки користувачів MySQL може використовуватися для створення та управління базою даних, в якій зберігаються зібрані дані. Створення таблиць MySQL для зберігання інформації про поведінку

користувача може бути виконано з використанням SQL-запитів. Наприклад, можна створити таблицю "логи", в якій зберігатимуться дані про відвідувані сторінки, дату та час відвідування, ідентифікатор користувача та інші відомості.

PHP надає зручний інтерфейс для взаємодії з MySQL. За допомогою PHP можна встановлювати з'єднання з базою даних, виконувати SQL-запити для читання та запису даних, отримувати та обробляти результати запитів. PHP також забезпечує механізми безпеки, такі як підготовлені вирази (prepared statements) та фільтрацію даних, щоб запобігти можливим атакам на базу даних.

Переваги використання PHP та MySQL для розробки систем відстеження поведінки користувачів в Інтернеті є такими:

– PHP та MySQL дозволяють створювати системи, які можна легко налаштовувати та розширювати відповідно до вимог бізнесу. Можна легко додавати нові функції, змінювати схему бази даних та обробляти великі обсяги даних;

– PHP та MySQL є дуже популярними технологіями з величезною спільнотою розробників. Це означає, що завжди можна знайти безліч ресурсів, документації, навчальних матеріалів та готових рішень для реалізації систем відстеження поведінки користувачів;

– MySQL забезпечує швидке зберігання та вилучення даних, що дозволяє обробляти великі обсяги інформації без значних затримок. PHP, у свою чергу, є швидкою та ефективною мовою програмування, що сприяє обробці запитів та виконанню операцій з базою даних;

– PHP та MySQL пропонують механізми для забезпечення безпеки даних. PHP забезпечує можливість використання підготовлених виразів, які запобігають атакам на кшталт SQL-ін'єкцій. MySQL також має вбудовані механізми для авторизації та шифрування даних;

– PHP та MySQL добре інтегруються з іншими технологіями, такими як HTML, CSS, JavaScript, фреймворки для веброзробки та інші компоненти вебстеку.

У результаті PHP і MySQL є потужним комбінацією для розробки систем відстеження поведінки користувачів в інтернеті. Вони забезпечують можливості зчитування, зберігання та аналізу даних, що дозволяє покращити користувальницький досвід, оптимізувати маркетингові стратегії та приймати більш проінформовані рішення на основі даних щодо поведінки користувачів.

2.4 Інструменти, необідні для реалізації застосунку

Бібліотека PHP-ML – це відкрита бібліотека машинного навчання для мови програмування PHP. Вона надає широкий набір алгоритмів машинного навчання та інструменти для виконання різних задач, пов'язаних з обробкою даних, класифікацією, кластеризацією, регресією та іншими аспектами машинного навчання.

Основна мета PHP-ML – зробити машинне навчання доступним для розробників PHP, дозволяючи їм використовувати потужні алгоритми машинного навчання без необхідності вивчення нових мов або перекладу вже наявних моделей. Бібліотека має простий інтерфейс та гнучкість у використанні.

Декілька основних можливостей, які надає PHP-ML:

– бібліотека надає алгоритми для класифікації даних, такі як наївний Байєс, k-найближчих сусідів, решітки рішень (decision tree), метод опорних векторів (SVM) та інші. Ви можете навчити модель на основі навчальних даних та використовувати її для прогнозування класу нових даних;

– PHP-ML також має підтримку алгоритмів кластеризації, таких як k-середніх (k-means), агломеративна кластеризація та інші. Ви можете використовувати ці алгоритми для групування даних без попереднього знання про кількість або структуру кластерів;

– PHP-ML надає алгоритми для побудови моделей регресії, що дозволяють прогнозувати числові значення на основі навчальних даних;

– бібліотека має функціонал для обробки текстових даних, такий як векторизація тексту, токенізація, видалення стоп-слів та інші операції, що допомагають підготувати дані для подальшого аналізу;

– PHP-ML надає інструменти для оцінки якості моделей машинного навчання, таких як розрахунок точності, виклик метрик, таких як точність, відновлення та F-мера;

– бібліотека також підтримує алгоритми навчання без вчителя, такі як аналіз головних компонентів (PCA), згорткова нейронна мережа (SOM) та інші.

Висновки до розділу 2

В результаті написання другого розділу було визначено методи збору даних про інтернет-користувачів, зокрема аналітика вебсайту, що є важливим інструментом для збирання, аналізу та використання інформації про поведінку та характеристики користувачів.

Розглянуто різноманітні види доступної інформації про користувача, що є метриками, які демонструють перевагу аналітики вебсайтів. Визначено, що процес аналізу даних в аналітиці вебсайтів включає збирання, обробку, візуалізацію та інтерпретацію даних з метою отримання цінної інформації про продуктивність та поведінку користувачів.

Було розглянуто використання PHP та MySQL для створення системи відстеження поведінки та збору даних користувача. PHP є популярною мовою програмування для веброзробки, яка має простий синтаксис і широкі можливості. Вона підтримується більшістю вебсерверів і працює на різних операційних системах.

Для виконання конкретних завдань у системі було розглянуто різні модулі та бібліотеки PHP. Наприклад, модуль GeoIP використовується для визначення географічного розташування користувача на основі його IP-адреси. Модуль браузера дозволяє отримати інформацію про браузер та операційну систему

користувача. Сесії в PHP використовуються для відстеження дій користувача, а функціональність PHP дозволяє збирати дані з форм та записувати їх до бази даних.

MySQL використовується як база даних для зберігання зібраних даних про поведінку користувачів. Вона пропонує ефективне зберігання та вилучення даних і добре інтегрується з PHP. PHP надає зручний інтерфейс для взаємодії з MySQL та забезпечує механізми безпеки для захисту даних.

3 ПРОЄКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Проєктування інтелектуальної системи

Для проєктування інтелектуальної системи зроби даних та аналізу поведінки користувача для початку треба розробити план її роботи, що зображений на рисунку 3.1.

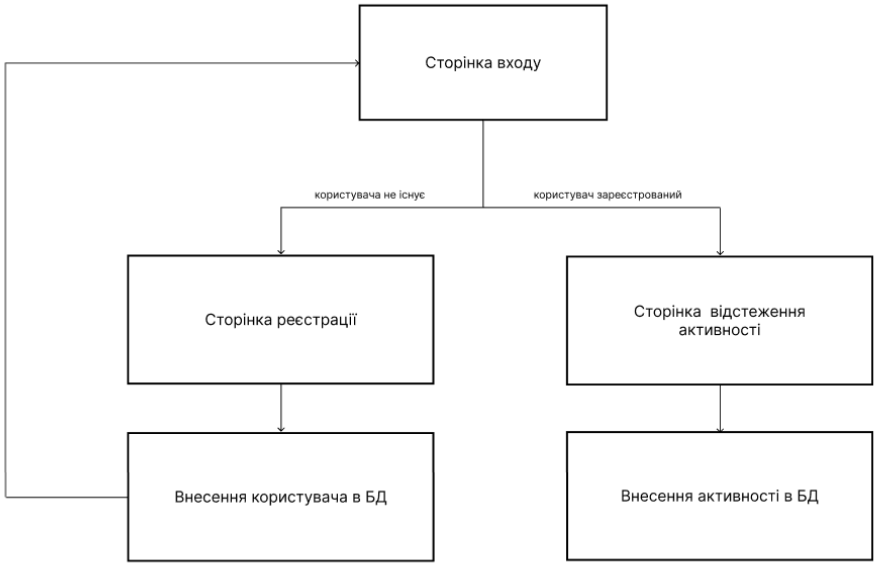

Рисунок 3.1 – План роботи інтелектуальної системи збору даних на аналізу поведінки користувача

Ідея полягає в тому, що це є системою автентифікації та збору даних про поведінку користувача у вебсередовищі, яка може бути доопрацьована та розширена відповідно до конкретних вимог та цілей проєкту.

Ця система використовує сесії для відстеження унікальності користувача та збереження його стану протягом сеансу взаємодії з вебпрограмою.

У разі успішної аутентифікації користувача (вході в систему) скрипт *login.php* створює сесію для цього користувача. Сесія є унікальним ідентифікатором (session ID), який зберігається на сервері і відправляється назад на

клієнтську сторону у вигляді cookie. Потім при кожному наступному запиті від користувача його session ID передається на сервер, дозволяючи серверу ідентифікувати користувача і зв'язати його зі збереженими даними сесії.

У контексті даної системи унікальність користувача визначається через сесію. Коли користувач виконує вхід, його ім'я та пароль перевіряються на відповідність у базі даних. Якщо дані правильні, створюється сесія, і користувач вважається успішно автентифікованим. Всі наступні дії цього користувача в рамках сесії будуть пов'язані з унікальним ідентифікатором сесії.

За допомогою даних сесії система відстежує активність користувача на сторінці активності та зберігає інформацію про цю активність у базі даних у таблиці "activity". Таким чином, кожен запис про активність користувача буде пов'язаний із його унікальною сесією.

Сесії забезпечують збереження стану користувача, дозволяючи вебпрограмі відстежувати його дії в рамках одного сеансу, а також відрізняти його від інших користувачів. Це дозволяє системі розрізняти унікальних користувачів та зберігати інформацію про їхню активність окремо.

Метод cookies та метод унікального користувача представляють різні способи ідентифікації користувачів та відстеження їхньої активності. Cookie є невеликими текстовими файлами, які зберігаються на комп'ютері користувача. Cookie використовуються для зберігання інформації про сеанс користувача на вебсайті, включаючи дані автентифікації, налаштування та інші дані.

Переваги:

– зручність використання: cookie автоматично відправляються на сервер з кожним запитом користувача, що забезпечує простоту взаємодії сесії на стороні сервера;

– підтримка браузерами: cookie підтримуються практично всіма сучасними веббраузерами.

Недоліки:

– обмежений обсяг даних: розмір cookie обмежений (зазвичай до 4 КБ), що може стати проблемою при необхідності зберігання великого обсягу даних;

– необхідність підтримки клієнтської сторони: користувач може вимкнути підтримку кук у своєму браузері або видалити їх, що призведе до втрати сеансової інформації;

– вразливість до атак: некоректна реалізація або використання небезпечних кук може становити загрозу для безпеки та приватності користувачів.

Метод унікального користувача використовує інші механізми ідентифікації, такі як IP-адреса, браузерні відбитки, унікальні ідентифікатори пристроїв і т.д.

Переваги:

– надійніша ідентифікація: метод унікального користувача може бути надійнішим, оскільки він не покладається лише на зберігання даних на стороні клієнта;

– не залежить від кук: не вимагає використання кук і не схильний до пов'язаних з ними обмежень і вразливостей.

Недоліки:

– складність реалізації: метод унікального користувача потребує складнішої логіки та інтеграції для ідентифікації користувачів;

– можливість помилок: ідентифікація користувача на основі IP-адреси або інших характеристик може бути неточною або схильна до помилок при використанні проксі-серверів або мереж із загальним доступом.

Обидва методи мають свої переваги та недоліки, і вибір між ними залежить від вимог конкретної системи, безпеки та приватності даних користувачів. У деяких випадках може бути корисно комбінувати обидва методи для досягнення більш надійної ідентифікації та відстеження користувачів.

Система передбачає створення бази даних, в яку вносять дані унікальні дані відвідувача та його активність на сторінці. Система передбачає автоматичне

створення таблиці у базі даних, якщо їх не існує. У цій базі даних є дві таблиці: "users" та "activity". Розглянемо кожну з них докладніше:

Таблиця "users":

– id (INT) - автоінкрементний ідентифікатор користувача;

– username (VARCHAR(50)) – ім'я користувача. Унікальне поле, що використовується для перевірки існування користувача та входу до системи;

– password (VARCHAR(255)) - пароль користувача.

Значення для цієї таблиці потрапляють із форми реєстрації. При надсиланні форми дані з полів "Ім'я користувача" та "Пароль" зберігаються в таблиці "users" у відповідних полях.

Таблиця "activity":

– id (INT) - ідентифікатор активності користувача;

– username (VARCHAR(50)) - ім'я користувача, пов'язане з цією активністю;

– login_time (DATETIME) – час входу користувача в систему;

– logout_time (DATETIME) – час виходу користувача із системи;

– duration (INT) - тривалість сеансу користувача в системі в секундах;

– browser (VARCHAR(50)) - інформація про браузер, який використовується користувачем;

– country (VARCHAR(100)) - країна користувача, визначена на основі його IP-адреси;

– clicks (INT) – кількість кліків, зроблених користувачем;

– bevavior (VARCHAR(50)) – класифікація користувача за активністю на сторінці.

Значення для цієї таблиці потрапляють із форми входу та сторінки активності. При успішному вході користувача в систему записується інформація про його активність (час входу, ім'я користувача тощо) в таблицю "activity". При виході користувача з системи, записується інформація про час виходу та тривалість сеансу, а також про кліки та інформацію про браузер та країну користувача.

Таким чином, база даних включає таблиці "users" та "activity", які зберігають інформацію про користувачів та їх активність у системі, відповідно.

Для визначення країни у системі використовується сервіс ip-api.com. Цей сервіс надає інформацію про місцеперебування на основі IP-адреси користувача.

Ip-api.com - це безкоштовний зовнішній сервіс, що надає інформацію про місцеперебування на основі IP-адреси. Він дозволяє визначити країну, регіон, місто, поштовий індекс та інші дані, пов'язані з IP-адресою користувача.

Базовий шлях API: *http://ip-api.com/php/{query}.*

{query} – може бути окремою адресою IPv4/IPv6 або ім’ям домену. Якщо запит не наданий, буде використано поточну IP-адресу.

Проведемо швидкий тест для визначання інформації за IP-адресою (рис. 3.2). Для використання поточної IP-адреси не вписуємо запит у базовий шлях.

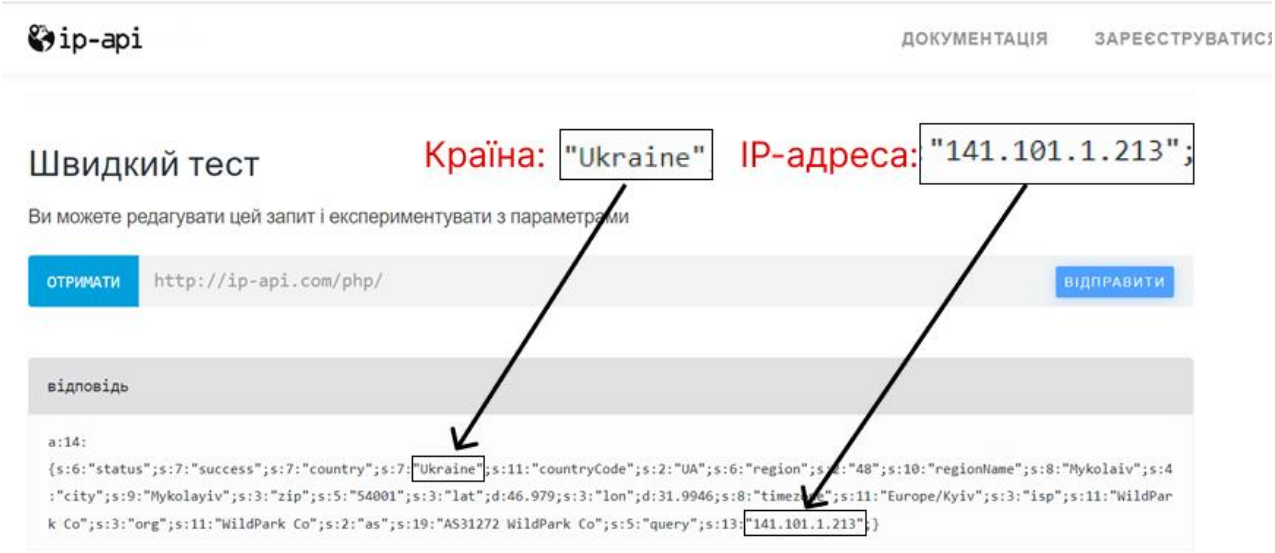

Рисунок 3.2 – Швидкий тест на сервері ip-api.com, що надає інформацію про місцеперебування на основі IP-адреси

У інформаційній системі ip-api.com використовується для отримання країни, звідки користувач виконує дії. У функції getCountryFromIP($ipAddress), яка визначена у файлі activity.php, IP-адреса передається як параметр в URL-адресу сервісу ip-api.com (рис. 3.3). Потім функція робить GET-запит до сервісу та отримує відповідь у форматі JSON.

```php
function getCountryFromIP($ipAddress) {
    $url = "http://ip-api.com/json/{$ipAddress}?fields=country";
    $response = file_get_contents($url);
    $data = json_decode($response, true);
    return $data['country'];
}
```

Рисунок 3.3 – Функція getCountryFromIP($ipAddress)

Таким чином, сервіс ip-api.com дозволяє отримати інформацію про країну користувача на основі його IP-адреси та відображати цю інформацію на вебсторінці.

Для отримання країни користувача крім сервісу ip-api.com існують інші підходи та інструменти. Деякі з них включають:

– веббраузери можуть надавати інформацію про розташування користувача за допомогою HTML5 Geolocation API. Цей API дозволяє отримати географічні координати користувача, які потім можна перетворити на країну за допомогою геокодування;

– крім ip-api.com, існує безліч інших сервісів, що надають інформацію про місцеперебування на основі IP-адреси. Деякі популярні послуги включають MaxMind, GeoIP, IPstack, і GeoIP2;

– для визначення розташування користувача на сервері можна використовувати спеціальні модулі та бібліотеки. Наприклад, для мови програмування PHP існують модулі GeoIP та GeoIP2, які дозволяють визначити країну за IP-адресою;

– деякі DNS-сервери надають інформацію про географічне розташування IP-адреси. Шляхом виконання спеціальних DNS-запитів можна отримати країну, пов'язану з певною IP-адресою;

– якщо система надає можливість входу через соціальні мережі, можна отримати інформацію про країну користувача через API відповідних соціальних мереж. Наприклад, Facebook, Twitter і Google надають дані про розташування користувачів, які можуть бути використані для визначення країни.

У виборі методу визначення країни слід враховувати вимоги системи, доступність та точність даних, а також можливості інтеграції та обмеження, пов'язані з кожним методом.

Для визначення браузера користувача у системі використовується функція *get_browser()* (рис. 3.4). Ця функція є вбудованою функцією PHP і надає інформацію про браузер, який використовується клієнтом для доступу до вебсторінки.

```
17  $userAgent = $_SERVER['HTTP_USER_AGENT'];
18  $browser = get_browser(null, true)['browser'];
```

Рисунок 3.4 – Функція get_browser()

Коли клієнт надсилає запит на сервер для завантаження вебсторінки, клієнт включає запит заголовок User-Agent, який містить інформацію про браузер та інші властивості клієнта. Сервер отримує заголовок User-Agent із запиту та передає його в PHP-скрипт для обробки. Функція get_browser() аналізує переданий заголовок User-Agent та повертає масив з інформацією про браузер клієнта, таку як ім'я браузера, версія, платформа та інші властивості.

Для визначення браузера користувача у вебсередовищі існує кілька способів. Ось деякі з них:

– отримати інформацію про браузер користувача із заголовка User-Agent, який надсилається браузером у HTTP-запиті. Заголовок User-Agent містить інформацію про браузер, операційну систему та інші параметри. Ви можете отримати доступ до цього заголовка на серверній стороні, використовуючи змінну $_SERVER['HTTP_USER_AGENT'] у PHP або аналогічні засоби в інших мовах програмування;

– використання у клієнтському JavaScript об'єкту navigator для отримання інформації про браузер користувача. Деякі з полів об'єкта navigator включають navigator.userAgent, navigator.vendor і navigator.platform, які містять інформацію про браузер, виробника та операційну систему відповідно;

– бібліотеки та сервіси, які надають API для визначення браузера користувача. Деякі з найпопулярніших сервісів включають WhichBrowser та BrowserStack.

Важливо, що визначення браузера на основі User-Agent не є абсолютно надійним, оскільки користувачі можуть змінювати або приховувати цей заголовок. Також варто пам'ятати про конфіденційність даних користувача та забезпечувати відповідність політикам конфіденційності та правилам використання даних під час збирання та відстеження інформації про браузер користувача.

Для визначення часу, проведеного на сайті, у цьому системи використовується наступний механізм:

У файлі *login.php* після успішного входу користувача до сесії зберігається час входу (рис. 3.5).

```
$_SESSION['login_time'] = date("Y-m-d H:i:s");
```

Рисунок 3.5 – Зберігання часу входу

У файлі *activity.php* відбувається отримання часу проведеного на сторінці шляхом обчислення різниці між поточним часом та часом входу користувача (рис. 3.6).

```
$loginTime = $_SESSION['login_time'];
$logoutTime = date("Y-m-d H:i:s");
$duration = strtotime($logoutTime) - strtotime($loginTime);
```

Рисунок 3.6 – Обчислення різниці між поточним часом та часом входу користувача

Потім, у файлі *activity.php,* на вебсторінці в розділі * 00:00:00 * відображається таймер, який оновлюється кожну секунду за допомогою JavaScript (рис. 3.7).

```
151  setInterval(updateTimer, 1000);
152
153    function updateTimer() {
154        var timerElement = document.getElementById("timer");
155        var clickCounterElement = document.getElementById("click-counter");
156
157        var duration = new Date().getTime() - new Date("<?php echo $_SESSION['login_time']; ?>").getTime();
158        var hours = Math.floor(duration / 3600000);
159        var minutes = Math.floor((duration % 3600000) / 60000);
160        var seconds = Math.floor((duration % 60000) / 1000);
161
162        // Обновление таймера
163        timerElement.textContent = formatTime(hours) + ":" + formatTime(minutes) + ":" + formatTime(seconds);
164    }
165
166    function formatTime(time) {
167        return time < 10 ? "0" + time : time;
168    }
```

Рисунок 3.7 – Оновлення таймера кожної секунди

Таким чином, при вході користувача зберігається час входу в сесію, а при виході або відстеження активності на сторінці обчислюється різниця між часом входу та поточним часом для визначення часу, проведеного на сайті. Цей час відображається на сторінці у вигляді таймера.

Окрім вже використаного методу з використанням JavaScript та таймера, є й інші методи, які можна застосувати:

– аналізувати серверні логи для визначення часу, коли користувач робить запити до вебсервера. При кожному запиті можна записувати тимчасову мітку та обчислювати різницю між послідовними запитами для отримання часу, проведеного користувачем на сайті. Однак цей метод може бути складним для реалізації та вимагає доступу до серверних логів;

– використовувати події браузера, такі як "unload" або "beforeunload", щоб відстежувати, коли користувач залишає сторінку. При спрацюванні такої події можна записати тимчасову мітку та обчислити різницю між часом входу та часом залишення сторінки. Це дозволить отримати приблизний час на сайті. Однак, це рішення не буде точним, якщо користувач просто закриє вкладку або браузер без явного залишення сторінки;

– використовувати cookie або локальне сховище браузера для збереження тимчасової мітки під час входу користувача на сайт і порівнювати її з поточним часом при кожному запиті сторінки. Це дозволить визначити час, проведений

користувачем на сайті. Однак користувач може видалити cookie або очистити локальне сховище, що призведе до втрати даних про час, проведений на сайті.

Для підрахунку кліків, зроблених користувачем на сайті, у системі у файлі *activity.php* знаходиться змінна *clickCounter*, яка ініціалізується значенням *<?php echo $_SESSION['clicks']; ?>*. Це значення виходить із сесії та представляє лічильник кліків користувача. Далі, в цьому ж файлі доданий обробник події кліка (рис. 3.8)

```
138   var clickCounter = <?php echo $_SESSION['clicks']; ?>;
139
140   function updateClickCounter() {
141     clickCounter++;
142     document.getElementById("click-counter").textContent = clickCounter.toString();
143   }
144
145   document.addEventListener("click", updateClickCounter);
```

Рисунок 3.8 – Обробник події кліка

Цей код збільшує значення clickCounter при кожному натисканні користувача та оновлює відповідний елемент з ідентифікатором click-counter на сторінці, відображаючи поточне значення лічильника.

Існує кілька інших способів відстеження кліків, зроблених користувачем на сайті:

– JavaScript для створення подій, які будуть спрацьовувати при натисканні на певні елементи або посилання на вашому сайті. Наприклад, можна додати обробники подій до кнопок, посилань або зображень і виконувати необхідні дії при кожному натисканні;

– Heatmap допомагають візуалізувати активність користувачів на сторінці, відображаючи гарячі та холодні області, які показують, де користувачі найбільше натискають. Це дозволяє зрозуміти, які елементи на сторінці привертають більше уваги і де користувачі виявляють найбільший інтерес;

– аналізувати журнали сервера, щоб побачити записи про кожен запит до вашого сайту, включаючи інформацію про кліки та дії користувачів. Цей метод

вимагає більше технічних навичок і може бути корисним для глибшого аналізу та відстеження користувальницької активності.

Вибір відповідного методу залежить від потреб та вимог системи. Часто використовується комбінація різних методів для отримання повнішої картини про поведінку користувачів на сайті.

Для класифікації користувача після виходу на "активний" та "пасивний" за час проведений на сайті та кількістю кліків використовується алгоритм k-найближчих сусідів.

Алгоритм k-ближніх сусідів є одним з найпростіших і широко використовуваних алгоритмів навчання без учителя та класифікації. Він використовується для розв'язання задач класифікації, регресії та пошуку аномалій в наборах даних.

На початку файлу logout.php підключається бібліотека PHP-ML (рис. 3.9). Це дозволяє використовувати алгоритм k-найближчих сусідів, який надається цією бібліотекою.

```
require 'vendor/autoload.php';
```

Рисунок 3.9 – Підключення бібліотеки PHP-ML

Далі, після підключення бібліотеки, визначається функція classifyUser(), яка виконує класифікацію користувача на основі кількості кліків та часу, проведеного на сайті. Усередині функції наведено використання алгоритму k-найближчих сусідів. У коді з використанням бібліотеки PHP-ML та алгоритму k-найближчих сусідів значення k вказано не явно. За замовчуванням використовується значення k = 3, якщо інше не вказано. Функція classifyUser() викликається для класифікації користувача та визначення поведінки (активний чи пасивний). Результат класифікації зберігається у змінній $behavior (рис. 3.10).

```php
function classifyUser($clicks, $duration) {
        $samples = [
          [10, 20],
          [3, 10],
      ];
      $labels = [
          'active',
          'passive',
      ];
      $classifier = new Phpml\Classification\KNearestNeighbors();
      $classifier->train($samples, $labels);
      $prediction = $classifier->predict([$clicks, $duration]);

      return $prediction;
}

$behavior = classifyUser($clicks, $duration);
```

Рисунок 3.10 – Функція classifyUser()

3.2 Опис програмної реалізації

Щоб розпочати роботу інтелектуальної системи збору та аналізу даних користувача у вебсередовищі для початку треба встановити Openserver для можливості розгорнути локальний сервер (рис. 3.11).

OpenServer – це пакет програмного забезпечення, який надає середовище для локальної розробки та запуску вебзастосунків на комп'ютері. Він включає всі необхідні компоненти для роботи вебсервера, такі як вебсервер Apache, бази даних MySQL і PHP-інтерпретатор.

OpenServer дозволяє створювати та тестувати вебпрограми на локальному комп'ютері без необхідності підключення до віддаленого сервера. Це зручно для розробників, оскільки вони можуть працювати над проєктами в локальному середовищі, перш ніж розгорнути програму на віддаленому сервері.

OpenServer забезпечує просте встановлення та налаштування всіх компонентів сервера та надає зручний інтерфейс для управління сервером та налаштуваннями проєктів.

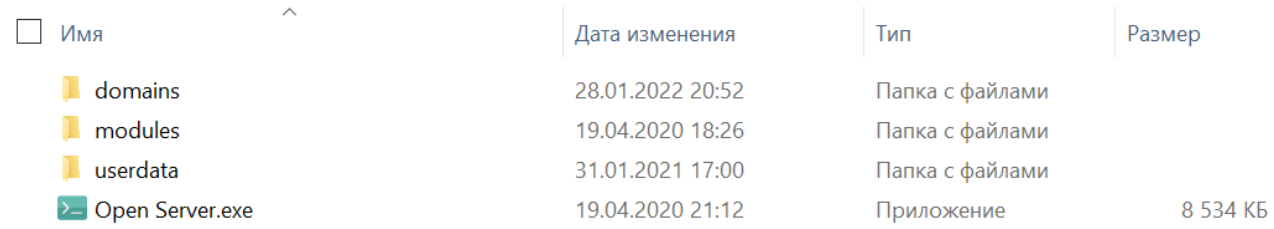

Рисунок 3.11 – Директорія з встановленим Openserver

Далі створюємо проєкт «webanalytics» у домені localhost (рис. 3.12). В ньому будуть знаходитися файли сторінок зі скриптами PHP, HTML розміткою та CSS стилями.

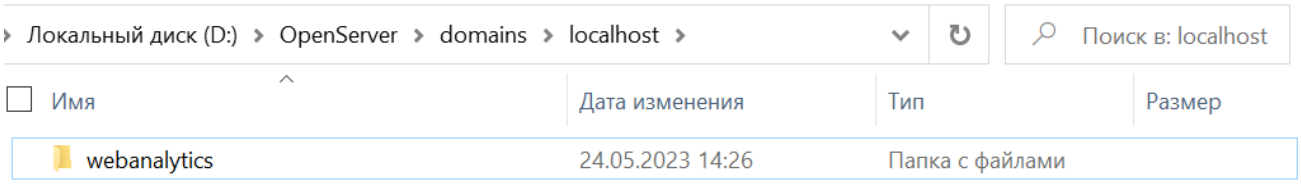

Рисунок 3.12 – Створений проєкт “webanalytics” у домені localhost

Запускаємо локальний сервер. Про те ще сервер успішно ввімкнено свідчить «зелений прапорець» в панелі управління (рис. 3.13).

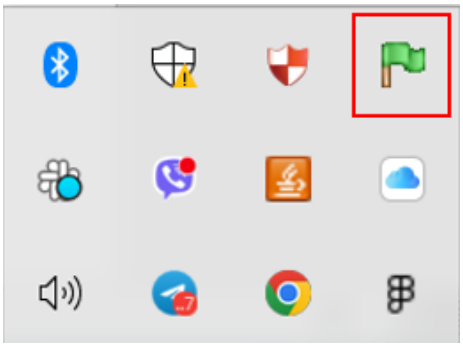

Рисунок 3.13 – Успішний запуск роботи сервера

Тепер треба створити бази даних у phpMyAdmin. Першим кроком розробки буде створення бази даних у phpMyAdmin. Для цього потрібно відкрити вкладку «додатково» та обрати «PhpMyAdmin» (рис. 3.14).

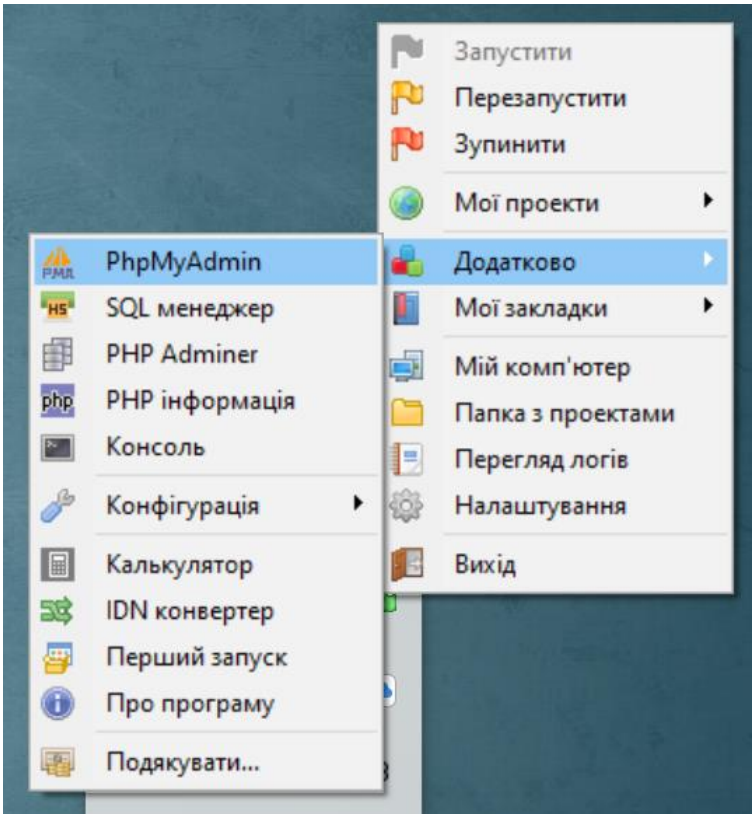

Рисунок 3.14 – Запуск PhpMyAdmin

PhpMyAdmin – це безкоштовний вебдодаток з відкритим вихідним кодом, призначений для управління базами даних MySQL через вебінтерфейс. Воно надає інтерфейс користувача, який дозволяє адміністраторам баз даних виконувати різні завдання, такі як створення, видалення та зміна таблиць, виконання SQL-запитів, імпорт та експорт даних, управління користувачами та привілеями, а також багато інших операцій, пов'язаних з управлінням базами даних.

PhpMyAdmin дозволяє адміністраторам зручно працювати з базами даних, не вимагаючи безпосереднього доступу до сервера баз даних або знання SQL-команд. Він надає графічний інтерфейс, який спрощує виконання завдань та надає інструменти для візуального представлення даних у таблицях та запитах.

PhpMyAdmin широко використовується у веброзробці та адмініструванні баз даних, особливо в середовищі, де використовується MySQL. Він надає зручні інструменти для роботи з базами даних та дозволяє адміністраторам ефективно управляти та підтримувати свої бази даних.

Щоб отримати доступ до управління базами даних треба ввести пароль та логін (стандартний пароль та логін – root, root) (рис. 3.15).

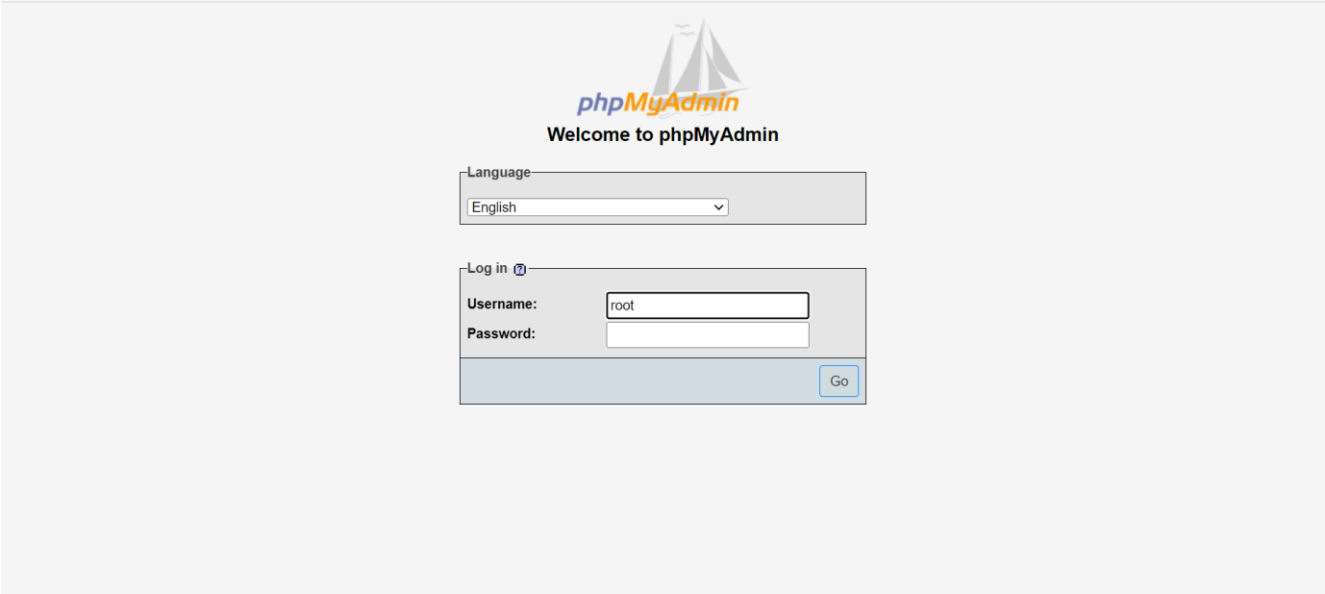

Рисунок 3.15 – Вхід до PhpMyAdmin

Тепер треба створити базу даних, в якій автоматично буде створено таблиці «user» та «activity» (рис. 3.16).

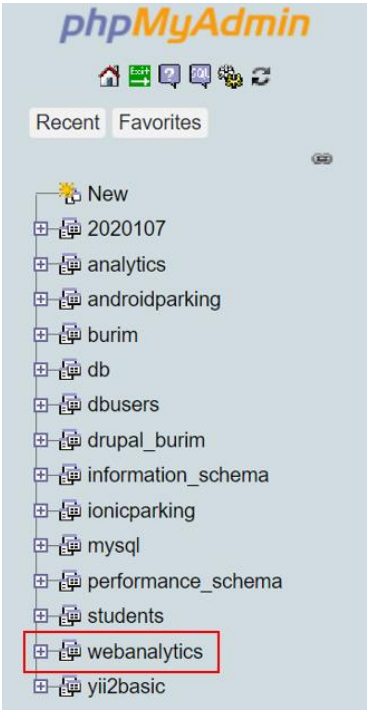

Рисунок 3.16 – База даних «webanalytics»

Тепер для успішного підключення до бази даних потрібно у файлах login.php, register.php, logout.php та activity.php вказати сервер, дані облікового запису, де створено базу даних для збору даних про поведінку користувача у вебсередовищі та назву бази даних (рис. 3.17).

```php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "webanalytics";

$conn = new mysqli($servername, $username, $password, $dbname);
```

Рисунок 3.17 – Підключення до бази даних

На сторінці входу вводимо ім’я та пароль (рис. 3.18).

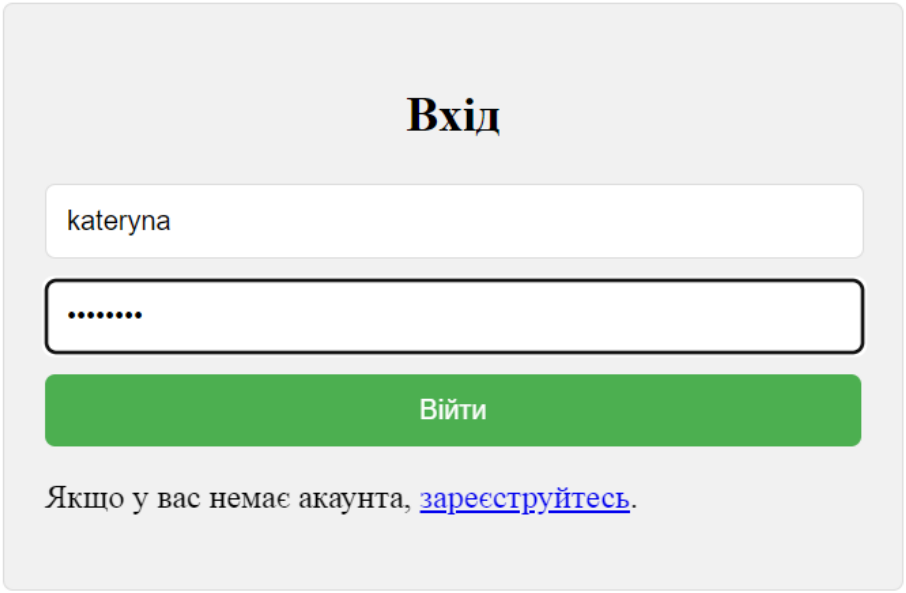

Рисунок 3.18 – Сторінка входу

Так як такого користувача не було зареєстровано, переходимо на сторінку реєстрації, за кнопкою «зареєструватися» через сторінку входу. На сторінці реєстрації вводимо ім’я та пароль, та натискаємо кнопку «Зареєструватися» (рис. 3.19).

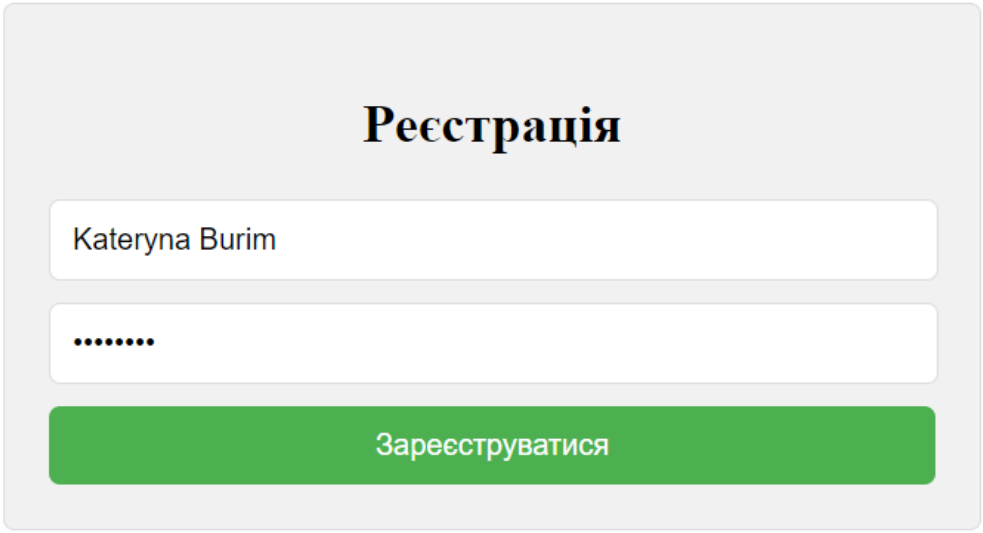

Рисунок 3.19 – Сторінка реєстрації

Якщо реєстрація пройшла успішно отримуємо повідомлення з проханням увійти з використанням нового облікового запису (рис. 3.20).

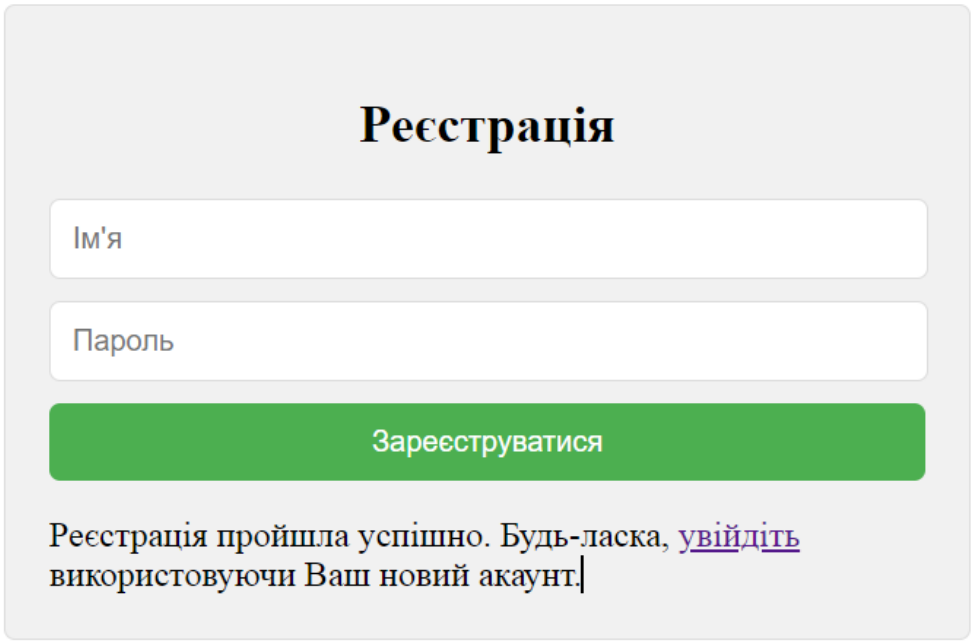

Рисунок 3.20 – Сторінка реєстрації

Перейдемо до бази даних та побачимо, що таблиця «users» була створена автоматично при створенні нового облікового запису (рис. 3.21). Дані користувача вказано вірно та занесено до таблиці (рис. 3.22).

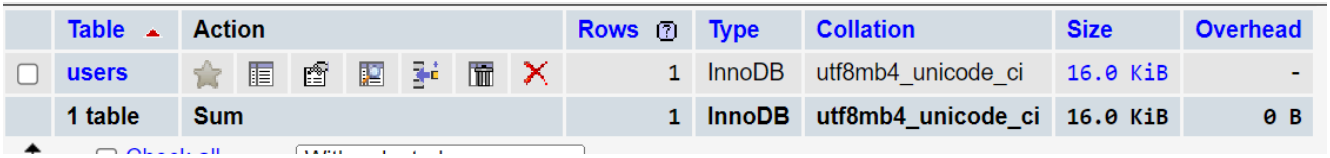

Рисунок 3.21 – Автоматичне створення таблиці «users»

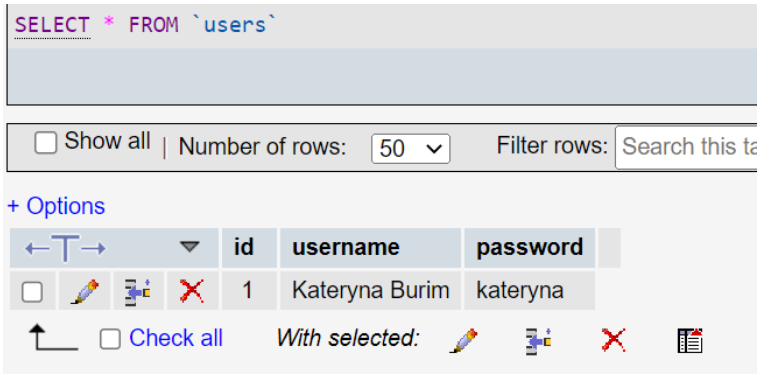

Рисунок 3.22 – Дані облікового запису зареєстрованого користувача

Виконаємо повторний вхід, з використанням нового облікового запису (рис. 3.23).

Рисунок 3.23 – Сторінка входу

Потрапляємо на сторінку відстеження активності. На ній відображено 4 картки: таймер(запускається з моменту входу на сторінку активності та зупиняється після виходу з неї), лічильник кліків, зроблених на сторінці після входу, назва браузера за допомогою, якого відвідувач зайшов на сайт та країна в за IP-адресою відвідувача (рис. 3.24).

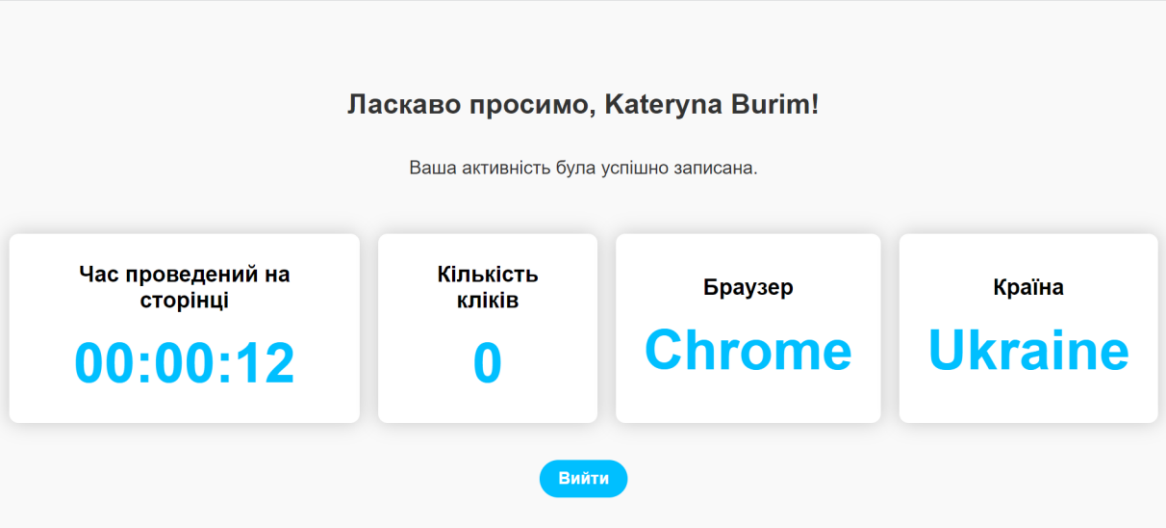

Рисунок 3.24 – Сторінка відстеження активності

Зафіксуємо активність на сайті та натиснемо кнопку «Вийти» (рис. 3.25).

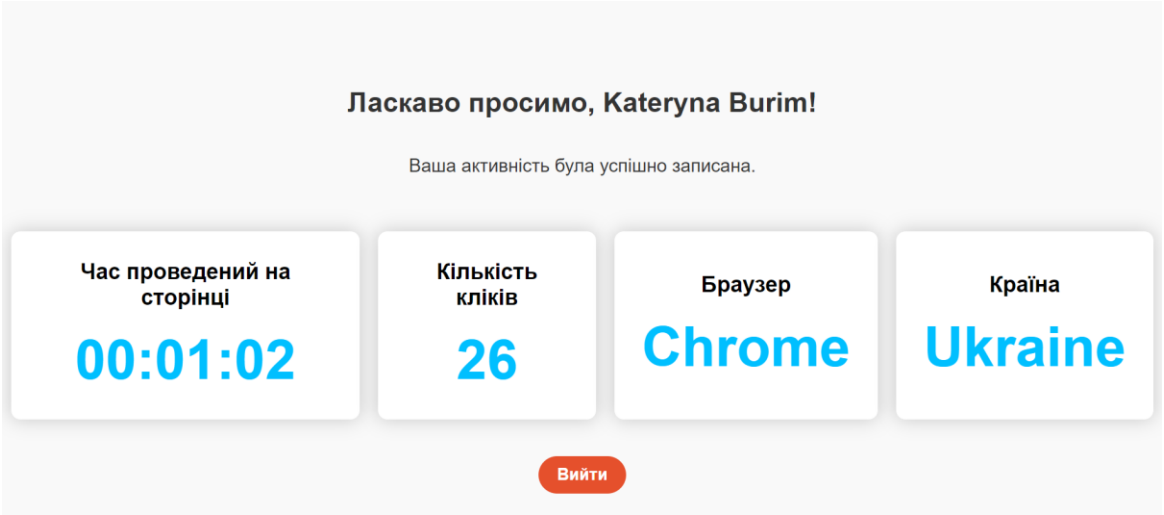

Рисунок 3.25 – Вихід зі сторінки та фіксація активності

Перейдемо до бази даних та побачимо, що таблиця «activity» була створена автоматично при вході користувача на сторінку активності (рис. 3.26). Дані активності користувача вказано вірно та занесено до таблиці (рис. 3.27).

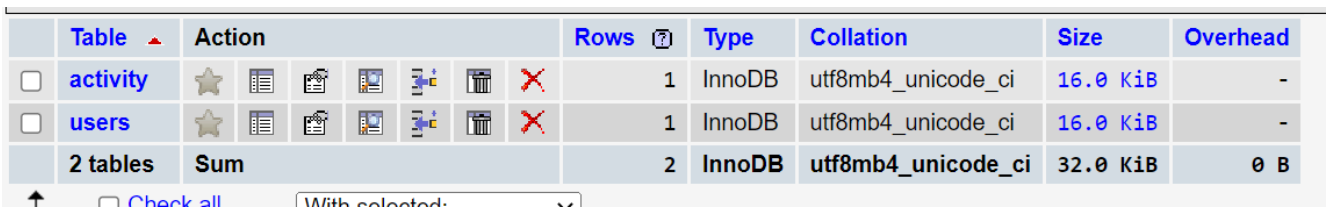

	Table	Action	Rows	Type	Collation	Size	Overhead
☐	activity		1	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
☐	users		1	InnoDB	utf8mb4_unicode_ci	16.0 KiB	-
	2 tables	Sum	2	InnoDB	utf8mb4_unicode_ci	32.0 KiB	0 B

Рисунок 3.26 – Автоматичне створення таблиці «activity»

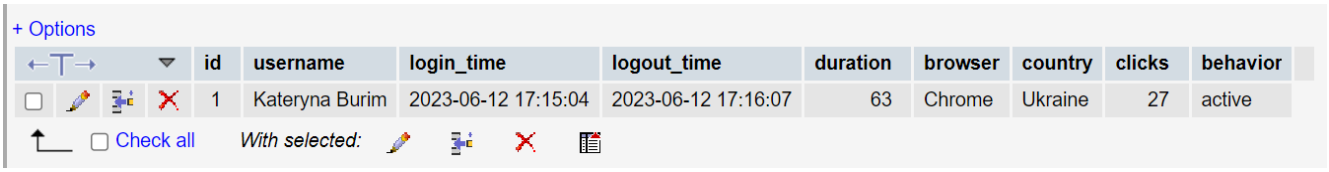

Рисунок 3.27 – Дані активності користувача

Висновки до розділу 3

В результаті написання третього розділу було визначено план роботи інтелектуальної системи зроби даних та аналізу поведінки користувача. Ідея полягає в тому, що це є системою автентифікації та збору даних про поведінку користувача у вебсередовищі, яка може бути доопрацьована та розширена відповідно до конкретних вимог та цілей проєкту. Ця система використовує сесії для відстеження унікальності користувача та збереження його стану протягом сеансу взаємодії з вебпрограмою.

Було описано програмну реалізацію інтелектуальної системи збору та аналізу даних користувача у вебсередовищі. Зареєстровано новий обліковий запис, який було успішно занесено до бази даних, та відстежено активність цього облікового запису на сторінці активності.

ВИСНОВКИ

В ході виконання бакалаврської роботи було розроблено інтелектуальну систему для збору та аналізу даних поведінки користувача у вебсередовищі. Вона дозволяє збирати та зберігати інформацію про активність користувачів, включаючи ім'я користувача, час входу, час виходу, тривалість перебування на сторінці та кількість кліків, використання API GeoIP для визначення країни користувача на основі його IP-адреси, збереження країни користувача разом з іншою інформацією про активність у базі даних, відображення стану активності користувачів у вебінтерфейсі, включаючи час перебування на сторінці та кількість кліків.

В основу розробки було покладено можливість масштабованості даної системи та проєкту в цілому.

Результати дослідження показують, що можуть знайти застосування в різних галузях, пов'язаних з вебресурсами та інтернет-технологіями. Наприклад, вони можуть бути використані в маркетингових дослідженнях та аналізі вебтрафіку для поліпшення конверсії сайту, розробці інтернет-реклами та збільшення прибутку компаній, що працюють в онлайн-бізнесі. Крім того, інтелектуальна система збору даних та аналізу поведінки користувача може бути корисною для оптимізації роботи вебсервісів та покращення користувацького досвіду в інтернеті. В сфері наукових досліджень інтелектуальна система може бути використана для аналізу поведінки користувачів на вебсайтах, що дозволить покращити розуміння їхньої поведінки та реакції на різні вебресурси.

В БКР було описано значущість даної роботи та необхідність розвитку обраної сфери дослідження, яка полягає у покращені ефективними, використанні та доступності кожного користувача.

Для досягнення поставленої мети виконано наступні завдання:

– проведено аналіз в потребі дослідження поведінки користувача у вебсередовищі;

– проведено аналіз технічних можливостей та інструментів для збору даних користувачів у вебсередовищі;

– визначено параметри та метрики, які необхідно збирати для аналізу поведінки користувачів;

– розроблено алгоритм для аналізу поведінки користувачів, використовуючи зібрані дані;

– проведено експериментальну перевірку розробленої системи.

Робота складається з вступу, трьох фахових розділів, висновків, переліку джерел посилання з 28 джерел, одного додатку та спеціальної частини з питань охорони праці та безпеки життєдіяльності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Simon Kemp. Digital 2022: global overview report (January 26th 2022) https://hootsuite.widen.net/s/gqprmtzq6g/digital-2022-globaloverview-report.

2. Data Never Sleeps 8.0. URL: https://www.domo.com/learn/data-never-sleeps-8.

3. Lovett, J. 2009. US Web Analytics Forecast, 2008 To 2014. Cambridge, MA: Forrester Research.

4. Hu, X., & Cercone, N. 2004. A Data Warehouse/Online Analytic Processing Framework for Web Usage Mining and Business Intelligence Reporting. International Journal of Intelligent Systems, 19(7), p. 585–606.

5. TagMan. (2012, March 14). Website speed matters: Study finds 1 second delay in website load time means a 7% reduction in conversions – Режим доступу https://www.liveseysolar.com/website-speed-study-finds-1-second-delay-in-website-load-time-means-a-7-reduction-in-conversions/ – Дата доступу : 04.05.2023.

6. Burby, J., & Brown, A. (2007, August 16). Web Analytics Definitions - Version 4.0. – Режим доступу http://www.digitalanalyticsassociation.org – Дата доступу : 05.05.2023.

7. Tappenden, A. F., & Miller, J. (2009). Cookies: A Deployment Study and the Testing Implications. ACM Trans. Web, 3(3), 9:1–9:49.

8. Ganapathi, A., & Zhang, S. (2011). Web Analytics and the Art of Data Summarization. In Managing Large-scale Systems via the Analysis of System Logs and the Application of Machine Learning Techniques (pp. 6:1–6:9). New York, NY, USA: ACM.

9. Clifton, B. (2012). Advanced Web Metrics with Google Analytics (3rd ed.). Indianapolis, IN: John Wiley & Sons., p. 11-16.

10. Zampatt, G. (2011, September). SharePoint Best Practices Creating and Configuring Service Applications With (and Without) PowerShell, Part2. The SolidQ Journal, 13. -– Режим доступу http://www.solidq.com/sqj/Pages/2011-September-Issue/ – Дата доступу : 09.05.2023.

10. Peterson, E., & Carrabis, J. (2008). Measuring the Immeasurable: Visitors Engagement. Web Analytics Demystified. -– Режим доступу http://www.webanalyticsdemystified.com/downloads/Web_Analytics_Demystified_and_NextStage_Global_-_Measuring_the_Immeasurable_-_Visitor_Engagement.pdf – Дата доступу : 10.05.2023.

11. Li Hairong. The Impact of Perceived Channel Utilities, Shopping Orientations and Demographics on the Consumer's Online Buying Behavior / Li Hairong, Cheng Kuo, Martha G. Russell. // Journal of Computer Medicated Communication. — 1999. — № 5(2). — С. 1-23.

12. Opentracker. (2011). Glossary. – Режим доступу http://www.opentracker.net/glossary – Дата доступу : 10.05.2023.

13. Schwartz, M. J. (2012, October 30). Yahoo To Ignore IE10 DNT Settings. InformationWeek. – Режим доступу https://www.eurchembull.com/uploads/paper/90e3c2b778e4f72fd0bbbde4c9fcf53a.pdf – Дата доступу : 12.05.2023.

14. Meeker, M. (2012). Internet Trends. – Режим доступу https://www.businessinsider.com/mary-meeker-2012-internet-trends-year-end-update-2012-12 – Дата доступу : 12.05.2023.

15. Rapoza, J. (2010, December 2). Web Analytics: A New View. InformationWeek. – Режим доступу http://www.informationweek.com/webanalytics-a-new-view/d/d-id/1094560 – Дата доступу : 12.05.2023.

16. Stanhope, J. (2012, January 1). The new face of Web analytics. KMWorld Magazine. – Режим доступу https://www.kmworld.com/Articles/Editorial/Features/The-new-face-of-Webanalytics-79583.aspx – Дата доступу : 12.05.2023.

17. Hoofnagle, Chris Jay. Understanding Privacy. Harvard Law Review, — 2009 — №4 — pp. 745-772.

18. Leon, Pedro Web Personalization Methods Using Cookies.Journal of Web Engineering, — 2017 —№ 1(2) — pp. 97-119.

ДОДАТОК А

Програмний код усіх сторінок системи

index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Збір даних поведінки користувача</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <div class="container">
    <h2>Вхід</h2>
    <form action="login.php" method="POST">
      <input type="text" name="username" placeholder="Ваше ім'я" required>
      <input type="password" name="password" placeholder="Пароль" required>

      <button type="submit">Війти</button>
    </form>
    <p>Якщо у вас немає акаунта,
        <a href="registration.html">зареєструйтесь</a>.</p>
  </div>
</body>
</html>
```

login.php

```
<?php session_start();

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "webanalytics";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

```
    $username = $_POST["username"];
$password = $_POST["password"];
    $checkUserQuery = "SELECT * FROM users WHERE username='$username'
AND password='$password'";
    $result = $conn->query($checkUserQuery);

    if ($result->num_rows > 0) {
        $_SESSION['login_time'] = date("Y-m-d H:i:s");
        $_SESSION['username'] = $username;
        header("Location: activity.php");
        exit;
    } else {
          $_SESSION['error_message'] = "Данного пользователя не существует";
        header("Location: index.html?login=failed");
        exit;
    }
}
$conn->close();
?>
```

registration.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Реєстрація</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <div class="container">
    <h2>Реєстрація</h2>

    <form action="register.php" method="POST">
      <input type="text" name="username" placeholder="Ім'я" required>
      <input type="password" name="password" placeholder="Пароль" required>
      <button type="submit">Зареєструватися</button>
    </form>
  </div>
</body>
</html>
```

register.php

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "webanalytics";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}

$checkTableQuery = "SHOW TABLES LIKE 'users'";
$result = $conn->query($checkTableQuery);
if ($result->num_rows == 0) {
  $createTableQuery = "CREATE TABLE users (
    id INT(11) AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(255) NOT NULL
  )";
  if ($conn->query($createTableQuery) !== TRUE) {
    echo "Помилка при створенні таблиці: " . $conn->error;
    $conn->close();
    exit;
  }
}

$message = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $username = $_POST["username"];
  $password = $_POST["password"];

  $checkUserQuery = "SELECT * FROM users WHERE username='$username'";
  $result = $conn->query($checkUserQuery);
  if ($result->num_rows > 0) {
    $message = "Користувач з таким ім'ям вже існує.";
  } else {
    $insertUserQuery = "INSERT INTO users (username, password) VALUES
('$username', '$password')";
```

```
        if ($conn->query($insertUserQuery) === TRUE) {
          $message = "Реєстрація пройшла успішно. Будь-ласка, <a
href='index.html'>увійдіть</a> використовуючи Ваш новий акаунт.";
        } else {
          $message = "Помилка при реєстрації: " . $conn->error;
        }
      }
    }

    $conn->close();
    ?>

    <!DOCTYPE html>
    <html>
    <head>
      <title>Реєстрація</title>
      <link rel="stylesheet" type="text/css" href="styles.css">
    </head>
    <body>
      <div class="container">
        <h2>Реєстрація</h2>
    <form action="register.php" method="POST">
          <input type="text" name="username" placeholder="Ім'я" required>
          <input type="password" name="password" placeholder="Пароль" required>
          <button type="submit">Зареєструватися</button>
      <p></p>
        </form>
        <?php echo $message; ?>
```

logout.php

```
    <?php
    session_start();
    require 'vendor/autoload.php';
    $servername = "localhost";
    $username = "root";
    $password = "";
    $dbname = "webanalytics";

    $conn = new mysqli($servername, $username, $password, $dbname);
```

```
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}

$loginTime = $_SESSION['login_time'];
$logoutTime = date("Y-m-d H:i:s");
$duration = strtotime($logoutTime) - strtotime($loginTime);
$clicks = $_SESSION['clicks'];

$username = $_SESSION["username"];

$userAgent = $_SERVER['HTTP_USER_AGENT'];
$browser = get_browser(null, true)['browser'];


$ipAddress = $_SERVER['REMOTE_ADDR'];
$country = getCountryFromIP($ipAddress);


function getCountryFromIP($ipAddress) {
   $url = "http://ip-api.com/json/{$ipAddress}?fields=country";
   $response = file_get_contents($url);
   $data = json_decode($response, true);
   return $data['country'];
}

function classifyUser($clicks, $duration) {
      $samples = [
       [10, 20],
       [3, 10],
   ];
   $labels = [
      'active',
      'passive',
   ];
   $classifier = new Phpml\Classification\KNearestNeighbors();
   $classifier->train($samples, $labels);
   $prediction = $classifier->predict([$clicks, $duration]);

   return $prediction;
```

```
}

$behavior = classifyUser($clicks, $duration);

$query = "INSERT INTO activity (username, login_time, logout_time, duration,
browser, country, clicks, behavior)
          VALUES ('$username', '$loginTime', '$logoutTime', '$duration', '$browser',
'$country', '$clicks', '$behavior')";
$conn->query($query);
session_unset();
session_destroy();

header("Location: index.html");
exit;
?>
```

activity.php

```
<?php session_start();

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "webanalytics";

$conn = new mysqli($servername, $username, $password, $dbname);

$userAgent = $_SERVER['HTTP_USER_AGENT'];
$browser = get_browser(null, true)['browser'];

$ipAddress = $_SERVER['REMOTE_ADDR'];
$country = getCountryFromIP($ipAddress);

function getCountryFromIP($ipAddress) {
   $url = "http://ip-api.com/json/{$ipAddress}?fields=country";
   $response = file_get_contents($url);
   $data = json_decode($response, true);
   return $data['country'];
}
```

```
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}

$checkTableQuery = "SHOW TABLES LIKE 'activity'";
$result = $conn->query($checkTableQuery);
if ($result->num_rows == 0) {
   $createTableQuery = "CREATE TABLE activity (
      id INT(11) AUTO_INCREMENT PRIMARY KEY,
      username VARCHAR(50) NOT NULL,
      login_time DATETIME,
      logout_time DATETIME,
      duration INT(11),
        browser VARCHAR(50),
        country VARCHAR (100),
        clicks INT(50),
        behavior VARCHAR(50)
   )";
   if ($conn->query($createTableQuery) !== TRUE) {
      echo "Ошибка при создании таблицы: " . $conn->error;
      $conn->close();
      exit;
   }
}

$username = $_SESSION['username'];
$loginTime = $_SESSION['login_time'];
$logoutTime = date("Y-m-d H:i:s");
$duration = strtotime($logoutTime) - strtotime($loginTime);
$_SESSION['clicks'] = 0;
$clicks = $_SESSION["clicks"];

$username = $_SESSION["username"];

$conn->close();
?>
<!DOCTYPE html>
<html>
<head>
   <title>Відстеження активності користувача</title>
```

```
    <link rel="stylesheet" type="text/css" href="style_activity.css">
</head>
<body>
<body>
<div class="container">
    <header>
        <h1>Відстеження активності користувача</h1>
    </header>

    <section class="activity-section">
        <h2>Ласкаво просимо, <?php echo $username; ?>!</h2>
        <p>Ваша активність була успішно записана.</p>
 <div class="cards">
        <div class="card">
            <div class="card-title">Час проведений на сторінці</div>
            <div id="timer" class="card-number">00:00:00</div>
        </div>

        <div class="card">
            <div class="card-title">Кількість кліків</div>
            <div id="click-counter" class="card-number">0</div>
        </div>
          <div class="card">
                    <div class="card-title">Браузер</div>
                    <div id="browser-info" class="card-number"></div>
          </div>
                        <div class="card">
                    <div class="card-title">Країна</div>
                    <div id="country-info" class="card-number"></div>
          </div>
 </div>
        <form action="logout.php" method="POST">
            <button type="submit" class="logout-btn"> Вийти</button>
        </form>

    </section>

    <footer>
        <p>&copy; <?php echo date("Y"); ?> Все права защищены.</p>
    </footer>
```

```
        </div>
        <script>
           setInterval(updateTimer, 1000);

          var clickCounter =<?php echo $_SESSION['clicks']; ?>;

          function updateClickCounter() {
            clickCounter++;

            document.getElementById("click-counter").textContent =
clickCounter.toString();

          }


          document.addEventListener("click", updateClickCounter);

            function updateTimer() {
               var timerElement = document.getElementById("timer");
               var clickCounterElement = document.getElementById("click-counter");

               var duration = new Date().getTime() - new Date("<?php echo
$_SESSION['login_time']; ?>").getTime();
               var hours = Math.floor(duration / 3600000);
                 var minutes = Math.floor((duration % 3600000) / 60000);
               var seconds = Math.floor((duration % 60000) / 1000);

               timerElement.textContent = formatTime(hours) + ":" +
formatTime(minutes) + ":" + formatTime(seconds);


            }

        var browserInfoElement = document.getElementById("browser-info");
        var browser = "<?php echo $browser; ?>";
        browserInfoElement.textContent = browser;

        var countryInfoElement = document.getElementById("country-info");
        var country = "<?php echo $country; ?>";
        countryInfoElement.textContent = country;
```

```
    function formatTime(time) {
        return time < 10 ? "0" + time : time;
    }
</script>
</body>
</html>
```

style-activity.css

```
body {
  background-color: #f5f5f5;
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
}

.container {
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            height: 100vh;
            background-color: #f9f9f9;
        }
          .cards {
                    display: flex;
            flex-direction: row;

          }
        .card {
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            padding: 30px;
            background-color: #fff;
            box-shadow: 0px 0px 20px 0px rgba(0, 0, 0, 0.2);
            border-radius: 10px;
                    margin: 10px;
            text-align: center;
```

```
    transition: transform 0.3s ease-in-out;
}

.card:hover {
    transform: scale(1.1);
}

.card-title {
    font-size: 24px;
    font-weight: bold;
    margin-bottom: 20px;
    color: #000;
}

.card-number {
    font-size: 60px;
    font-weight: bold;
    color: #00BFFF;
}

.logout-btn {
    display: inline-block;
    padding: 10px 20px;
    background-color: #00BFFF;
    color: #fff;
    font-size: 18px;
    font-weight: bold;
    border-radius: 30px;
    border: none;
    cursor: pointer;
    transition: background-color 0.3s ease-in-out;
    text-decoration: none;
    margin-top: 30px;
}

.logout-btn:hover {
    background-color: #e5502c;
}

.activity-section {
```

```
        display: flex;
        flex-direction: column;
        align-items: center;
        justify-content: center;
        text-align: center;
        margin-top: 50px;
    }

    h1 {
        font-size: 40px;
        font-weight: bold;
        margin-bottom: 30px;
        text-align: center;
        color: #ff5733;
    }

    h2 {
        font-size: 30px;
        font-weight: bold;
        margin-bottom: 20px;
        text-align: center;
        color: #333;
    }

    p {
        font-size: 20px;
        margin-bottom: 50px;
        text-align: center;
        color: #333;
    }

footer {
  text-align: center;
  font-size: 12px;
  color: #999;
  margin-top: 30px;
}
```

styles.css

```
.container {
```

```
  max-width: 400px;
  margin: 200px;
  padding: 20px;
  background-color: #f1f1f1;
  border: 1px solid #ddd;
  border-radius: 5px;
}

h2 {
  text-align: center;
}

form {
  margin-top: 20px;

}

input[type="text"],
input[type="password"] {
  width: 380px;
  padding: 10px;
  margin-bottom: 10px;
  border: 1px solid #ddd;
  border-radius: 5px;
}

button {
  width: 380px;
  padding: 10px;
  background-color: #4CAF50;
  color: #fff;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background-color: #45a049;
}
```

```
p.success {
  color: green;
  text-align: center;
  margin-top: 10px;
}
```