

Київський національний торговельно-економічний університет

Кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка програмного забезпечення моніторингу відвідування web-ресурсів»

Студентки 4 курсу, 13 групи,

спеціальності
122 «Комп'ютерні науки»

_____ *підпис студента*

Пилипенко
Анастасія
Олександрівна

Науковий керівник
кандидат фізико-математичних наук

_____ *підпис керівника*

Філімонова Тетяна
Олегівна

Гарант освітньої програми
кандидат технічних наук, доцент

_____ *підпис керівника*

Демідов Павло
Георгійович

Київ 2021

Київський національний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем
Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри _____ **Затверджую**
Філімонова Т.О
«18» грудня 2020р.

Завдання
на випускню кваліфікаційну роботу (проект) студентці

Пилипенко Анастасія Олександрівна
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)

«Розробка програмного забезпечення моніторингу відвідування web-ресурсів»

Затверджена наказом ректора від «15» грудня 2020 р. № 3780

2. Строк здачі студентом закінченої роботи 31 травня 2021 року

3. Цільова установка та вихідні дані до роботи

Мета роботи: обґрунтування та розробка програмного забезпечення моніторингу відвідування web-ресурсів

Об'єкт дослідження: процес моніторингу відвідування web-ресурсів.

Предмет дослідження: засоби створення програмного забезпечення моніторингу відвідування web-ресурсів

4. Перелік графічного матеріалу _____

5. Консультанти по роботі із зазначенням розділів, за якими здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Філімонова Т.О.	22.12.2020 р.	22.12.2020 р.
2	Філімонова Т.О.	22.12.2020 р.	22.12.2020 р.
3	Філімонова Т.О.	22.12.2020 р.	22.12.2020 р.

6. Зміст випускної кваліфікаційної роботи (проекту) (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ

1.1 Аналіз предметної області

1.2 Огляд етапів проектування програмного забезпечення

1.3 Аналіз варіантів розв'язання досліджуваної задачі

РОЗДІЛ 2. Організація розробки програмного забезпечення моніторингу відвідування web-ресурсів

2.1 Загальна концепція

2.2 Модель програмного забезпечення моніторингу web-ресурсів

2.3 Алгоритм розробки програмного забезпечення

РОЗДІЛ 3. Програмна реалізація

3.1 Програмна реалізація бази даних

3.2 Програмна реалізація проекту

ВИСНОВКИ

ДОДАТКИ

7. Календарний план виконання роботи

№ Пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	фактично
1	2	3	4
1	<i>Вибір теми випускної кваліфікаційної роботи</i>	05.10.2020	05.10.2020
2	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>	18.12.2020	18.12.2020
3	<i>Вступ</i>	03.02.2021	03.02.2021
4	<i>РОЗДІЛ 1. Обґрунтування доцільності розробки</i>	26.02.2021	26.02.2021
5	<i>РОЗДІЛ 2. Організація розробки програмного забезпечення моніторингу відвідування web-ресурсів</i>	04.04.2021	04.04.2021
6	<i>РОЗДІЛ 3. Програмна реалізація</i>	12.05.2021	12.05.2021
7	<i>Висновки</i>	14.05.2021	14.05.2021
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	20.05.2021	20.05.2021
9	<i>Попередній захист випускної кваліфікаційної роботи</i>	26.05.2021	
11	<i>Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи</i>	27.05.2021	
12	<i>Представлення готової зшитої випускної кваліфікаційної роботи на кафедрі</i>	31.05.2021	
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	За розкладом роботи ЕК	

8. Дата видачі завдання «22» грудня 2020 р.

9. Керівник випускної кваліфікаційної роботи (проекту)

Філімонова Т.О.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

Пилипенко А.О.

(прізвище, ініціали, підпис)

12. Відгук керівника випускної кваліфікаційної роботи (проекту)

Керівник випускної кваліфікаційної роботи (проекту)

31.05.2021 р.

(підпис, дата)

13. Висновок про випускну кваліфікаційну роботу (проект)

Випускна кваліфікаційна робота (проект) студента _____

(прізвище, ініціали)

може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми _____

(підпис, прізвище, ініціали)

Демідов П.Г.

Завідувач кафедри _____

(підпис, прізвище, ініціали)

Пурський О.І.

« _____ » 2021 р.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ	10
1.1 Аналіз предметної області.....	10
1.2 Огляд етапів проектування програмного забезпечення.....	13
1.3 Аналіз варіантів розв'язання досліджуваної задачі.....	15
РОЗДІЛ 2. Організація розробки програмного забезпечення моніторингу відвідування web-ресурсів	22
2.1 Загальна концепція.....	22
2.2 Модель програмного забезпечення моніторингу web-ресурсів.....	24
2.3 Алгоритм розробки програмного забезпечення.....	30
РОЗДІЛ 3. Програмна реалізація	33
3.1 Програмна реалізація бази даних.....	33
3.2. Програмна реалізація проекту.....	35
ВИСНОВКИ	42
ДОДАТКИ	45

Анотація

У випускній кваліфікаційній роботі здійснено комплексну розробку моделі та автоматизованої системи моніторингу відвідувачів сайту з метою аналізу поведінки користувачів сайту для оцінювання ефективності функціонування сайту та задоволення інтересів користувача. Представлена робота є спробою аналізу даних веб-лічильників сайту за окремими періодами часу. Теоретично обґрунтовано основні положення формування і проведення впровадження лічильника на сайт. Створено базу даних для формування звітів.

Ключові слова: MySQL, PHP, лічильник, статистика

Anotation

In the final qualifying work, a comprehensive development of a model and automated monitoring system for site visitors to analyze the behavior of site users to assess the effectiveness of the site and meet the interests of the user. This work is an attempt to analyze the data of web counters of the site for certain periods of time. The main provisions of the formation and implementation of the meter on the site are theoretically substantiated. A database for generating reports has been created.

Keywords: MySQL, PHP, counter, statistics

ВСТУП

Лічильник відвідувань - скрипт, що дозволяє вебмайстру переглянути статистику відвідування свого сайту.

Статистика, включає в себе наступну інформацію:

- зведена статистика по відвідуванню (кількість відвідувань, переглядів та їх співвідношення за сьогоднішні та вчорашній дні, тиждень та місяць)
- список сторінок, з яких прийшли відвідувачі
- відвідуваність сайту (погодинно за сьогоднішню та вчорашню добу, подово за місяць та помісячно за рік)
- список відвідувачів
- переглянуті сторінки Вашого ресурсу за два дні
- порівняння відвідуваності Вашого сайту з будь-яким іншим, що приймає участь у рейтингу

Актуальність теми полягає в тому, що робота подібних сервісів заснована на аналізі поведінки відвідувачів, по яких посиланнях переходять люди найчастіше, на що звертають свою увагу частіше, після яких сторінок найчастіше потенційні клієнти переходять в розділ покупок, які сторінки знижують відвідуваність, вся ця інформація обробляється відповідними програмами і відсилається власнику сайту. Це значно підвищує шанси на успішний бізнес в інтернеті. І, як висновок, ця сфера розробки дуже затребувана в сучасному світі, який все більше переходить в інтернет простір.

Практичне значення роботи: за допомогою моніторингу відвідувань сайтів можна в реальному часі відстежувати, які веб-ресурси відвідували користувачі, а також дізнаватися на яких вони знаходяться в даний момент. Для перегляду доступні URL-адреси, час відкриття і час активності на сторінці.

Метою роботи є обґрунтування та розробка програмного забезпечення моніторингу відвідування web-ресурсів. Для досягнення поставленої мети необхідно було вирішити наступні завдання:

- провести аналітичне дослідження методів моніторингу сайтів;
- дослідити можливості використання лічильників на сайтах;
- розробити модель лічильнику;
- розробити програмне забезпечення моніторингу відвідування web-ресурсів.

Об'єктом дослідження є процес моніторингу відвідування web-ресурсів.

Предметом дослідження являють собою засоби створення програмного забезпечення моніторингу відвідування web-ресурсів.

Методами дослідження є загальнонаукові підходи до наукового пізнання (системний, структурний, модельний, порівняльний). Застосовано методи інформаційного та документознавчого аналізу, що дало можливість проаналізувати специфіку та розвиток вебсайтів як інформаційних систем та окреслити комплекс актуальних питань формування та використання ресурсів веб-сайту.

РОЗДІЛ 1. ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ

1.1 Аналіз предметної області

Лічильник відвідуваності — інструмент моніторингу кількості відвідувачів сайту, він допомагає вести статистику відвідувань, аналізувати, наскільки сайт популярний і відомий. Як правило, лічильники відвідувань ділять на приховані, невидимі звичайним користувачам, і відкриті, які кожен відвідувач може спостерігати на сторінках сайту. І ті, й інші користуються популярністю приблизно в рівній мірі.

Щоб підрахувати кількість відвідувачів сайту, сервіси статистики відвідувань використовують, як правило, один загальний принцип: при кожному завантаженні сторінок сайту, на яких встановлено лічильник, викликається спеціальна програма, яка передає серверу важливі дані про відвідування. У перелік цих даних може включатися IP-адреса відвідувача, браузер відвідувача, адреса сторінки, з якої прийшов відвідувач, адреса сторінки, на яку прийшов відвідувач, а також різні параметри виклику сторінки.

На стороні сервера інша програма обробляє отриману від лічильника інформацію і записує її в базу даних. Потім оновлена інформація при необхідності повертається клієнту: відвідувач в цьому випадку бачить зміни в лічильнику відвідувань. При цьому зазвичай відвідувачам демонструються на сайті дані лічильника за день, тиждень, місяць або за весь час роботи лічильника з моменту його встановлення. Вважається, що це допомагає розкрутці сайту, як в пошукових системах, так і в соціальних мережах.

Цікаво, що зазвичай лічильники відвідуваності враховують не тільки відвідувачів, але ще і так звані хости, хіти і сесії. У кожного лічильника зазвичай є власне визначення того, що слід розуміти під цими термінами, але, як правило, під хостами розуміти прийнято відвідувачів з унікальною IP-адресою (відвідувачі з декількох комп'ютерів з одною IP-адресою зараховуються як один хост), під хітами — кількість переглядів сторінок (зараховується кожне оновлення сторінки, що

містить лічильник), а під сесіями — кількість переглядів сторінок сайту протягом певного періоду (зазвичай 15 хвилин).

Слід зазначити, що кожен сервіс, який веде статистику відвідувань сайту, робить це по-своєму, тому дані про відвідуваність одного і того ж сайту при використанні різних лічильників відвідувань можуть бути різними. Для отримання більш об'єктивних даних статистики зазвичай на сайт встановлюють кілька лічильників відвідуваності, а потім підраховують середнє арифметичне їх показників.

Ефективність співробітника, який працює за комп'ютером, багато в чому залежить від його концентрації. Зараз кожен ПК підключений до інтернету і це є великою спокусою для працівників офісної сфери, щоб в робочий час почати вирішувати особисті питання. Те ж саме відвідування соціальних мереж з робочих комп'ютерів або стороннє листування сильно відволікають від основної роботи, якою він повинен займатися. У цій сфері роботодавець часто задається питанням: «А як же проконтролювати, чим займається працівник за ПК?». Тепер для цього є спеціальні програми, які виконують моніторинг відвідування сайтів співробітниками в робочий час.

Навіть у самих старанних співробітників виникає бажання протягом робочого дня перевірити особисту пошту, поспілкуватися з друзями в месенджерах, заглянути в улюблені соціальні мережі або розслабитися, проходячи черговий рівень гри. Використання програмного забезпечення спростить контроль дій співробітників в інтернеті.

За допомогою програми можна в режимі реального часу відслідковувати:

- Які сайти відвідують працівники протягом дня.
- Скільки часу проводить персонал на тому чи іншому проекті.
- Відвідування сайтів дійсно необхідно при роботі або використовується з особистої примхи співробітників.

Програма фіксує час відкриття і закриття кожної сторінки в браузері, час знаходження на сайті, дії користувача - перегляди, коментарі, лайки, репости. Всі відвідані URL-адреси, заголовки і активність користувача стають доступні адміністрації компанії. Якщо в браузерах історію відвідувань легко видалити, то в програмі це зробити неможливо.

Програма дозволяє виконувати гнучкі налаштування:

- Дозволяти чи блокувати відвідування певних сайтів на окремих робочих місцях.
- Встановлювати найточніший годинник для роботи співробітників в інтернеті.
- Вирішувати / блокувати доступ групам, відділам, підрозділам.
- Контролювати звернення користувачів до пошуку, вести облік пошукових запитів.
- Реєструвати факти відвідування сайтів, соціальних мереж.
- Здійснювати контроль за використанням електронної пошти, месенджерів для спілкування.
- Підсумувати час, проведений в інтернеті на певному ресурсі протягом дня.
- Виконувати контентний аналіз документів, що публікуються на сайтах.
- Створювати тіньові копії файлів, що завантажуються в мережу.
- Робити скріншоти екрану із заданою частотою для зазначених ресурсів.

У програмі можна розділити сайти за типами - поштові сервіси, довідники, торговельні сервіси, розважальні портали. Для кожного типу легко заборонити або обмежити доступ.

Програма допоможе відстежувати особисту переписку через корпоративну пошту компанії, спілкування в месенджерах, несанкціоновані дзвінки по SIP-телефонії. За допомогою знімків з веб-камери, скріншотів робочих столів, записів з мікрофона можна визначити, чи з користю проводить співробітник час в інтернеті

або віддається розвагам. Програма дозволить знизити витрати на послуги інтернет-провайдера і витрати на оплату корпоративної пошти [2].

1.2 Огляд етапів проєктування програмного забезпечення

Процес розробки програмного забезпечення - сукупність ряду послідовних дій, спрямованих на розробку програмного забезпечення (ПЗ).

Існує кілька моделей такого процесу, кожна з яких описує свій підхід, у вигляді завдань і/або діяльностей, які мають місце в ході процесу. Вибір тієї або іншої моделі здійснюється відповідно до обраної методології розробки програмного забезпечення.

Процес розробки складається з безлічі підпроцесів, або дисциплін, деякі з яких показані нижче. У моделі водоспаду вони йдуть одна за одною, в інших аналогічних процесах їх порядок або склад змінюється.

Процес розробки:

- Аналіз вимог - Специфікація програмного забезпечення.
- Проєктування програмного забезпечення.
- Програмування.
- Тестування програмного забезпечення.
- Системна інтеграція.
- Впровадження програмного забезпечення (або Установка програмного забезпечення).
- Супровід програмного забезпечення.

Водоспадна модель життєвого циклу передбачає послідовне виконання всіх етапів проєкту в строго фіксованому порядку. Перехід на наступний етап означає повне завершення робіт на попередньому етапі. Вимоги, визначені на стадії формування вимог, суворо документуються у вигляді технічного завдання і фіксуються на весь час розробки проєкту. Кожна стадія завершується випуском

повного комплексу документації, достатньої для того, щоб розробка могла бути продовжена іншою командою розробників.

Етапи проекту у відповідності з каскадною моделлю:

- Формування вимог;
- Проектування;
- Реалізація;
- Тестування;
- Впровадження;
- Експлуатація та супровід.

Модель ІІД передбачає розбиття життєвого циклу проекту на послідовність ітерацій, кожна з яких нагадує «міні-проект», включаючи всі процеси розробки в застосуванні до створення менших фрагментів функціональності, порівняно з проектом в цілому. Мета кожної ітерації — отримання версії програмної системи, що працює та включає функціональність, визначену інтегрованим змістом усіх попередніх і поточної ітерації. Результат фінальної ітерації містить всю необхідну функціональність продукту. Таким чином, із завершенням кожної ітерації продукт отримує приріст — інкремент — до його можливостей, які, розвиваються еволюційно. Ітеративність, інкрементальність і еволюційність в даному випадку є вислів одного і того ж сенсу різними словами зі злегка різних точок зору.

Відмінною особливістю спіральної моделі є спеціальна увага, що приділяється ризикам, що впливає на організацію життєвого циклу, і контрольним точкам. Боем формулює 10 найбільш поширених (за пріоритетами) ризиків:

- Дефіцит фахівців.
- Нереалістичні терміни і бюджет.
- Реалізація не відповідає функціональності.
- Розробка неправильного користувальницького інтерфейсу.
- Перфекціонізм, непотрібна оптимізація та покращення деталей.
- Безперервний потік змін.

- Брак інформації про зовнішні компоненти, що визначають оточення системи або залучених в інтеграцію.
- Недоліки в роботах, що виконуються зовнішніми (стосовно проєкту) ресурсами.
- Недостатня продуктивність одержуваної системи.
- Розрив у кваліфікації фахівців різних областей.

У сьогоденній спіральній моделі визначено наступний загальний набір контрольних точок:

- Concept of Operations (COO) — концепція (використання) системи;
- Life Cycle Objectives (LCO) — цілі і зміст життєвого циклу;
- Life Cycle Architecture (LCA) — архітектура життєвого циклу; тут можливо говорити про готовність концептуальної архітектури цільової програмної системи;
- Initial Operational Capability (IOC) — перша версія створюваного продукту, придатна для дослідної експлуатації;
- Final Operational Capability (FOC) — готовий продукт, розгорнутий (встановлений і налаштований) для реальної експлуатації.[3]

1.3 Аналіз варіантів розв'язання досліджуваної задачі.

Питання, як дізнатися і перевірити наскільки часто відвідують ваш сайт, виникає одразу після запуску сайту. І це правильно – чим більше відвідують, тим краще.

Моніторити, тобто стежити за роботою сайту, можна з різною метою: чи для визначення популярності за кількістю відвідувачів, за певний період часу, чи для контролю роботи хостингової компанії (моніторити доступність ресурсу), чи для повідомлення про зміну певного інформаційного наповнення сайту.

Але найперше, що цікавить власників сайтів, це рівень відвідуваності їх ресурсів. Скільки разів, коли заходили на їх сайт, з яких сторінок здійснювався

перехід, за якими ключовими словами їх знаходили, яка позиція у пошуку – за цим варто слідкувати для забезпечення розвитку інтернет ресурсу[4].

Розпочнемо з відвідуваності. Отож, як дізнатися, перевірити і переглянути статистику відвідуваності сайту.

Для цього можна скористатися наступними способами:

1. Найточніший спосіб перевірки статистики відвідуваності – використання так званої серверної статистики відвідуваності. Більшістю віртуальних хостингів надається дана послуга. Якщо ж ваш проєкт розміщений на реальному сервері, то ви самостійно зможете встановити і активізувати відповідні скрипти, які надають можливість дізнатися, перевірити і провести детальний аналіз відвідуваності вашого сайту.

Найпопулярнішими програмами для роботи з серверною статистикою є:

- AWStats – аналізатор логів сервера.
- CNStats STD – сучасна система для збору і аналізу статистики відвідуваності сайту.

2. Дані системи надають детальні відомості про те хто, коли і звідки відвідував ваш ресурс, за яким ключовими словами вас було знайдено, з яких сторінок відвідувачі потрапляли на ваш сайт. Незважаючи на переваги даних систем – результати їх роботи будуть цікаві лише вам. А з часом дізнатися і перевірити статистику вашого сайту захочуть і потенційні рекламодавці, наприклад, для розміщення банерної реклами. Тоді необхідно буде вказати на статистику відвідуваності на вашому сайті.

3. В багатьох CMS, як платних так і безкоштовних, надається можливість відслідковувати статистику відвідуваності сайтів. Якщо такої можливості не закладено в самій CMS, то однозначно знайдуться сторонні розширення перевірки і аналізу статистики відвідуваності. Проте, як правило, системи збору статистики в CMS далеко не надійні, і зазвичай потенційні

рекламодавці не дуже вірять тому, що нарахує статистика відвідуваності вашої CMS.

4. Найчастіше для аналізу статистики відвідуваності сайту використовують лічильники і рейтинги сайтів за відвідуваністю (клієнтські системи статистики відвідувань). Основна задача при використанні лічильників відвідуваності полягає в їх правильному встановленні – впровадженні коду лічильників у всі, без виключення, сторінки власного ресурсу для забезпечення точності перевірки відвідуваності сайту. Лічильник чи рейтинг відвідуваності представляє собою або html-код, який одночасно із завантаженням сайту буде відвантажувати з сервера лічильника картинку цього ж лічильника, або код лічильника містить певний скрипт, який знову ж таки відвантажується з третього сервера. В результаті при завантаженні у будь-якому браузері довільної сторінки сайту, відбувається звернення до третього сервера для завантаження або картинку лічильника відвідуваності, або скрипту. Третій сервер тільки рахує ці завантаження. Як приклади популярних лічильників виступають сервіси Яндекс і Google (Google-Analytics) [4].

Для успішного розвитку комерційного інтернет-проекту необхідно проводити аналіз його роботи з метою отримання відповідей на наступні питання:

- скільки відвідувачів має сайт, як швидко збільшується їх кількість, що на сайті для них є цікавим;
- з яких країн і пошукових сервісів вони потрапляють на сайт, куди уходять;
- якою операційною системою, браузером вони користуються, з якою роздільною здатністю
- чи здійснили вони дію, заради якої створювався сайт, який час вони були на сайті
- чи ефективні рекламні кампанії або партнерські програми.

Основним завданням веб-аналітики є моніторинг роботи веб-сайтів на підставі якого визначається веб-аудиторія і вивчається поведінка веб-відвідувачів

для прийняття рішень з розвитку та розширення функціональних можливостей веб-ресурсу [1].

Існує два основних метода підрахунку статистики: використання аналізатора логів і застосування лічильника відвідувань(рис.1.1).



Рис. 1. 1 Методи отримання статистики

Веб лог-аналізатор — це внутрішня локальна програма, що встановлюється на комп'ютер користувача, як правило, сервер.

Кожен з методів має свої недоліки і переваги(табл.1.1). В більшості випадків використовуються лічильники відвідування. Але, ні один з методів не забезпечує повної достовірності статистичних даних, і на практиці показання лічильників, і лог-аналізаторів можуть розрізнятися в десятки разів. Оптимальним рішенням є комбінація обох методів збору інформації, оскільки тільки в цьому випадку можливо отримати найбільш близькі до реальності дані.

Таблиця 1.1

Порівняна характеристика лічильників засобів статистики

Інструмент аналізу	Переваги	Недоліки
Лічильники відвідування	надають послугу безкоштовно процедура встановлення лічильника дуже проста результати обробки надаються швидко і в наглядному	інформація зберігається на сервері статистики вірогідність даних невелика тому, що дуже часто вони губляться погрішність даних, що

	<p>вигляді дозволяють порівняти відвідування сайту з сайтами-конкурентами, а також установити географію відвідувачів дозволяють точно ідентифікувати користувача, параметри браузера, екрану комп'ютеру тощо</p>	<p>збираються лічильниками значна (може складати до 30%) лічильник не фіксує відвідування сайту пошуковими машинами лічильник зменшує швидкість завантаження сторінки лічильник не реєструє завантаження мультимедійних файлів, флеш-об'єктів, зображень тощо</p>
Аналізатори логів	<p>Надають саму повну статистику про відвідування Web-сайту. Сторінки сайту відкриваються швидше, тому що браузер не завантажує логотипи лічильників з сервера статистики.</p>	<p>Не дозволяють порівняти відвідування сайту з сайтами-конкурентами – у них не має для цього даних. Не можливо установити географію відвідувачів. Аналізатори логів не використовують файли cookie.</p>

Аналізатори логів. Лог-файл (журнал) – файл на сервері, призначений для фіксації даних про користувача: IP-адреси, URL-адреси сторінки, що запитуються, а також дані про різні елементи сторінки (текст, графіка, мул.файли), що також запитуються для її створення.

Аналізаторами логів називають програми, які вмюють аналізувати серверні логи. Оскільки web-сервер фіксує всі звернення до сайту, то звіти, що формуються

на їх основі забезпечують саму повну статистику про його відвідувачів. За допомогою звітів, наприклад, можна дізнатися про загальну кількість хітів і хостів, а також з'ясувати, з яких посилань приходили на сайт відвідувачі, на яких саме сторінках вони були, які файли завантажили, скільки часу провели, з яких сторінок пішли тощо.

Іншими словами, наведена у звітах лог-аналізаторів статистика відрізняється максимально можливою повнотою.

Найбільш відомими з існуючих некомерційних аналізаторів логів є програми Analog, Webalizer і AWStats. З комерційних продуктів в якості прикладів лог-аналізаторів можна привести такі, як Deep Log Analyzer, WebLog Expert і AlterWind Log Analyzer. Крім спеціальних пакетів для аналізу логів працювати з серверними логами іноді можуть і інші - наприклад програми, що призначені для аудиту та оптимізації сайтів, зокрема Page Promoter та Semonitor.

Лічильники відвідування. Інформація на лічильнику відвідування формується, як правило зовнішнім сервером статистики під час її відкриття. Повторне відвідування сторінки фіксується за допомогою файлів cookie.

На відміну від аналізаторів, лічильники відвідувань збирають дані для аналізу самостійно, правда для цього потрібно розмістити на сторінках сайту спеціальний код відслідкування, за яким при зверненні до сторінки записуються дані про відвідувача. Лічильники ведуть загальну статистику відвідуваності з детальним розподілом за часом, фіксують хости і хіти, виявляють унікальних відвідувачів, разом з кількістю відвіданих ними сторінок і часом перебування на кожній з них. Крім того, лічильники можуть запам'ятовувати дані про шляхи переміщення відвідувача по сайту, починаючи з точки заходу на сайт і закінчуючи точкою виходу.

Висновки

Отже, зараз дуже багато видів діяльності і робіт пов'язані з необхідністю контактування персоналу з комп'ютерами та інтернетом. Далеко не всі з співробітників сумлінно виконують свої робочі обов'язки і займаються саме роботою на наданому обладнанні. У зв'язку з чим суттєво знижується продуктивність праці і ефективність набраного штату в цілому. Звичайними шляхами відстежити, що шукав, читав і дивився співробітник в інтернеті, проблематично, так як багато з них вміють чистити історію браузера, як цілком, так і вибірково. Відповідно, для роботодавця тут ніякої інформації не залишається. Тому, програма з моніторингу здійснює зчитування відвіданих веб-ресурсів відразу, в той момент, як тільки працівник на нього зайшов. Більш того, вона не тільки зчитує, а й зберігає всі адреси з браузера, а також копіює для «Адміністратора» всі завантажені з інтернету файли. Задача аналізу поведінки людини в інтернеті не нова, в сучасному світі достатньо програм, технологій а також сервісів.

РОЗДІЛ 2. Організація розробки програмного забезпечення моніторингу відвідування web-ресурсів

2.1 Загальна концепція

Процес розробки складається з безлічі підпроцесів, або дисциплін, деякі з яких показані нижче. Етапи проєкту у відповідності з каскадною моделлю:

- Формування вимог;
- Проєктування;
- Реалізація;
- Тестування;
- Впровадження;
- Експлуатація та супровід.

Специфікація вимог до програмного забезпечення - специфікація вимог для програмної системи - це повний опис поведінки системи що розробляється. Вона включає множину прецедентів які описують всі взаємодії, які користувачі мають з програмним забезпеченням. Прецеденти також відомі як функціональні вимоги. На додачу до прецедентів SRS також включає нефункціональні (чи додаткові) вимоги. Нефункціональні вимоги є вимогами які накладають обмеження на проєкт, чи реалізацію (такі як вимоги інженерії продуктивності, стандарти якості, чи обмеження проєктування) [6].

Аналіз вимог полягає в визначенні потреб та умов, які висуваються щодо нового, чи зміненого продукту, враховуючи можливо конфліктні вимоги різних замовників, таких як користувачі чи бенефіціари. Аналіз вимог є критичним для успішної розробки проєкту. Вимоги мають бути задокументованими, вимірними, тестовними, пов'язаними з бізнес-потребами, і описаними з рівнем деталізації достатнім для конструювання системи. Вимоги можуть бути архітектурними, структурними, поведінковими, функціональними, та не функціональними.

Проєктування програмного забезпечення - це процес вирішення задач та планування для створення програмного рішення. Після того як мета і специфікація

програми описані, розробник створить дизайн проєкту, або найме дизайнера для розробки плану рішення. В дизайн включаються як описи низькорівневих компонентів, алгоритмів, так і огляд архітектури.

Проєктуванню зазвичай підлягають:

1. Архітектура програмного забезпечення
2. Компоненти ПЗ
3. Користувацькі інтерфейси

В процесі проєктування ПЗ застосовують різні моделі - блок-схеми, ER-діаграми, DFD тощо.

Програмування - процес проєктування, написання, тестування, зневадження і підтримки комп'ютерних програм. Програмування поєднує в собі елементи інженерії, фундаментальних наук (перш за все математики) і мистецтва.

У вужчому значенні програмування розглядається як кодування - реалізація у вигляді програми одного чи кількох взаємопов'язаних алгоритмів (у сучасних умовах це здійснюється з застосуванням мов програмування). У ширшому сенсі процес програмування охоплює і створення, тобто розробку, алгоритмів, і аналіз потреб майбутніх користувачів програмного забезпечення.

Тестування програмного забезпечення - це процес технічного дослідження, призначений для виявлення інформації про якість продукту відносно контексту, в якому він має використовуватись. Техніка тестування також включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою оцінки.

Системна інтеграція в інженерії - поєднання компонентів підсистем в єдину систему та забезпечення їх роботи у якості єдиної системи. В області інформаційних технологій системна інтеграція є процесом об'єднання різних обчислювальних систем і програмних застосунків фізично або функціонально. Системна інтеграція полягає у розробці комплексних рішень, призначених для

досягнення максимальної ефективності функціонування системи шляхом налагодження ефективної взаємодії її підсистем.

Інсталяція (встановлення) — процес встановлення програмного забезпечення на комп'ютер кінцевого користувача. Виконується особливою програмою (пакетним менеджером), присутнім в операційній системі (наприклад, RPM і APT в Linux, Windows Installer в Microsoft Windows), або ж тим, що вже входить до складу самого програмного забезпечення засобом встановлення. В операційній системі GNU дуже поширене використання системи GNU toolchain і її аналогів для компіляції програмного забезпечення безпосередньо перед встановленням.

Супроводження програмного забезпечення — процес покращення, оптимізації та виправлення дефектів у програмному забезпеченні після його вводу до експлуатації [7].

Цей процес стандартизовано організацією ISO — ISO/IEC 14764.

У зв'язку з вирішенням так званої проблеми 2000 року (пов'язаної з кодуванням дат у новому тисячолітті, зокрема, у двохсимвольному форматі) супроводження почало розглядатися, як важливіший процес, що здійснюють розробники. Після змін система має вирішувати ті самі задачі, а також мати план перенесення інформації в інші БД. Супровід відповідно до стандартів ISO/IEC 12207 і ISO/IEC 14764 проводиться з метою виконання і модифікації програмного продукту в процесі експлуатації за умови збереження його цілісності [8].

2.2 Модель програмного забезпечення моніторингу web-ресурсів

Для досягнення поставленої мети необхідно забезпечити виконання наступних завдань:

1. аналіз існуючих в даний час систем моніторингу сайтів;
2. відбір аналогів з існуючих систем даного типу за різними критеріями;
3. розробка пакету моделей автоматизованої системи, що дозволяє здійснювати моніторинг сайтів з функціональними можливостями, виявленими в прототипі;

4. розробка пакету моделей на створення автоматизованої системи моніторингу сайтів;
5. розробка автоматизованої інтелектуальної системи моніторингу сайтів;
6. впровадження системи в дослідну експлуатацію.

В даний час на ринку не представлені системи моніторингу сайтів, а також системи, призначені для просування сайтів, що дозволяють здійснювати автоматичну обробку статистичних даних з метою видачі рекомендацій і конкретних методів щодо виключення проблемних місць сайтів, безпосередньо впливають ні ефективність їх функціонування.

Розглядається безліч аналогів - систем моніторингу сайтів було розбите на типи, відповідні функціоналу, що дозволяє здійснювати моніторинг певних видів характеристик сайтів:

1. працездатність;
2. відвідуваність;
3. SEO-оптимізація.

Беремо у приклад Google Analytics. У Google Analytics відслідковуються наступні типи даних про взаємодію відвідувачів з контентом сайту:

1. сторінки, через які відвідувачі входили на сайт і виходили з нього;
2. як часто і скільки часу відвідувачі переглядали ті чи інші сторінки;
3. наскільки активно відвідувачі шукали певний контент на сайті;
4. ступінь взаємодії з певним типом контенту, наприклад слайд-шоу або вбудованим відео;
5. як часто відвідувачі натискали на оголошення AdSense, а також дохід від цих кліків.

Виходячи з цих даних можна визначити, наскільки сайт сприяє досягненню поставлених завдань.

Програмне забезпечення являє собою сервіс, який здійснює підрахунок кількості відвідувань сайту. Для того щоб організувати роботу такого сервісу на своєму ресурсі, необхідно розмістити код конкретного лічильника на тих

сторінках, інформація про відвідування яких необхідна. Як правило, виконавши це, можна дізнатися такі дані:

- Кількість унікальних відвідувачів сайту. Коли користувач вперше за добу заходить на сторінку ресурсу - він додає до показника лічильника одиницю. Повторне відвідування протягом доби вже не буде вважатися унікальним;
- Візити. Встановити лічильник відвідувань на сайт можна і для того, щоб визначити: скільки разів протягом доби конкретний користувач заходив на інформаційний ресурс;
- Загальна кількість відвідувань сайту. По тому, наскільки даний показник більше числа унікальних відвідувачів, можна визначити: чи хочуть люди повертатися на сторінки ресурсу (рис. 2.1) [9].

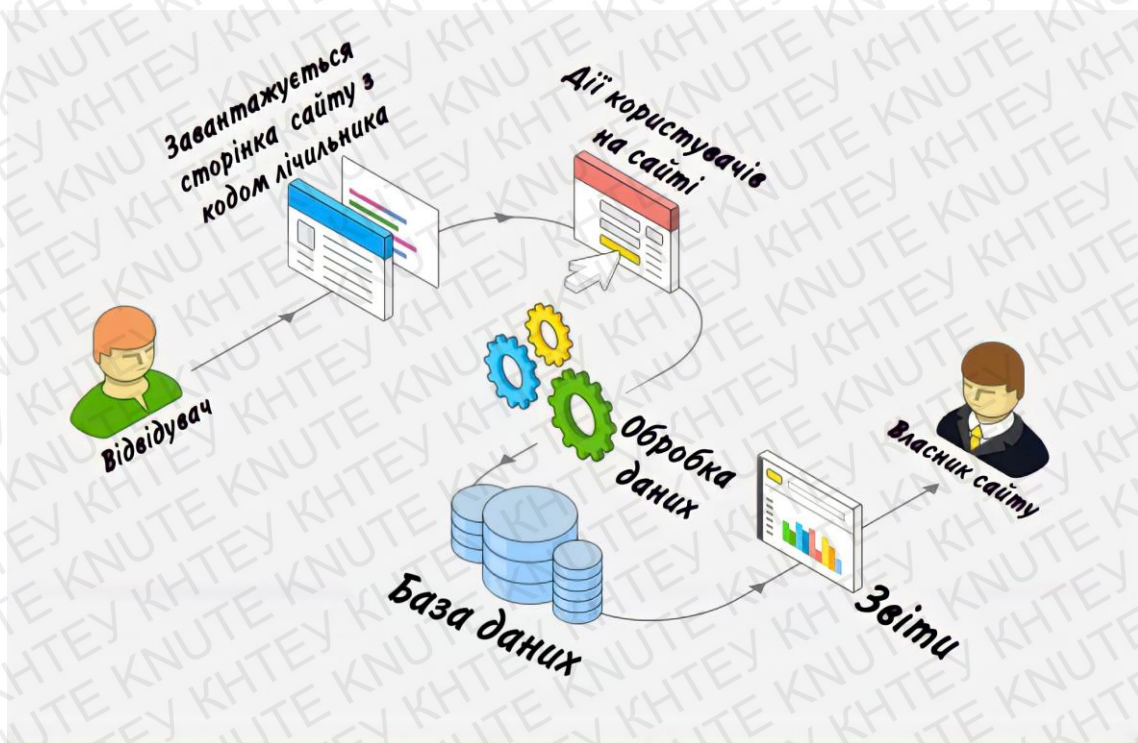


Рис. 2. 1 Модель програмного забезпечення

Найпростіший лічильник виглядає приблизно так: при запиті до сервера, тобто, ще перед завантаженням сайту в браузер, скрипт звертається до деякого файлу-лічильнику, в якому зберігається число - кількість відвідувань, - отримує це число і збільшує на одиницю. Потім стирає файл і записує в нього нове збільшене

значення. Таким чином при кожному відвідуванні сайту значення лічильника збільшується [11].

Такий стан речей нас, звичайно, не влаштовує, тому що при відвідуванні сайту однією людиною лічильник швидше за все збільшиться не на один, а на всі п'ятдесят (за умови, що ця людина вирішить пройтися по сторінках вашого ресурсу). Але є способи обійти цю "проблему". Один з них такий: людина заходить на сайт, лічильник збільшується, відкривається сесія. При його переході на іншу сторінку, скрипт бачить його відкриту сесію і продовжує її, не збільшуючи значення лічильника. А сесія закриється тільки при його остаточному відході з сайту. Тобто алгоритм приблизно такий(рис.2.2):

```
якщо (немає сесії) {створити сесію; збільшити значення лічильника;}  
якщо (сесія) продовжити сесію;
```



Рис. 2. 2 Алгоритм розробки лічильника

Спочатку створюється база даних(рис.2.3) (далі - БД), що складається з однієї записи з одним полем. Його значення - 0.



Рис. 2. 3 Алгоритм розробки БД

Далі по вищеописаному алгоритму в БД вносяться нові записи, які містять поточну дату в форматі рік.номерМісяця.деньМісяця. Так можна підрахувати скільки завантажень було зроблено сьогодні і за поточний місяць. Тобто скрипт проходить по всіх записах і порівнює їх значення - дати. Але як тільки він натикається на черговий запис, місяць якої не збігається з поточним, він видаляє всі записи починаючи з цієї та до другої включно, підраховуючи їх кількість. До першого запису він додає цю кількість. Так виходить, що неважко визначити і кількість відвідувань сайту в цілому, з початку його існування, а точніше з початку існування лічильника [10].

У нашому випадку лічильник має два елементи: унікальні відвідування та перегляди.

Унікальний відвідувач сайту – це людина, що завітала на онлайн-майданчик у певний проміжок часу. Якщо в цей же період вона загляне на веб-ресурс ще раз, то вона вже не буде таким.

У цього терміна є ряд нюансів. Унікальним вважається той, хто зайшов з конкретного пристрою, що має свій IP. Але ви можете відвідати сторінку і з іншого

комп'ютера. Тоді ваш візит знову буде унікальним. Крім того, відвідувачами можуть бути не тільки люди.

Так що таке унікальний відвідувач сайту. Ним може бути спам-бот, пошуковий робот, програма, яка імітує реального користувача. Ці заходи на сайт будуть також визнані унікальними(рис.2.4). Під імітуванням розуміються ті дані, які зчитує лічильник, встановлений на сайті: джерело трафіку, IP-адресу, браузер і інші.

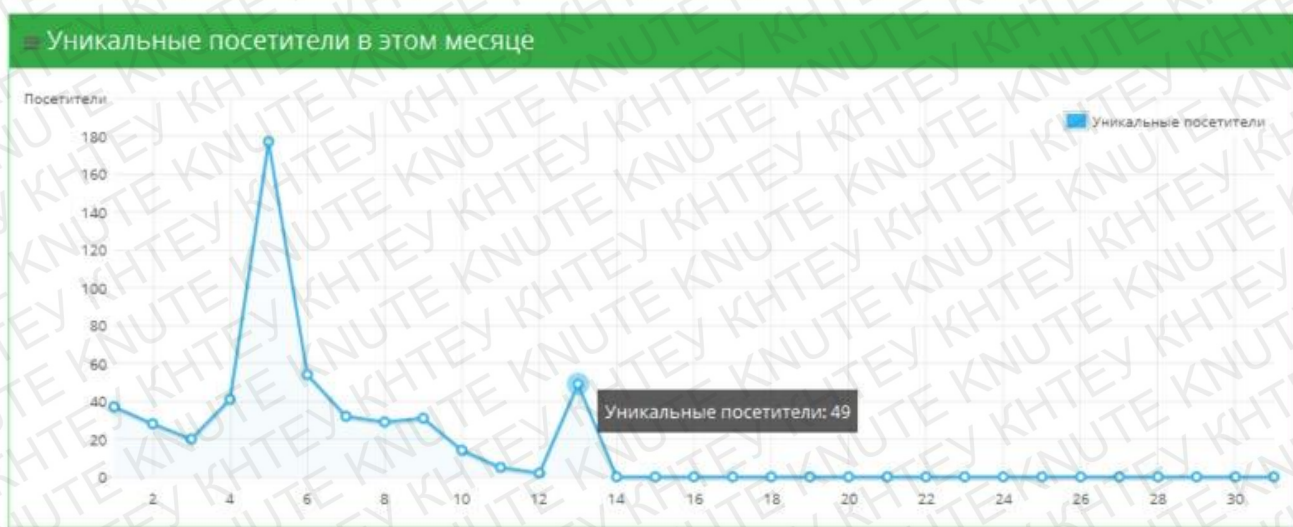


Рис. 2. 4 Унікальні відвідувачі

Доступ до такого лічильника дозволяє переглядати кількість візитів і переглядів(табл.2.1). Тільки кількість людей, які відкривають доступ до такої статистики, дуже невелика. Комерційний сайт з відкритою Метрикою – взагалі рідкість(рис.2.5).



Рис. 2. 5. Статистика відвідуваності

Таблиця 2.1

Перелік задач, що можна вирішити за допомогою засобів Інтернет-статистики

Оцінка загальної аудиторії сайту	Оцінка якості аудиторії
Загальна статистика: кількість відвідувачів та переглянутих ними сторінок, кількість нових відвідувачів, розподіл відвідувачів за регіонами.	Цільова аудиторія: кількість відвідувачів, що стали цільовими (виконали дію, заради якої створювався сайт)
Джерела відвідувачів: від куди вони потрапили на сайт, які пошукові фрази ними були використані.	Від куди потрапили на сайт цільові відвідувачі: можна визначити найбільш ефективну рекламу сайту.
Докладна оцінка відвідуваності сайту: найпопулярніші розділи сайту, глибина перегляду сайту, шляхи переміщення по сайту.	Технічні параметри: браузер, роздільна здатність монітору, мова, тип операційної системи, швидкість передачі даних тощо.

2.3 Алгоритм розробки програмного забезпечення

Для реалізації лічильника переглядів сторінок сайту нам знадобиться (рис.2.6):

1. Php файли зі сторінками сайту
2. Файл .htaccess для завдання кодування сервера в значенні UTF-8
3. Папка css з файлом стилів
4. Папка inc з підключеними файлами. Їх три:
 - Файл підключення до бази даних
 - Файл, який буде підключатися в потрібному місці сторінки для виведення лічильника на екран
 - Основний скрипт, в якому прописана логіка роботи лічильника

5. Папка admin з файлом index.php. У цьому файлі ми зможемо переглянути статистику відвідувань сайту.

admin	14.05.2021 15:46	Папка с файлами	
css	04.06.2020 18:43	Папка с файлами	
images	04.06.2020 19:49	Папка с файлами	
inc	14.05.2021 13:47	Папка с файлами	
js	04.06.2020 12:12	Папка с файлами	
index.php	14.05.2021 17:29	Файл "PHP"	11 КБ
картини.php	14.05.2021 17:30	Файл "PHP"	8 КБ
контакти.php	14.05.2021 17:31	Файл "PHP"	8 КБ
форум.php	14.05.2021 17:31	Файл "PHP"	9 КБ

Рис. 2. 6 Файли необхідні для розробки ПЗ

Для створення серверу ми використовуємо програму Open Server.

Для реалізації лічильника переглядів постів на форумі:

1. База даних MySQL
2. Файл database.php
3. Скрипт у якому прописана логіка роботи.

Короткий опис роботи. Спочатку скачуємо та встановлюємо програму Open Server. Налаштовуємо функцію PhpMyAdmin для бази даних. Додаємо сайт у базу програми та налаштуємо айпі. Далі переходимо в налаштування PhpMyAdmin(натискаємо правою кнопкою миші на програми та обираємо «Додатково» у меню обираємо PhpMyAdmin. У браузері відкривається база даних для цього айпі. Створюємо акаунт та переходимо до заповнення таблиці бази даних. Після цього прописуємо код підключення до бази, код логіки програми та код «Адміна»(для перегляду звітів). Далі підключаємо лічильник до сайту(на кожну сторінку).

Висновки

Отже, програмне забезпечення являє собою сервіс, який здійснює підрахунок кількості відвідувань сайту. Для того щоб організувати роботу такого сервісу на своєму ресурсі, необхідно розмістити код конкретного лічильника на тих сторінках, інформація про відвідування яких необхідна. У цьому розділі була описана загальна концепція програмного забезпечення. В ній розписано, що саме знадобиться у роботі та які додаткові програми було встановлено. Основна робота буде зроблена в програмі SublimeText, яка додатково налаштовується для роботи з файлами PHP. Також встановлена програма Open Server для виграження сайту на сервер, щоб в подальшому використовувати базу даних MySQL. Детальний опис створення ПЗ буде прописаний у розділі 3.

РОЗДІЛ 3. Програмна реалізація

3.1 Програмна реалізація бази даних

У першу чергу скачуємо та встановлюємо програму Open Server. Open Server - це програмна платформа для веб-фахівців, що забезпечує створення і функціонування в середовищі Windows повноцінного сервера [12].

Це збірка всіх необхідних компонентів, за допомогою якої можна створювати на своєму комп'ютері сайти будь-якої складності, за допомогою будь-якої обраної CMS. Створені таким чином сайти в подальшому (у міру готовності) переносяться на оплачувані хостинги.

На вибір нам пропонується 3 збірки дистрибутива: базова, преміум і максимальна. Якщо мова йде тільки про створення і тестування сайту, можна обійтися базовою, в ній присутні потрібні компоненти - Apache, PHP і MySQL.

Далі нам буде запропоновано встановити середу Microsoft Visual C ++. В офіційних дистрибутивах актуальних версій Windows вона повинна бути встановлена.

Після установки Microsoft Visual C ++ комп'ютер перезавантажиться. Запускаємо OpenServer за новою. Доступ до функціоналу платформи стане доступний в меню, що з'являється на її значку в системному треї. Запуск сервера здійснюється опцією з зеленим прапорцем «Запустити».

Отже, сервер запущений. Його працездатність, до речі, можна перевірити, ввівши в адресний рядок браузера: localhost.

Якщо все нормально, побачимо тестову вітальну напис «Open Server Panel працює;-)».

Далі в папці установки OpenServer «OSPanel» відкриваємо підпапку «domains». Додаємо в неї папку з сайтом.

Наступний етап - створення бази даних MySQL сайту. В меню OpenServer натискаємо «Додатково», вибираємо «PhpMyAdmin». У вікні браузера за замовчуванням відкриється інтерфейс веб-додатки. У формі авторизації необхідно

ввести лише ім'я користувача: root. Поле пароля залишаємо порожнім. Тиснемо «Вперед».

Переходимо до створення бази, де у нас буде лежати вся інформація. Назвемо її stats і створимо в ній дві таблиці (рис.3.1).

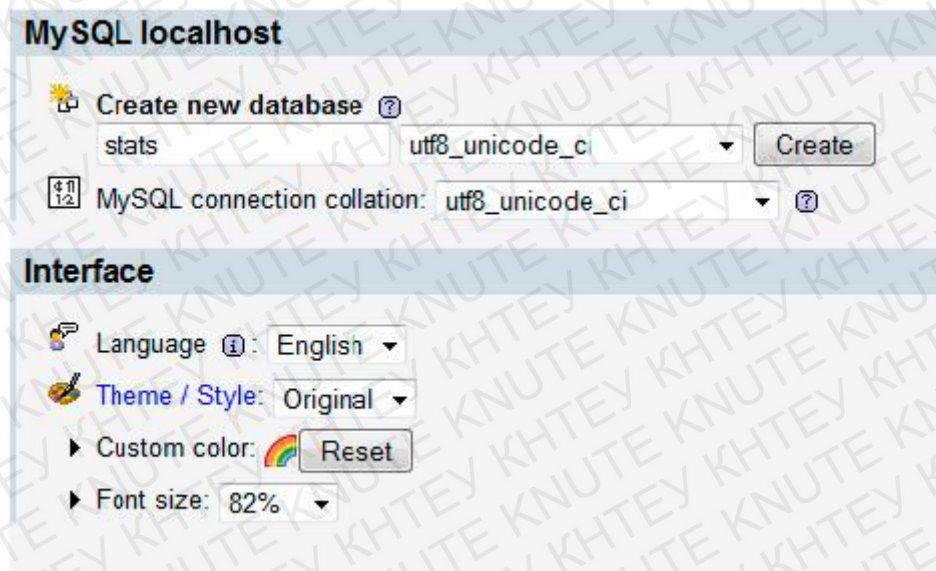


Рис.3.1 База даних stats

Отже, створюємо першу таблицю. Назвемо її ips - в ній будуть зберігатися ідентифікатори IP-адреси і самі IP-адреси відвідувачів сайту.

У таблиці будуть 2 поля(рис.3.2):

- id_id (int (12), первинний ключ, автоінкремент) - ідентифікатор IP-адреси;
- ip_address (varchar (50)) - сама IP-адреса відвідувача.

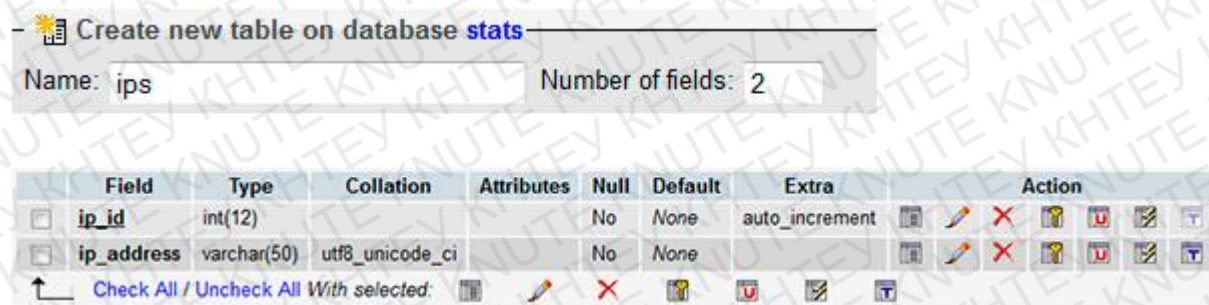


Рис.3.2 Таблиця ips

Далі створюємо другу таблицю. Назвемо її visits - в ній буде зберігатися інформація про відвідування сайту.

У таблиці будуть 4 поля (рис.3.3):

- visit_id (int (12), первинний ключ, автоінкремент) - ідентифікатор візиту;
- date (date) - дата візиту;
- hosts int (12) - кількість хостів (унікальних відвідувань);
- views int (12) - кількість хітів (переглядів сторінок).

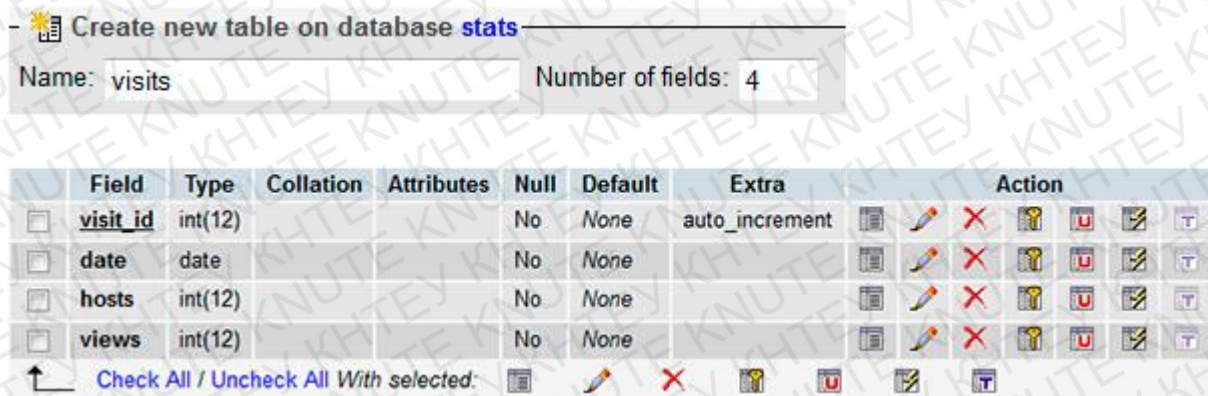


Рис.3.3 Таблиця visits

У підсумку база буде виглядати ось так (рис. 3.4):

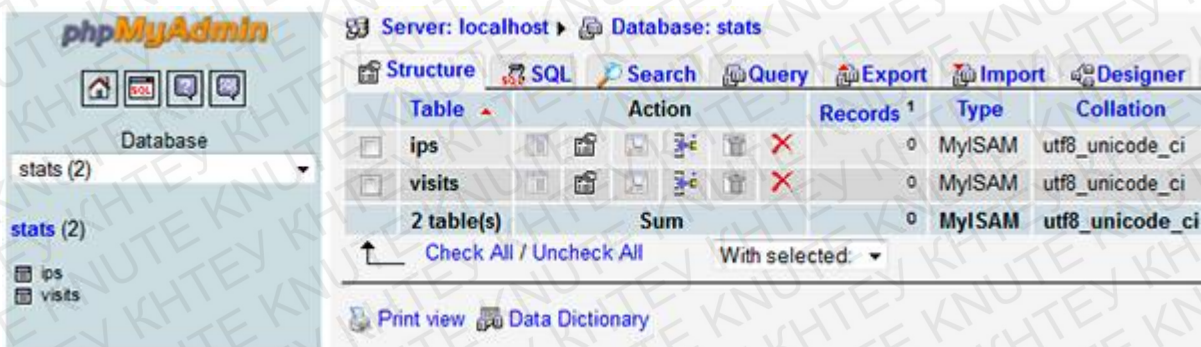


Рис.3.4 Готова база даних

3.2 Програмна реалізація проекту

Тепер, коли база даних створена, можна перейти до кількох положень, від яких ми будемо відштовхуватися при написанні скрипта:

1. Визначати унікального відвідувача ми будемо по IP-адреси. 1 IP-адреса дорівнює 1 унікальному відвідувачеві (хосту).
2. Заходи протягом 1 дня з одною і тією ж IP-адресою не збільшують кількість унікальних відвідувачів в цей день.
3. Будь-який візит на сайт протягом доби збільшує лічильник відвідувань (хітів) на 1 (незалежно від того, чи відбувається він з унікальної за сьогодні IP-адреси чи ні).
4. Початок доби ми будемо визначати наступним чином: коли людина заходить на сайт, ми будемо робити перевірку, чи існують в таблиці visits будь-які записи за сьогоднішню дату. Якщо ні, значить поточний відвідувач є першим відвідувачем в даних добі.
5. Раз на добу йде очищення таблиці з IP-адресами для того, щоб ми могли заносити в неї IP-адреси вже поточного дня і збирати статистику саме за сьогоднішніми відвідинам сайту (робиться це як раз за фактом виявлення того, що до нас завітав перший в поточних добі відвідувач).

Відкриваємо код сайту. На самому початку ми підключаємо файл `count.php`, який містить всю логіку роботи лічильника.

У нижній частині, перед закриваючим тегом `body` ми підключаємо файл `show_stats.php`, в якому міститься сам код лічильника, тобто в цьому файлі вже формується безпосередньо те, що ми будемо виводити на екран.

Тобто на кожній сторінці сайту у нас буде виводитися лічильник, і буде вказано, скільки було за сьогодні унікальних відвідувачів і переглядів.

Про всяк випадок створюємо в корені сайту файл `.htaccess` з однією єдиною директивою для завдання кодування UTF-8.

У файлі `db.php` вказані параметри підключення до бази даних - хост, ім'я бази, логін і пароль.

У цьому ж файлі (рис.3.5) виконується безпосередньо з'єднання, а ідентифікатор з'єднання заноситься в змінну `$db`. У разі помилки виводиться повідомлення про неможливість підключення до бази.

```

1 <?Php
2 //·Хост·(зазвичай·localhost)
3 $Db_host·="localhost";
4 //·Ім'я·бази·даних
5 $Db_name·="stats";
6 //·Логін·для·підключення·до·бази·даних
7 $Db_user·="root";
8 //·Пароль·для·підключення·до·бази·даних
9 $Db_pass·="";
10
11 $Db·=·mysqli_connect·($db_host,·$db_user,·"",·$db_name)·or·die·("Неможливо·підключитися·до·БД");

```

Рис.3.5 Файл db.php

Пишемо основний файл з логікою count.php (рис.3.6). Отже, спочатку ми підключаємо файл bd.php, що забезпечує зв'язок з базою даних, після чого починаємо писати основний код.

```

<?Php·include·'db.php';
//·Вказуємо·кодування,·в·якій·буде·отримана·інформація·з·бази
@mysqli_query·($db,·'set·character_set_results·="utf8"');
//·Отримуємо·IP·адреса·відвідувача·і·зберігаємо·поточну·дату
$visitor_ip·=·$_SERVER·['REMOTE_ADDR'];
$date·=·date·("Y-m-d");
//·Дізнаємося,·чи·були·відвідування·за·сьогодні
$res·=·mysqli_query·($db,·"SELECT·visit_id·FROM·visits·WHERE·date·=·'$date·'"·)·or·die·("Проблема·при·підключенні·до·БД");
//·Якщо·сьогодні·ще·не·було·відвідувань
if·(mysqli_num_rows·($res)·==·0)
{
····//·Очищаємо·таблицю·ips
····mysqli_query·($db,·"DELETE·FROM·ips·");
····//·Заносимо·в·базу·IP·адреса·поточного·користувача
····mysqli_query·($db,·"INSERT·INTO·ips·SET·ip_address·=·'$visitor_ip'");
····//·Заносимо·в·базу·дату·відвідування·і·встановлюємо·кількість·переглядів·і·унік·відвідувань·в·значення·1
····$Res_count·=·mysqli_query·($db,·"INSERT·INTO·visits·SET·date·=·'$date',·hosts·=·1,·views·=·1");
}
//·Якщо·відвідування·сьогодні·вже·були
else
{
····//·Перевіряємо,·чи·є·вже·в·базі·IP·адреса,·з·якого·відбувається·звернення
····$Current_ip·=·mysqli_query·($db,·"SELECT·ip_id·FROM·ips·WHERE·ip_address·=·'$visitor_ip'");
····//·Якщо·такий·IP·адреса·вже·сьогодні·був·(тобто·це·не·унікальний·відвідувач)
····if·(mysqli_num_rows·($current_ip)·==·1)
····{
········//·Додаємо·для·поточної·дати·+1·перегляд·(хіт)
········mysqli_query·($db,·"UPDATE·visits·SET·views·=·views·+·1·WHERE·date·=·'$date'");
······}
····//·Якщо·сьогодні·такого·IP·адреси·ще·не·було·(тобто·це·унікальний·відвідувач)
····else
····{
········//·Заносимо·в·базу·IP·адреса·цього·відвідувача
········mysqli_query·($db,·"INSERT·INTO·ips·SET·ip_address·=·'$visitor_ip'");
········//·Додаємо·в·базу·+1·унікального·відвідувача·(хост)·і·+1·перегляд·(хіт)
········mysqli_query·($db,·"UPDATE·visits·SET·hosts·=·hosts·+·1,·views·=·views·+·1·WHERE·date·=·'$date'");
······}
}
}

```

Рис.3.6 Файл count.php

Насамперед ми отримуємо з глобального масиву \$_SERVER IP-адресу відвідувача і зберігаємо поточну дату.

Дата нам необхідна для того, щоб перевірити, чи були вже за сьогодні якісь відвідування, чи ні - це як раз наш наступний крок.

Далі у нас можливо тільки 2 варіанти: або відвідувань сьогодні ще не було, або відвідування сьогодні вже були.

У першому випадку (якщо відвідувань ще не було) і поточний відвідувач є першим в цих добі, ми виробляємо очищення таблиці з IP-адресами.

Це потрібно нам для того, щоб почати збір статистики вже саме за сьогоднішній день.

Крім того, ми вносимо до бази IP-адресу поточного користувача, а в базі даних для поточного дня проставляем в поля переглядів і унікальних відвідувачів по одиниці, тому що це перша людина, що приніс за сьогоднішній день, перший хіт і перший хост.

Другий варіант - коли відвідування сьогодні вже були.

В цьому випадку ми перевіряємо, чи є вже в базі IP-адреса, з якої відбувається звернення, тому що нам потрібно враховувати лише унікальні IP-адреси в межах однієї доби.

Якщо з'ясується, що така IP-адреса вже сьогодні була, то ми до поточного значення переглядів додаємо +1 і більше нічого.

Якщо ж такої IP-адреси сьогодні ще не було (тобто відвідувач унікальний), то зберігаємо в базу його IP і оновлюємо значення як кількості унікальних відвідувачів, так і переглядів (тобто збільшуємо на одиницю значення обох полів).

Далі пишемо файл, що виводить лічильник на сторінки `show_stats.php` (рис.3.7). Файл `show_stats.php` ми використовуємо на кожній сторінці для того, щоб вивести статистичну інформацію.

```

<?Php
//·Вказуємо·кодування,·в·якій·буде·отримана·інформація·з·бази
@mysqli_query·($db,·'set·character_set_results·="utf8"');

//·Витягуємо·статистику·за·поточною·датою·(змінна·date·потрапляє·сюди·з·файлу·count.php,·який,·в·свою·чергу,
$res·=·mysqli_query·($db,·"SELECT·views`,·hosts`·FROM·visits`·WHERE·date`·="·$date"");
$row·=·mysqli_fetch_assoc·($res);

echo·'<p>·Унікальних·відвідувачів:·'.·$row·['hosts']·.'<br/>';
echo·'Переглядів:·'.·$row·['views']·.'</p>';

```

Рис.3.7 Файл show_stats.php

Тут ми просто отримуємо з бази статистику за поточною датою, після чого використовуємо ключі результуючого масиву \$row для виведення на екран кількості унікальних відвідувачів (поле host) і переглядів (поле views).

Створюємо файл "адміна"(рис.3.8). У цьому файлі ми представимо статистику відвідувань в табличному вигляді.

```

1 <?Php·include·'../inc/db.php';·?>
2
3 <H2>·Статистика·</·h2>
4
5 <P>·<a·href="·?interval=1">·За·сьогодні·</a>·</·p>
6 <P>·<a·href="·?interval=7">·За·останній·тиждень·</a>·</·p>
7
8 <Table·style·="border:·1px·solid·silver;">
9
10 <Tr>
11 ····<Td·style·="border:·1px·solid·silver;">·Дата·</·td>
12 ····<Td·style·="border:·1px·solid·silver;">·Унікальних·відвідувачів·</·td>
13 ····<Td·style·="border:·1px·solid·silver;">·Переглядів·</·td>
14 </·Tr>
15
16 <?·Php
17 //·Якщо·в·масиві·$_GET·є·елемент·interval·(тобто·був·клік·по·одній·з·посилань·вище)
18 if·($_GET·['interval'])
19 {
20   $·Interval·=$_GET·['interval'];
21
22   ····//·Якщо·в·якості·параметра·передано·не·число
23   ····if·(!·is_numeric($·interval))
24   ····{
25   ········echo·'<p>·<b>·Неприпустимий·параметр!·</b>·</p>';
26   ····}
27
28   ····//·Вказуємо·кодування,·в·якій·буде·отримана·інформація·з·бази
29   ····@mysqli_query($·db,·'set·character_set_results·="utf8"');
30
31   ····//·Отримуємо·з·бази·дані,·відсортувавши·їх·за·датою·в·зворотному·порядку·в·кількості·interval·штук
32   $·Res·=·mysqli_query($·db,·"SELECT·*·FROM·visits·ORDER·BY·`date`·DESC·LIMIT·$·interval");
33
34   ····//·Формуємо·висновок·рядків·таблиці·в·циклі
35   while($·row·=·mysqli_fetch_assoc($·res))
36   ····{
37   echo·'<tr>
38   <Td·style·="border:·1px·solid·silver;">·'·.$·Row·['date']·'·</Td>
39   <Td·style·="border:·1px·solid·silver;">·'·.$·Row·['hosts']·'·</Td>
40   <Td·style·="border:·1px·solid·silver;">·'·.$·Row·['views']·'·</Td>
41   </Tr>·';
42   }
43   }
44   ?>
45
46 </Table>

```

Рис.3.8 Файл «адміна»

Крім того, ми реалізуємо можливість отримувати статистику за певний період часу.

Спочатку еднаємося з базою. Потім формуємо два посилання з параметрами, які дозволять нам при кліці по ним отримувати дані за певний інтервал часу (в прикладі: за 1 день і за 1 тиждень).

Ми також перевіряємо глобальний масив \$_GET на наявність переданого нами параметра (interval) і, в разі, якщо значення цього параметра не є числом, ми виводимо відповідне попередження.

Якщо ж все нормально, то ми робимо запит до бази, витягуючи з неї все записи, відсортовані за датою в зворотному порядку в кількості штук, заданих в параметрі interval.

Після цього ми запусків цикл while і в ньому формуємо рядки таблиці, виводячи в їх осередки елементи результуючого масиву \$row (date, hosts і views), в результаті чого отримуємо список всіх відвідувань за вказаний період часу.

У фінальному результаті лічильника має такий вигляд (рис.3.9).

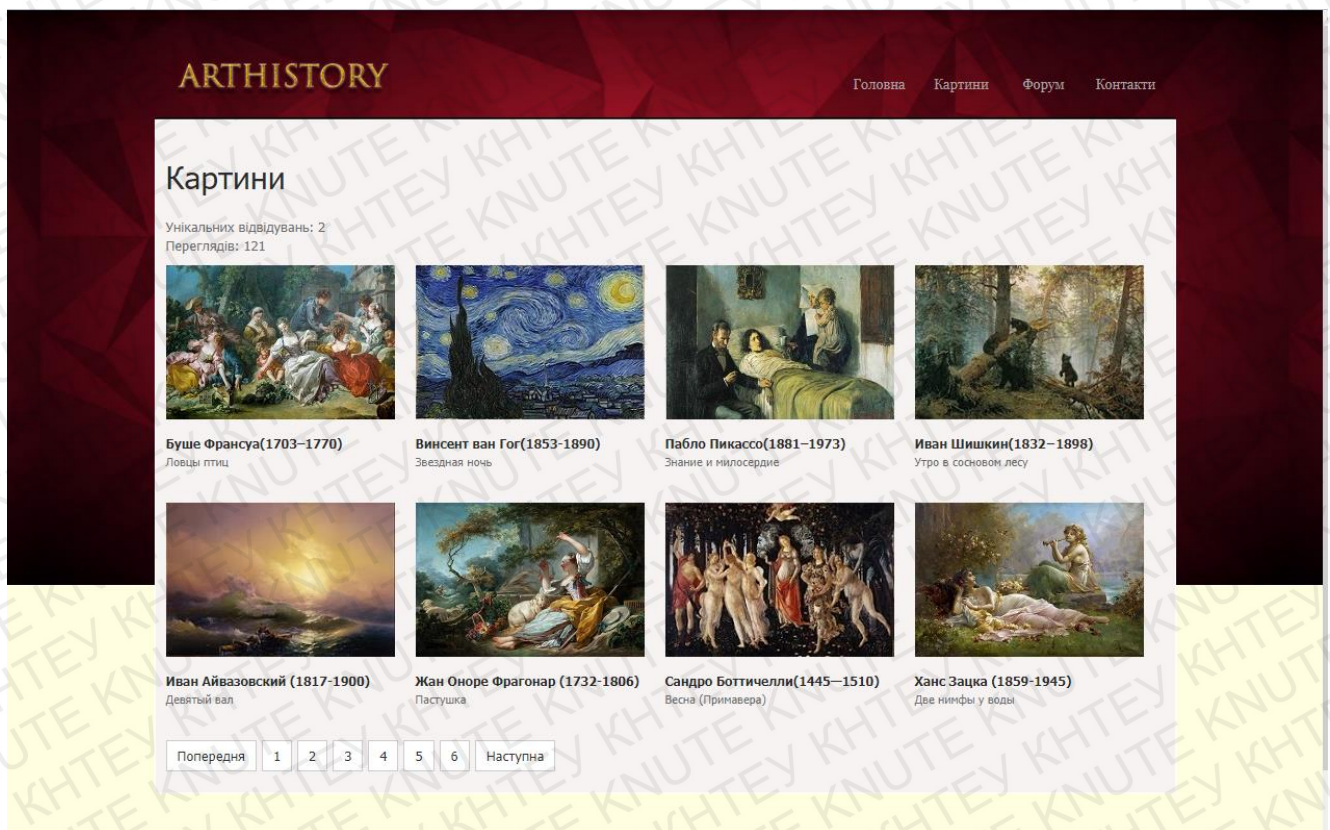


Рис.3.9 Фінальний вигляд лічильника на сайті

ВИСНОВКИ

У випускній кваліфікаційній роботі представлено результати теоретичних і практичних досліджень, що полягають у розробці програмного забезпечення, яке реалізує всі необхідні вимоги, які висувалися в постановці завдання до даної роботи. Розроблене програмне забезпечення в свою чергу є модифікуючим додатком, тобто при необхідності може бути доповнено різними функціями.

Готова програма відповідає всім вимогам і має зрозумілий призначений для користувача інтерфейс. В результаті проведених досліджень були отримані такі

ВИСНОВКИ:

1. Програма з моніторингу здійснює зчитування відвіданих веб-ресурсів відразу, в той момент, як тільки працівник на нього зайшов. Більш того, вона не тільки зчитує, а й зберігає всі адреси з браузера, а також копіює для «Адміністратора» всі завантажені з інтернету файли. Задача аналізу поведінки людини в інтернеті не нова, в сучасному світі достатньо програм, технологій, а також сервісів.
2. Для того щоб організувати роботу такого сервісу на своєму ресурсі, необхідно розмістити код конкретного лічильника на тих сторінках, інформація про відвідування яких необхідна.
3. У результаті на основі теоритичної інформації було розроблене програмне забезпечення по моніторингу відвідування web-ресурсів. Програмне забезпечення відслідковує кількість переглядів сторінки, а також унікальних відвідувачів. Переносить інформацію в базу даних та формує звіти. Звіти можуть бути за сьогоднішній день, за тиждень та за місяць.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Веб-аналитика : От систем статистики посещаемости до визуализации поведения посетителей сайтов [Электронный ресурс]. – Режим доступа : <http://altblog.ru/veb-analitika-ot-sistem-statistiki-poseshhaemosti-do-vizualizacii-povedeniya-posetitelej-sajtov/>.
2. Забарна Е. М. Система та критерії маркетингового аналізу сайтів вищих навчальних закладів [Електронний ресурс] / Е. М. Забарна, К. Ю. Соловійова // Ефективна економіка : електронне наукове фахове видання. – 2013. – №1. – Режим доступу : <http://www.economy.nayka.com.ua/?op=1&z=1748>.
3. Куліков В. О. Пошук історичної інформації в Інтернеті : посібник [для студентів історичних факультетів] / В. О. Куліков. – Харків : Вид-во ХНУ імені В. Н. Каразіна, 2007. – 91 с.
4. Мелихов Д. С. Анализ сайта : справочник веб-аналитика / Д. С. Мелихов, И. И. Сарматов. – К. : Аналитик Интеллект Сервис, 2011. – 209 с.
5. Мелихов Д. С. Веб-аналитика : шаг к совершенству / Д. С. Мелихов, И.И. Сарматов. – К. : Аналитик Интеллект Сервис, 2010. – 112 с.
6. Розкручування і просування сайта. Як визначити цільову аудиторію сайта? [Електронний ресурс]. – Режим доступу : <http://webstudio2u.net/ua/web-promotion/197-target-audience.html>.
7. Розкручування сайта в рейтингах [Електронний ресурс] // Buildsait : створення, просування, оптимізація сайта. – Режим доступу : <http://buildsait.pp.ua/index.php/promotion-web-site/49-web-site-promotion-in-ratings>.
8. Сарматов И. Зачем нужны счетчики? [Электронный ресурс] / Игорь Сарматов. – Режим доступа : <http://www.spybox.com.ua/blog/pochemu-sleduet-ubrat-vse-schetchiki-posesheniy-na-sayte.htm#axzz2yaGfXqEY>.
9. Сенник М. Проект. Лічильник відвідування сайта [Електронний ресурс] / М. Сенник. – Режим доступу : <http://www.senyk.poltava.ua/projs/counter.html>.
10. Скорпіо Л. Статистика сайта [Електронний ресурс] / Л. Скорпіо. – Режим доступу : <http://international-site.net/uk-ua/zakazchiku/5/statistika-sajtu>.

11. Що розуміється під оцінкою якості освітніх послуг? [Електронний ресурс] // Центр міжнародних проєктів НДІ прикладних інформаційних технологій “Євроосвіта”. – Режим доступу : <http://euroosvita.net/?category=21&id=229>.
12. Юрчак І. Ю. Конспект лекцій. Веб-проекування [Електронний ресурс] / І. Ю. Юрчак. – Режим доступу : <http://victoria.lviv.ua/html/gim/document.html>.

ДОДАТКИ

Файл count.php

```
<? Php include 'db.php';  
// Вказуємо кодування, в якій буде отримана інформація з бази  
@mysqli_query ($ db, 'set character_set_results = "utf8"');  
// Отримуємо IP-адреса відвідувача і зберігаємо поточну дату  
$ Visitor_ip = $ _SERVER [ 'REMOTE_ADDR'];  
$ Date = date ( "Y-m-d");  
// Дізнаємося, чи були відвідування за сьогодні  
$ Res = mysqli_query ($ db, "SELECT` visit_id` FROM `visits` WHERE` date` = '$  
date' ") or die (" Проблема при підключенні до БД ");  
// Якщо сьогодні ще не було відвідувань  
if (mysqli_num_rows ($ res) == 0)  
{  
    // Очищаємо таблицю ips  
    mysqli_query ($ db, "DELETE FROM` ips` ");  
    // Заносимо в базу IP-адреса поточного користувача  
    mysqli_query ($ db, "INSERT INTO` ips` SET `ip_address` = '$ visitor_ip'");  
    // Заносимо в базу дату відвідування і встановлюємо кількість переглядів і уник.  
    відвідувань в значення 1  
    $ Res_count = mysqli_query ($ db, "INSERT INTO` visits` SET `date` = '$ date',`  
hosts` = 1, `views` = 1");  
}  
// Якщо відвідування сьогодні вже були  
else  
{  
    // Перевіряємо, чи є вже в базі IP-адреса, з якого відбувається звернення  
    $ Current_ip = mysqli_query ($ db, "SELECT` ip_id` FROM `ips` WHERE`  
ip_address` = '$ visitor_ip' ");  
    // Якщо такий IP-адреса вже сьогодні був (тобто це не унікальний відвідувач)
```

```

if (mysqli_num_rows ($ current_ip) == 1)
{
    // Додаємо для поточної дати +1 перегляд (хіт)
    mysqli_query ($ db, "UPDATE` visits` SET `views` = ` views` + 1 WHERE `date`
= '$ date'");
}
// Якщо сьогодні такого IP-адреси ще не було (тобто це унікальний відвідувач)
else
{
    // Заносимо в базу IP-адреса цього відвідувача
    mysqli_query ($ db, "INSERT INTO` ips` SET `ip_address` = '$ visitor_ip'");
    // Додаємо в базу +1 унікального відвідувача (хост) і +1 перегляд (хіт)
    mysqli_query ($ db, "UPDATE` visits` SET `hosts` = ` hosts` + 1, `views` = `
views` + 1 WHERE `date` = '$ date'");
}
}
}

```

Файл show_stats.php

```

<? Php
// Вказуємо кодування, в якій буде отримана інформація з бази
@mysqli_query ($ db, 'set character_set_results = "utf8"');
// Витягуємо статистику за поточною датою (змінна date потрапляє сюди з файлу
count.php, який, в свою чергу, підключається в кожному з звичайних файлів)
$ Res = mysqli_query ($ db, "SELECT` views`, `hosts` FROM` visits` WHERE `date`
= '$ date'");
$ Row = mysqli_fetch_assoc ($ res);
echo '<p> Унікальних відвідувачів:'. $ Row ['hosts']. '<br />';
echo 'Переглядів:'. $ Row ['views']. '</ P>';

```

Файл db.php

```
<? Php
// Хост (зазвичай localhost)
$ Db_host = "localhost";
// Ім'я бази даних
$ Db_name = "stats";
// Логін для підключення до бази даних
$ Db_user = "root";
// Пароль для підключення до бази даних
$ Db_pass = "";
$ Db = mysqli_connect ($ db_host, $ db_user, "", $ db_name) or die ("Неможливо
підключитися до БД");
```

Файл «адміна»

```
<? Php include '../inc/db.php'; ?>
<H2> Статистика </ h2>
<P> <a href="?interval=1"> За сьогодні </a> </ p>
<P> <a href="?interval=7"> За останній тиждень </a> </ p>
<Table style = "border: 1px solid silver;">
<Tr>
  <Td style = "border: 1px solid silver;"> Дата </ td>
  <Td style = "border: 1px solid silver;"> Унікальних відвідувачів </ td>
  <Td style = "border: 1px solid silver;"> Переглядів </ td>
</ Tr>
<? Php
// Якщо в масиві GET є елемент interval (тобто був клік по одній з посилань вище)
```

```

if ($ _GET [ 'interval'])
{
$ Interval = $ _GET [ 'interval'];
// Якщо в якості параметра передано не число
if (! is_numeric ($ interval))
{
echo '<p> <b> Неприпустимий параметр! </ b> </ p>';
}
// Вказуємо кодування, в якій буде отримана інформація з бази
@mysqli_query ($ db, 'set character_set_results = "utf8"');
// Отримуємо з бази дані, відсортувавши їх за датою в зворотному порядку в
кількості interval штук
$ Res = mysqli_query ($ db, "SELECT * FROM `visits` ORDER BY `date` DESC
LIMIT $ interval");

// Формуємо висновок рядків таблиці в циклі
while ($ row = mysqli_fetch_assoc ($ res))
{
echo '<tr>
<Td style = "border: 1px solid silver;"> '. $ Row [ 'date']. '</ Td>
<Td style = "border: 1px solid silver;"> '. $ Row [ 'hosts']. '</ Td>
<Td style = "border: 1px solid silver;"> '. $ Row [ 'views']. '</ Td>
</ Tr> ';
}
}
?>
</ Table>

```