

**Київський національний торговельно-економічний університет**

**Кафедра комп'ютерних наук та інформаційних систем**

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**

на тему:

**«Проектування та створення інформаційно-аналітичної  
Web-системи оцінювання попиту  
на продукцію легкої промисловості»**

Студентки 4 курсу, 9 групи,  
спеціальності  
122 «Комп'ютерні науки»

*підпис студента*

Поліщук  
Аліни  
Олександрівни

Науковий керівник  
кандидат фізико-математичних наук,  
доцент

*підпис керівника*

Самойленко Ганна  
Тимофіївна

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 систем  
Спеціальність 122 «Комп'ютерні науки»

Зав. кафедри \_\_\_\_\_ **Затверджую**  
**Пурський О.І.**  
« » грудня 2020р.

## Завдання на випускну кваліфікаційну роботу студенту

**Поліщук Аліні Олександрівні**  
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи:  
«Проектування та створення інформаційно-аналітичної Web-системи оцінювання попиту на продукцію легкої промисловості»  
Затверджена наказом ректора від « » грудня 2020 р. № 4111
2. Строк здачі студентом закінченої роботи 29 травня 2021 року
3. Цільова установка та вихідні дані до роботи  
Мета роботи: Розробка інформаційно-аналітичної Web-системи оцінювання попиту, з урахуванням сучасних вимог . до продукції легкої промисловості  
Об'єкт дослідження: процес проектування інформаційно-аналітичної Web-системи оцінювання попиту  
Предмет дослідження: засоби створення інформаційно-аналітичної Web-системи оцінювання попиту .
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	Самойленко Г.Т.	16.12.2020 р.	16.12.2020р.
2	Самойленко Г.Т.	16.12.2020 р.	16.12.2020р.
3	Самойленко Г.Т.	16.12.2020 р.	16.12.2020 р.

6. Зміст випускного кваліфікаційного проекту (перелік питань за кожним розділом)

### ВСТУП

#### РОЗДІЛ 1. 1. Сучасний стан проектування інформаційних систем

##### 1.1. Роль інформаційних систем в діяльності підприємств

##### 1.2. Огляд підходів щодо організації інформаційно-аналітичних систем

#### Висновки до розділу

#### РОЗДІЛ 2. Організація розробки інформаційно-аналітичної Web-системи

##### 2.1. Моделювання інформаційного забезпечення

##### 2.2. Проектування інформаційно-аналітичної Web-системи оцінювання попиту

#### Висновки до розділу

#### РОЗДІЛ 3. Розробка інформаційно-аналітичної Web-системи оцінювання попиту на продукцію легкої промисловості

##### 3.1. Розробка користувацького інтерфейсу

##### 3.2. Програмування сайту

##### 3.3. Оцінка отриманих результатів проектування

#### Висновки до розділу

### ВИСНОВКИ

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ



## 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>	02.10.2020	02.10.2020
2	<i>Розробка та затвердження завдання на випускну кваліфікаційну роботу</i>	16.12.2020	16.12.2020
3	<i>Вступ</i>	03.02.2021	
4	<i>РОЗДІЛ 1. Сучасний стан проектування інформаційних систем.</i>	28.02.2021	
5	<i>РОЗДІЛ 2. <u>Організація розробки інформаційно-аналітичної Web-системи.</u></i>	06.04.2021	
6	<i>РОЗДІЛ 3. Розробка інформаційно-аналітичної Web-системи оцінювання попиту на продукцію легкої промисловості.</i>	12.05.2021	
7	<i>Висновки</i>	15.05.2021	
8	<i>Здача випускної кваліфікаційної роботи на кафедрі науковому керівнику</i>	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>	29.05.2021	
13	<i>Публічний захист випускної кваліфікаційної роботи</i>	За розкладом роботи ЕК	

8. Дата видачі завдання «16» грудня 2020 р.

9. Керівник випускної кваліфікаційної роботи

Самойленко Г.Т.

(прізвище, ініціали, підпис)

10. Гарант освітньої програми

Демідов П.Г.

(прізвище, ініціали, підпис)

11. Завдання прийняв до виконання студент-дипломник

Поліщук А.О.

(прізвище, ініціали, підпис)



## АНОТАЦІЯ

бакалаврської дипломної роботи Поліщук Аліни Олександрівни  
на тему «Проектування та створення інформаційно-аналітичної Web-  
системи оцінювання попиту».

Метою роботи є проектування та створення інформаційно-аналітичної Web-системи оцінювання попиту.

Для виконання роботи розглянуто головні принципи створення та розробки веб сайтів. Дослідження особливості побудови сайтів за допомогою систем керування вмістом Wordpress. Перед вибором системи керування вмістом для розробки сайту також була проведена порівняльна характеристика найпопулярніших безкоштовних систем керування вмістом: Joomla!, Drupal та Wordpress.

У ході виконання даної дослідницької роботи було розроблено імітаційну модель аналітичної веб системи та налаштовано API Google Trends, API Yandex Wordstst та API Megaindex.

Загальний обсяг роботи: 49 сторінок, 15 рисунків, 1 таблиця, 22 посилання.

Ключові слова: аналітична веб-система, веб сайт, Wordpress, імітайне моделювання.

## ANNOTATON

Bachelor`s degree of Polishchuk Alina Alexandrovna  
entitled " Design and creation of information-analytical Web-system of demand  
estimation"

The purpose of the work is to design and create an information-analytical Web-  
system of demand assessment.

To perform the work, the main principles of creating and developing websites are considered. Research features of building sites using Wordpress content management systems. Before choosing a content management system for site development, a comparative description of the most popular free content management systems was also conducted: Joomla !, Drupal and Wordpress.

In the course of this research work, a simulation model of an analytical web system was developed and the Google Trends API, the Yandex Wordstst API and the Megaindex API were configured.

Total volume of work: 49 pages, 15 figures, 1 table, 22 references.

Keywords: analytical web system, web site, Wordpress, simulation.



## ЗМІСТ

ВСТУП .....	10
РОЗДІЛ 1 Сучасний стан проектування інформаційних систем .....	12
1.1 Роль інформаційних систем в діяльності підприємств .....	12
1.2 Огляд підходів щодо організації інформаційно-аналітичних систем .....	13
1.3 Етапи розробки веб-сайтів .....	15
РОЗДІЛ 2 Організація розробки інформаційно-аналітичної Web-системи .....	19
2.1 Аналіз web-технологій для створення сайтів .....	19
2.2 Моделювання інформаційного забезпечення .....	27
2.3 Імітаційне моделювання аналітичної веб-системи .....	28
2.4 Алгоритм роботи інформаційно-аналітичної Web-системи .....	31
РОЗДІЛ 3 Розробка інформаційно-аналітичної Web-системи оцінювання попиту на продукцію легкої промисловості .....	37
3.1 Проектування інформаційно-аналітичної web-системи .....	37
3.2 Розробка користувацького інтерфейсу .....	40
3.3 Оцінка отриманих результатів проектування .....	46
ВИСНОВКИ .....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	49
ДОДАТОК А .....	52
ДОДАТОК Б .....	56



ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І  
ТЕРМІНІВ

CMS — Content management system.

ПФ СУБД — Повнофункціональна система управління базами даних.

SQL — Structured Query Language.

PHP — Hypertext Preprocessor.

SEO — Search engine optimization.

WWW — World Wide Web.

HTML — HyperText Markup Language.

XHTML — eXtensible Markup Language.

XML — Extensible Markup Language.

SGML — Standard Generalized Markup Language.

YML — Yandex Market XML.

CSS — Cascading Style Sheets.

## ВСТУП

В сучасних реаліях успіх будь-якого підприємства залежить від інформаційного забезпечення, що дозволяє підтримувати ефективну комунікацію з клієнтами та в середині самого підприємства, а також ресурсів, що дозволяють створювати, ефективно аналізувати великий обсяг інформації.

Створення аналітичного веб-сайту – один із альтернативних методів аналітики, яка базується на попиті користувачів та на попередніх пошукових запитах. Головною перевагою такої системи є візуальне порівняння, що допоможе подальшому прийнятті рішення.

Веб-рішення все частіше інтегруються в інформаційну інфраструктуру підприємства, стають його невід'ємною частиною. Веб-орієнтовані інформаційні системи не потребують нічого від користувача. Веб-додатки, з яких побудована інформаційна система не пред'являють ніяких вимог до апаратної платформи.

Ефективність роботи сайту залежить від різних факторів, основні з яких:

- адаптивний та дружній інтерфейс;
- доступність викладеної інформації;
- адаптивність БД до пошукових систем;

*Актуальність* дослідження полягає в тому, що створення веб-сайту дозволить економити ресурси, скорочувати час на аналітику та збільшити кількість операцій одночасно.

Під *об'єктом* дослідження розуміємо процес проектування аналітичної веб системи.

*Предметом* дослідження постають засоби створення аналітичної веб системи.

*Мета* дослідження – полягає в обґрунтуванні та розробці веб-сайту, з урахуванням сучасних світових тенденцій побудови організаційних та функціональних інформаційних структур підприємств.

*Для досягнення поставленої мети дослідження головними завданнями є:*

- аналіз трендів серед топових веб-сайтів;
- обрання оптимального методу створення веб-сайту;
- дослідження сучасних методів розробки веб-сайтів;
- проектування та розробка веб-сайту.

Методи наукових досліджень, що використовувались для написання роботи:

- емпіричні (порівняння, вимірювання);
- комплексні (моделювання);
- теоретичні (сходження від абстрактного до конкретного).

Практична цінність даної роботи полягає в оптимізації процесу підрахунку показників попиту на заданий товар шляхом поєднання різних веб ресурсів в один, в обхід санкцій аналізованих сайтів.



## **РОЗДІЛ 1 Сучасний стан проектування інформаційних систем**

### **1.1 Роль інформаційних систем в діяльності підприємств**

Інформаційні технології активно входять в сучасне життя, в тому числі в організацію виробничого процесу, діяльність якого неможлива без відповідної системи управління. У сучасних умовах ефективне управління являє собою цінний ресурс організації поряд з фінансовими, матеріальними, людськими та іншими ресурсами. Підвищення ефективності управлінської діяльності стає одним з напрямків вдосконалення діяльності підприємства в цілому. Найбільш очевидним способом підвищення ефективності протікання трудового процесу є його автоматизація. Застосування інтегрованої інформаційної системи, яка відповідала б вимогам підприємства дозволяє керівнику мінімізувати витрати і підвищити оперативність управління підприємством в цілому. Інформаційні ресурси в управлінні підприємством мають першорядне значення як обов'язкова передумова успішного виробничого процесу [1].

Інформаційна система - інформаційний контур разом із засобами збору, передачі, обробки та зберігання інформації, а також персоналом, що здійснює ці дії з інформацією. Місія інформаційних систем - виробництво потрібної для організації інформації для забезпечення ефективного управління всіма її ресурсами, створення інформаційного та технологічного середовища для здійснення управління організацією. З точки зору управлінських рішень інформаційна система розглядається як спеціальний інструмент в руках посадових осіб, що реалізують переробку інформації в процесі професійної діяльності. Ефективність комплексної автоматизації управлінських структур в основному залежить від ступеня охоплення всіх сфер корпоративного управління.

Багато в чому саме через це в останні роки стала настільки популярною тенденція побудови і впровадження комплексних корпоративних



інформаційних систем. Найпоширенішою корпоративною інформаційною системою в розвинених країнах з ринковою економікою є система планування ресурсів підприємства, створена для обробки ділових операцій організації та для сприяння комплексному та оперативному (в режимі реального часу) планування, виробництва та надання послуг клієнтів. Зокрема, ERP-системи мають такі характеристики:

- готове програмне забезпечення, розроблене для середовища клієнт-сервер, як традиційної, так і базується на інтернет-технологіях;
- можливість інтегрувати більшість бізнес-процесів;
- обробка більшості з усіх ділових операцій організації;
- використання бази даних всього підприємства;
- забезпечення доступу до будь-яких даних організації в режимі реального часу;
- можливість інтегрувати обробку ділових операцій і дій з планування.[2]

## **1.2 Огляд підходів щодо організації інформаційно-аналітичних систем**

Будь-яка теоретична або практична сфера діяльності використовує властиві тільки їй способи вирішення поставлених завдань. Ці способи називаються методами.

Метод – це спосіб досягнення якої-небудь мети, вирішення конкретної задачі; сукупність прийомів або операцій практичного або теоретичного освоєння дійсності.

Застосування ефективних методів і засобів створення інформаційної системи, правильна побудова технології її створення дають змогу суттєво знизити витрати та скоротити терміни розробки, забезпечуючи якісне створення системи обробки даних, які відповідають вимогам користувачів [3].

При створенні інформаційної системи використовують цілий комплекс методів і засобів.

Засоби створення інформаційної системи – це типові проектні рішення, пакети прикладних програм, типові проекти чи інструментальні засоби проектування інформаційної системи [4].

Завданням випускної кваліфікаційної роботи є: проектування та створення інформаційно-аналітичної Web-системи оцінювання попиту. При розробці необхідно, щоб API (Application Programming Interface) інформаційної системи працювало на остатньому рівні без використання технології VPN. Зважаючи на те, що система має отримувати аналітичну інформацію з ресурсів, що можуть блокуватися на окремих територіях, модулі API повинні бути прописані на першому рівні підключення до аналітичної системи, що дозволить оптимізувати швидкість звернення окремих API та зменшити час на останньому рівні роботи.

Під час розробки веб-сайту необхідно звернути увагу на наступні проблеми, а саме:

- кількість інформації, яка буде одночасно аналізуватись;
- дослідження типових задач по матеріалам з літературних джерел;
- обрати оптимальну методику розв'язку.
- необхідно з'ясувати наявність ресурсів, необхідних для вирішення задачі на комп'ютері;
- потрібно проаналізувати задачу та визначити її масштаб, для можливості поділу її на окремі блоки;
- необхідно створити алгоритм розв'язку задачі.

Всі методи розробки сайтів можна умовно розділити на 2 основні групи, в залежності від типу автоматизації:

- Програмований – це методи ручного написання сайтів на одній або декількох мовах web-програмування. Програмовий метод розробки сайту передбачає самостійне створення дизайну сайту, а також

програмного коду сайту. Можливе використання будь-яких графічних редакторів. Можливе використання вже готових шаблонів сайтів. Вибір мови визначається знаннями, якими володіють програмісти, що реалізують проект.

- Візуальний метод містить у собі опції автоматизованого створення сайтів, за допомогою спеціальних конструкторів сайтів, або ж систем керування контентом (CMS).

Методи розробки сайтів з використанням CMS. CMS являє собою певну готову візуальну й програмну структуру, яка містить різноманітні доповнення. Користувач заповнює шаблон контентом, та отримує повноцінний веб-сайт.

При створенні сайтів з використанням візуального методу необхідно вибрати інструменти – редактори візуального конструювання, які забезпечують користувачеві можливість працювати без підключення до мережі Інтернет.

Візуальні редактори діляться на дві великі групи:

- online-редактори – це спеціальні механізми для створення сайтів (як правило, дуже простих), які розташовані безпосередньо в Інтернеті, найчастіше на сервісах, що надають хостинг (платний або безкоштовний). Наприклад, wix.com чи sites.google.com. [15].
- Автоматизовані методи розробки передбачають поділ структури веб-сайту на «дизайн» та «контент». Розробник отримує доступ окремо до контенту, окремо до дизайну. Під час ручної розробки сайту поділу структури на два окремі розділи, дизайн і вміст - не відбувається.

### **1.3 Етапи розробки веб-сайтів**

В інтернеті можна знайти багато варіантів класифікацій розробки веб сайтів за етапами, проте всі вони мають такі позиції як:



- **Планування.** На етапі планування, перш за все, слід визначити призначення майбутнього сайту: це буде персональний сайт або сайт організації, електронна енциклопедія чи сайт бібліотеки, сайт для дистанційного навчання тощо. Тут же доцільно визначити, буде сайт тематичним чи різні його сторінки будуть присвячені різним темам і яким саме.
- **Розробка.** Наступний крок — розробка структури сайту. При розробці структури сайту слід визначитися з необхідною кількістю сторінок та встановити зв'язки між ними. Розрізняють лінійну, ієрархічну та довільну структури сайту. [6]

Лінійну структуру веб-сайту доцільно використовувати у разі послідовного представлення інформації, наприклад, про товари та послуги або матеріали навчального посібника. Перегляд таких сайтів здійснюється послідовно: від початкової (головної) до останньої сторінки. Кожна сторінка має посилання тільки на одну, наступну сторінку сайту. Інколи, для зручності навігації по сайту до сторінки також додається посилання на попередню сторінку.

При ієрархічній структурі створюється одна сторінка (головна), яка не має попередніх, решта сторінок мають лише одну попередню сторінку (рис. 1.1).

При ієрархічній структурі кожна сторінка може містити посилання на довільну кількість сторінок сайту. Така структура найкраще підходить для сайтів, що містять різну за тематикою інформацію: каталогів, збірань статей з різних тем або добірок послань.



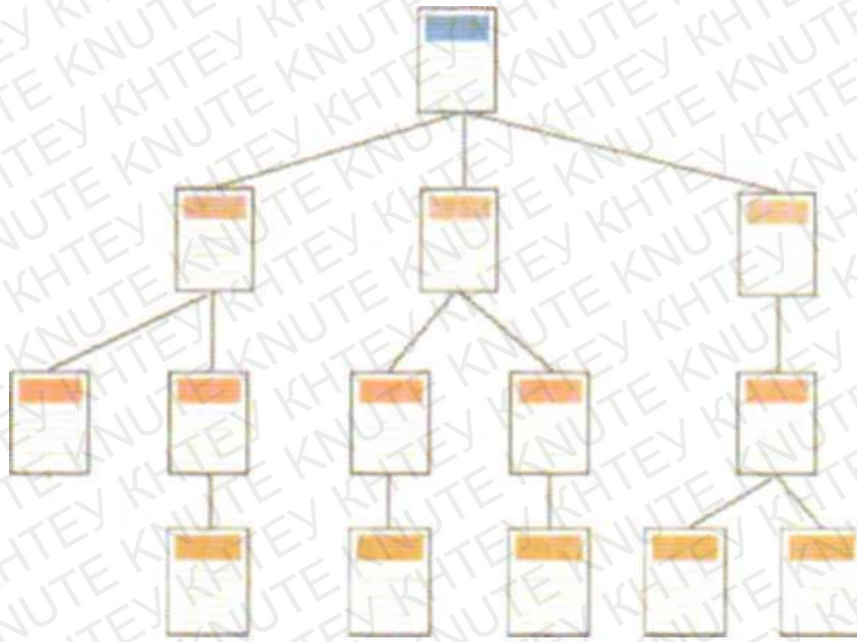


Рис. 1.1. Ієрархічна структура

Найчастіше для створення сайтів використовують довільну структуру. При такій структурі сайту його сторінки пов'язані між собою довільним чином (Рис. 1.2). У сайту довільної структури можна виділити фрагменти, які є лінійними або ієрархічними [7].

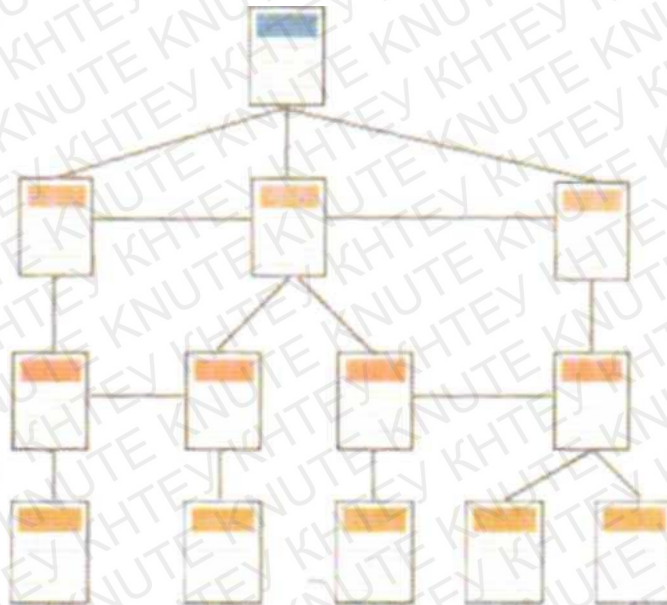


Рис. 1.2. Довільна структура

Створення. На етапі створення відбувається наповнення веб-сторінок конкретними матеріалами, а також створення гіперпосилань для зручної навігації сайтом. При цьому потрібно слідувати правилам оформлення (дизайну) веб-сторінок, якими ви користувалися при створенні презентацій.

- Тестування. Дуже важливим при створенні сайту є етап тестування. На цьому етапі потрібно перевірити:
- чи правильно працюють усі гіперпосилання;
- чи зручною є навігація сайтом;
- чи відкриваються при відкритті сторінок графічні зображення;
- чи зручно розташовані для сприйняття матеріали на сторінках тощо.

У разі необхідності, потрібно внести зміни в наповнення або структуру сайту, змінити гіперпосилання.

- Розміщення. На наступному етапі потрібно визначити, де буде розміщено створений вами сайт. Веб-сайт можна розмістити:
- на власному сервері установи;
- на сервері вашого провайдера;
- на сервері організації, яка спеціалізується у наданні послуг розміщення сайтів користувачам Інтернету;
- на сервері, який надає послуги вільного і безкоштовного розміщення сайтів [8].

## РОЗДІЛ 2 Організація розробки інформаційно-аналітичної Web-системи

### 2.1 Аналіз web-технологій для створення сайтів

Створення інтерактивних компонентів – це завдання для вебпрограмування. Мова програмування – штучна мова, призначена для запису комп'ютерних програм. Мова програмування визначає набір лексичних, синтаксичних і семантичних правил, що визначають зовнішній вигляд програми і дії, які виконає виконавець (зазвичай - ЕОМ) під її управлінням [9]. HTML не є мовою програмування, це мова розмітки тексту, що використовує спеціальні оператори – теги (tag) чи інша назва дескриптори (descriptor) для розмітки текстового документа. Ці позначки вказують в якому вигляді буде виведено текстовий чи інший елемент у вікні браузера.

HTML (Hyper text Markup Language – Мова гіпертекстової розмітки) – це мова опису структури сторінок документів, яка дозволяє звичайний текст формувати в абзаци, заголовки, списки та інші структури, створювати посилання на інші сторінки. Це текстова мова, в якій інструкції з форматування, що називаються тегами, вбудовані в розділи документа, які містять конкретну інформацію. Теги повідомляють браузерам, як формувати і представляти інформацію на екрані. За основу моделі розмітки документів у HTML прийнята тегова модель. Тегова модель описує документ як сукупність контейнерів, кожен з яких починається і закінчується тегами. Теги HTML-документів в основному є простими і зрозумілими для використання, оскільки вони створені за допомогою загальноживаних слів англійської мови, зрозумілих скорочень і позначень [10].

XML (англ. EXtensible Markup Language) - розширювана мова розмітки. Рекомендований Консорціумом Всесвітньої павутини (W3C). Специфікація XML описує XML-документи і частково описує поведінку XML-процесорів (програм, які читають XML-документи і забезпечують доступ до їх вмісту).



XML розроблявся як мова з простим формальним синтаксисом, зручний для створення і обробки документів програмами і одночасно зручний для читання і створення документів людиною, з підкресленням націленості на використання в Інтернеті. Мова називається розширюваним, оскільки ним не фіксується розмітка, яка використовується в документах: розробник вільний створити розмітку відповідно до потреб конкретної області, будучи обмеженим лише синтаксичними правилами мови. Розширення XML – це конкретна граматики, створена на базі XML і представлена словником тегів і їх атрибутів, а також набором правил, що визначають які атрибути і елементи можуть входити до складу інших елементів. Поєднання простого формального синтаксису, зручності для людини, розширюваності, а також базування на кодуваннях Юнікод для подання змісту документів привело до широкого використання як власне XML, так і безлічі похідних спеціалізованих мов на базі XML в найрізноманітніших програмних засобах [11].

XHTML (англ. Extensible hypertext markup language – мова розмітки гіпертексту) - сімейство мов розмітки веб-сторінок на основі XML, які повторюють і розширюють можливості HTML 4. Специфікації XHTML 1.0 і XHTML 1.1 є рекомендаціями консорціуму Всесвітньої павутини. Головна відмінність XHTML від HTML полягає в обробці документа. Документи XHTML обробляються своїм модулем (парсером) аналогічно документам XML. В процесі цієї обробки помилки, допущені розробниками, не справляються. XHTML відповідає специфікації SGML, оскільки XML є її тирсою. HTML володіє безліччю особливостей в процесі обробки і фактично перестав ставитися до сімейства SGML, що і закріплено в чернетці специфікації HTML 5 [12].

CSS (Cascading Style Sheets – каскадні таблиці стилів) - формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки [16]. Переважно використовується як засіб опису, оформлення зовнішнього вигляду веб-сторінок, написаних за допомогою мов розмітки



HTML і XHTML, але може також застосовуватися до будь-яких XML-документах, наприклад, до SVG або XUL. CSS використовується творцями веб-сторінок для завдання кольорів, шрифтів, розташування окремих блоків і інших аспектів представлення зовнішнього вигляду цих веб-сторінок. Основною метою розробки CSS було розділення опису логічної структури веб-сторінки (яке проводиться за допомогою HTML або інших мов розмітки) від опису зовнішнього вигляду цієї веб-сторінки (яке тепер проводиться за допомогою формальної мови CSS). Такий поділ може збільшити доступність документа, надати велику гнучкість і можливість управління його поданням, а також зменшити складність і повторюваність в структурному вмісті. Крім того, CSS дозволяє представляти один і той же документ в різних стилях або методах виведення, таких як екранне уявлення, друковане подання, читання голосом (спеціальним голосовим браузером або програмою читання з екрану), або при виведенні пристроями, що використовують шрифт Брайля.

Веб-програмування здійснюється за допомогою спеціальних програмних засобів - скриптів. Ці програмні засоби поділяються на два основних види: серверні і клієнтські. Серверні скрипти виконуються на стороні сервера, тобто того комп'ютера, на якому розміщений сайт. Вони виконуються ще до завантаження сторінок сайту на комп'ютер користувача. У свою чергу, клієнтські скрипти виконуються на комп'ютері клієнта вже після завантаження сторінки з сервера і не вимагають її додаткової перезавантаження. Мови програмування, на яких виконуються і ті, і інші скрипти різні. Деякі з мов використовуються тільки для створення серверних скриптів, інші – тільки для клієнтських, а багато мов - для тих і інших [17].

PHP – це широко використовувана мова сценаріїв загального призначення з відкритим вихідним кодом. Говорячи простіше, PHP це мова програмування, спеціально розроблений для написання web-додатків (сценаріїв), що виконуються на Web-сервері [18]. Аббревіатура PHP означає "Hypertext Preprocessor (Препроцесор Гіпертексту)". Синтаксис мови бере початок з C,

Java і Perl. PHP досить простий для вивчення. Перевагою PHP є надання веб-розробникам можливості швидкого створення динамічно генеруються веб-сторінок. Значною відмінністю PHP від будь-якого коду, що виконується на стороні клієнта, наприклад JavaScript, є те, що PHP-скрипти виконуються на стороні сервера. Ви навіть можете налаштувати свій сервер таким чином, щоб HTML-файли оброблялися процесором PHP так, що клієнти навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл або результат виконання скрипта [14].

Python – високорівнева мова програмування загального призначення, орієнтований на підвищення продуктивності розробника і читання коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій. [13] Python підтримує кілька парадигм програмування, в тому числі структурний, об'єктно-орієнтоване, функціональне, імперативне і аспектно-орієнтоване. Основні архітектурні риси - динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень і зручні високорівневі структури даних. Код в Python організовується у функції та класи, які можуть об'єднуватися в модулі (вони в свою чергу можуть бути об'єднані в пакети). Еталонної реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ [20]. Він поширюється під вільною ліцензією Python Software Foundation License, що дозволяє використовувати його без обмежень в будь-яких додатках, включаючи пропріетарні [21]. Є реалізації інтерпретаторів для JVM (з можливістю компіляції), MSIL (з можливістю компіляції), LLVM та інших. Проект PyPy пропонує реалізацію Python з використанням JIT-компіляції, яка значно збільшує швидкість виконання Python-програм. Python – активно розвивається мова програмування, нові версії (з додаванням / зміною мовних властивостей) виходять приблизно раз в два з половиною роки. Внаслідок цього і деяких інших причин на Python відсутні стандарт ANSI, ISO або інші офіційні стандарти, їх роль виконує CPython JavaScript –

мульти-парадигменна мова програмування. Підтримує об'єктноорієнтована, імперативний і функціональний стилі. Є реалізацією мови ECMAScript (стандарт ECMA-262 [22]).

JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить в браузерах як мова сценаріїв для додання інтерактивності веб-сторінок [12, С. 32]. Основні архітектурні риси: динамічна типізація, слабка типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу. На JavaScript вплинули багато мов, при розробці була мета зробити мову схожим на Java, але при цьому легким для використання непрограмістів. Мовою JavaScript не володіє будь-яка компанія або організація, що відрізняє його від ряду мов програмування, використовуваних в веб-розробці [12]. JavaScript спочатку створювався для того, щоб зробити web-сторінки «живими». Програми на цій мові називаються скриптами. У браузері вони підключаються безпосередньо до HTML і, як тільки завантажується сторінка - тут же виконуються. SQL (англ. Structured query language – «мова структурованих запитів») – декларативна мова програмування, застосовувана для створення, модифікації та управління даними в реляційній базі даних, керованої відповідною системою управління базами даних [15].

SQL є перш за все інформаційно-логічною мовою, призначеною для опису, зміни і вилучення даних, що зберігаються в реляційних базах даних. SQL можна назвати мовою програмування, при цьому він не є Тьюринг-повним, але разом з тим стандарт мови специфікацією SQL / PSM передбачає можливість його процедурних розширень. Спочатку SQL був основним способом роботи користувача з базою даних і дозволяв виконувати наступний набір операцій:

- створення в базі даних нової таблиці;
- додавання в таблицю нових записів;
- зміна записів;



- видалення записів;
- вибірка записів з однієї або декількох таблиць (відповідно до заданого умовою);
- зміна структур таблиць.

Згодом SQL ускладнився - збагатився новими конструкціями, забезпечив можливість опису та управління новими збереженими об'єктами (наприклад, індекси, уявлення, тригери і процедури) - і став набувати рис, властиві мовам програмування [15]. При всіх своїх змінах SQL залишається єдиним механізмом зв'язку між прикладним програмним забезпеченням і базою даних. У той же час сучасні СУБД, а також інформаційні системи, що використовують СУБД, надають користувачеві розвинені засоби візуального побудови запитів.

MySQL – вільна реляційна система управління базами даних [16]. Розробка та підтримка сайту MySQL здійснює корпорація Oracle, яка отримала права на торговельну марку разом з поглиненої Sun Microsystems, яка раніше придбала шведську компанію MySQL AB. Продукт поширюється як під GNU General Public License, так і під власною комерційною ліцензією. Крім цього, розробники створюють функціональність за замовленням ліцензійних користувачів. Саме завдяки такому замовленню, майже у найбільш ранніх версіях, з'явився механізм реплікації. MySQL є рішенням для малих і середніх додатків. Входить до складу серверів WAMP, AppServ, LAMP і в портативні збірки серверів Денвер, XAMPP, VertrigoServ. Зазвичай MySQL використовується як сервер, до якого звертаються локальні або видалені клієнти, проте в дистрибутив входить бібліотека внутрішнього сервера, що дозволяє включати MySQL в автономні програми. Гнучкість СУБД MySQL забезпечується підтримкою великої кількості типів таблиць: користувачі можуть вибрати як таблиці типу MyISAM, що підтримують повнотекстовий пошук, так і таблиці InnoDB, що підтримують транзакції на рівні окремих записів.



Більш того, СУБД MySQL поставляється із спеціальним типом таблиць EXAMPLE, що демонструє принципи створення нових типів таблиць. Завдяки відкритій архітектурі і GPL-ліцензуванню, в СУБД MySQL постійно з'являються нові типи таблиць [17]. Тож, веб-програмування – це створення сайтів і програм, які працюють в мережі. Важливою ланкою між програмістом і програмами є мова програмування, а сполучною ланкою між веб-розробником і сайтами - мова веб-програмування.

Сучасні інформаційні системи цілковито створені на CMS. Для реалізації даного проекту необхідно використати CMS, так як вони включають в себе інтеграцію з пошуковими системами та аналітиками найпопулярніших пошукових систем.

Drupal (друпал) - система керування вмістом (CMS), написана на мові PHP і використовує як сховище даних реляційну базу даних (підтримуються MySQL, PostgreSQL та інші). Drupal можна застосовувати для побудови різних сайтів, починаючи від блогів, новинних порталів, інформаційних сайтів, форумів і закінчуючи соціальними мережами. Drupal є вільним програмним забезпеченням, захищеним ліцензією GPL. На рис.2.1 показано вигляд

цієї

системи

[6].

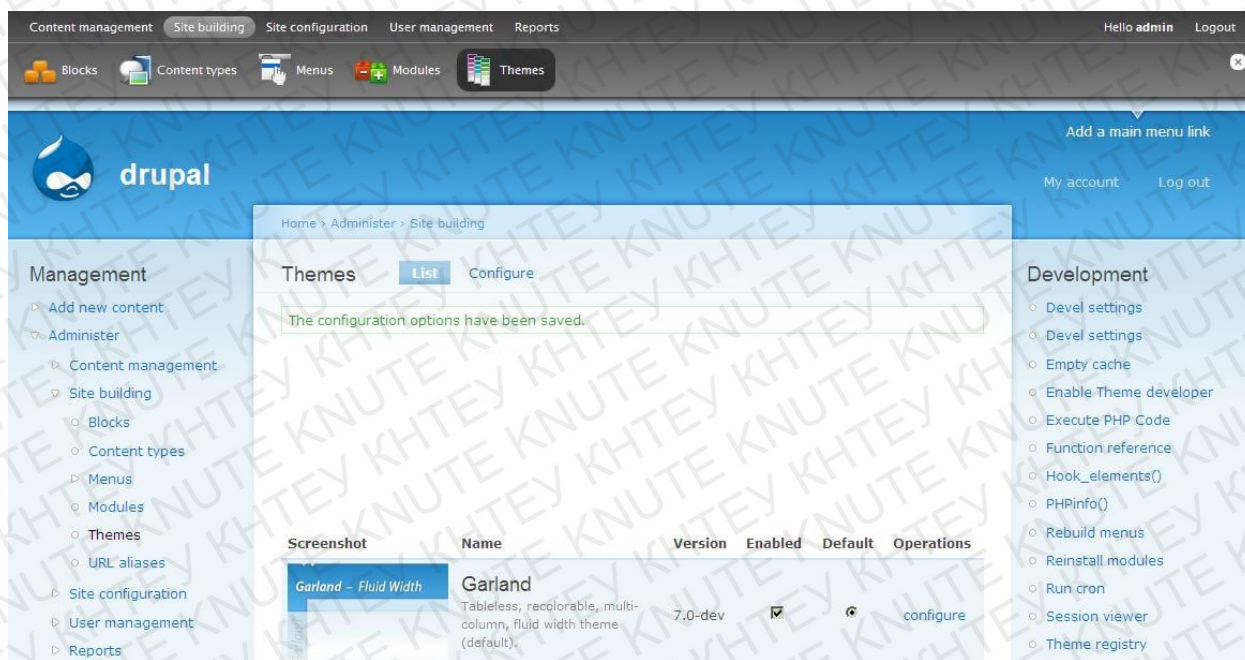


Рис. 2.1. Панель системи Drupal

Joomla! - Система керування вмістом, написана на мовах PHP і JavaScript, що використовує в якості сховища базу даних MySQL. Є вільним програмним забезпеченням, поширюваним під ліцензією GNU GPL. На рис. 2.2 показано вигляд панелі керування Joomla!.

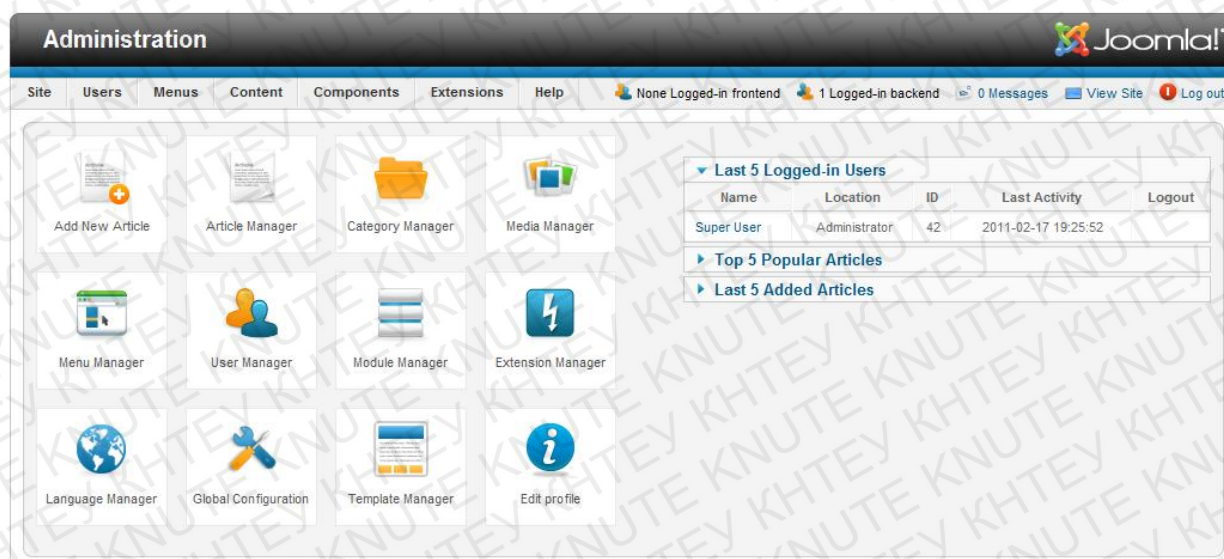


Рис. 2.2. Вигляд панелі керування Joomla!

WordPress - система керування вмістом сайту (CMS) з відкритим вихідним кодом, поширювана під GNU GPL . Написана на мові PHP, в якості бази даних використовує MySQL. Сфера застосування - від блогів до досить складних новинних ресурсів і навіть Інтернет-магазинів. Вбудована система «тем» і «плагінів» разом з вдалою архітектурою дозволяє конструювати практично будь-які проекти. На рис. 2.3 показано вигляд системи WordPress.



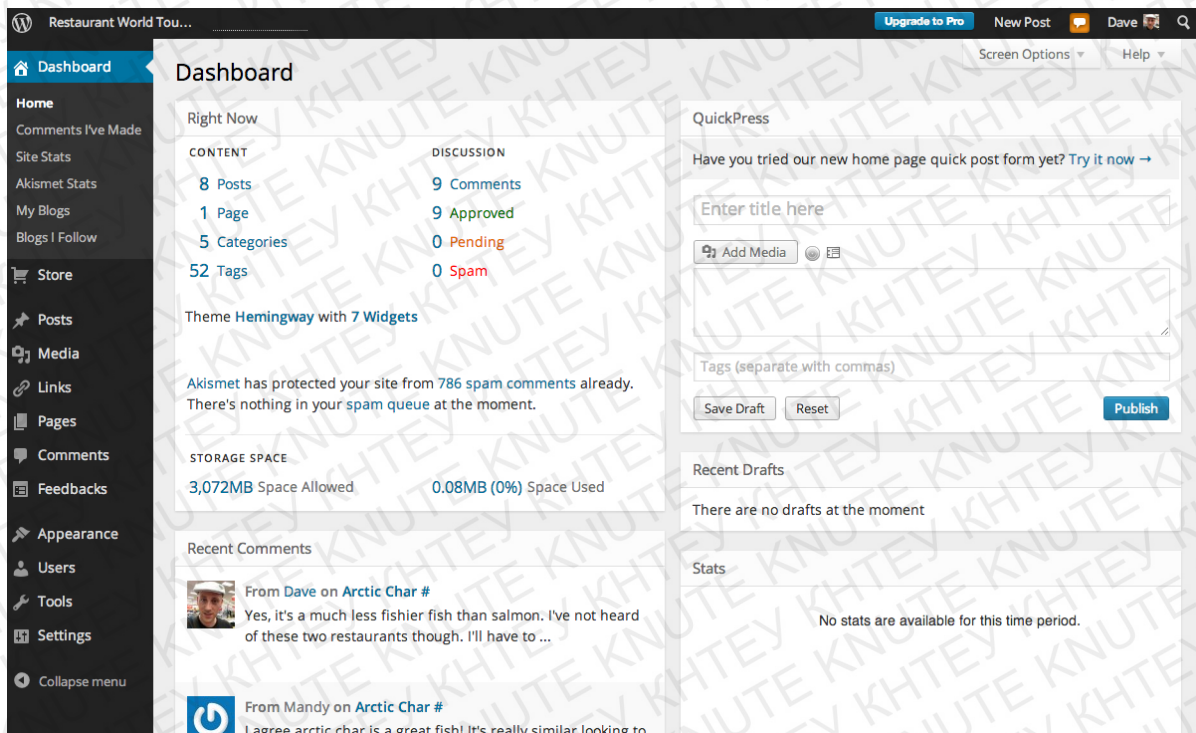


Рис 2.3. Вигляд системи WordPress

## **2.2 Моделювання інформаційного забезпечення**

Структуру сайту умовно можна поділити на два рівні – логічний і фізичний.

Проектування і розробка сайтів включає наступні етапи:

- Затвердження первинного технічного завдання розробки сайту.
- Визначення структурної схеми сайту та створення моделі сайту.
- Веб-дизайн – створення графічних елементів сайту, стилів (вибір кольорової гамми, визначення стилю текстової інформації: шрифти, кольори, форматування, визначення стилю графічних об'єктів: рамок, кнопок, списків, тощо)
- Вибір готових систем та модулів та/або розробка чи вдосконалення програмного коду, модулів та інших елементів сайту.
- Після розробки і введення у експлуатацію сайти потребують супроводу: технічного, а інколи і адміністративного. Це було втілено за допомогою комплексу програмних засобів WordPress.



Метою роботи є проектування та створення інформаційно-аналітичної Web-системи оцінювання попиту. Створення інформаційно-аналітичної системи буде починатися з розробки логічної структури в програмному комплексі Erwin, що дозволить побачити логічну схему взаємодії між усіма ланками. Наступним етапом буде формування аналітичної сторінки. Вона буде формуватися за рахунок створення поля для вводу даних та 3х блоків аналітичних показників. Кожен з блоків за допомогою API буде демонструвати аналізовані данні аналітичних програм пошукової мережі. Саме це допоможе порівняти попит на пропозицію серед декількох аудиторій.

### **2.3 Імітаційне моделювання аналітичної веб-системи**

Найпопулярнішою системою та найпростішою для освоєння є система AllFusion Process Modeler 7 (раніше BPwin) - інструмент для моделювання, аналізу, документування та оптимізації бізнес-процесів. AllFusion Process Modeler 7 можна використовувати для графічного представлення бізнес-процесів. Графічно представлена схема виконання робіт, обміну інформацією, документообігу візуалізує модель бізнес-процесу. Графічний виклад цієї інформації дозволяє перевести завдання управління організацією з області складного ремесла в сферу інженерних технологій.

Ця програма допомагає чітко документувати важливі аспекти будь-яких бізнес-процесів: дії, які необхідно зробити, способи їх здійснення і контролю, що вимагаються для цього ресурси, а також візуалізувати одержувані від цих дій результати. В основу продукту закладені загальновизнані методології моделювання, наприклад, методологія IDEF0. Простота і наочність моделей Process Modeler спрощує взаєморозуміння між усіма учасниками процесів. Поширеність самого AllFusion Process Modeler 7 дозволяє вести узгодження функціональних моделей з партнерами в електронному вигляді [18].

Найголовнішою особливістю використання програмного комплексу для створення імітаційної моделі аналітичної веб системи є розширена

практичність. Збір, представлення та аналіз бізнес-процесів підприємства може бути тривалою і віднімає багато часу процедурою. Покращена інтеграція з AllFusion ERwin AllFusion Data Modeler. Покращена синхронізація моделей з AllFusion ERwin DM включає опцію "Remove ERwin Links", яка дозволяє розробникам моделей зробити об'єкти і властивості, які раніше належали одній з моделей AllFusion ERwin DM, доступними для експорту в інші моделі.

Фільтрація даних при експорті в XML-формат. При збереженні моделі в файл XML-формату (XML Export), розробники моделей можуть фільтрувати об'єкти і властивості, вибрані для експорту, а так само можуть обмежувати вибрані елементи для специфічних діаграм в моделі процесу, незалежно від наявності підмоделей.

Функціональні можливості AllFusion Process Modeler 7 (BPwin):

- Підтримка декількох нотацій.
- Численні технології моделювання та підтримка різноманітних нотацій забезпечують всебічний і детальний аналіз і проектування в складній операційному середовищі.
- AllFusion Process Modeler 7 (BPwin) забезпечує комплексне використання та автоматичне узгодження найпопулярніших нотацій моделювання бізнес-процесів, потоків робіт IDEF3 і потоків даних (DFD).
- Інтуїтивно-зрозумілий графічний інтерфейс.
- Аналіз показників витрат і продуктивності. AllFusion Process Modeler 7 (BPwin) повністю підтримує методи розрахунку собівартості за обсягом господарської діяльності (функціонально-вартісний аналіз, ABC).
- Функціонально-вартісний аналіз, реалізований в AllFusion Process Modeler 7, дозволяє оцінити вартісні і тимчасові характеристики бізнес-процесів.

- AllFusion Process Modeler 7 (BPwin) дозволяє налаштувати збір додаткової суттєвої для вашого бізнесу інформації за допомогою UDP - властивостей, визначених користувачем. Введена інформація може бути відображена в звітах, згенерованих за допомогою генератора звітів. AllFusion Process Modeler 7 і експортований в інші програми, наприклад в Microsoft Word, Microsoft Excel.
- Організаційні графіки. Організаційна структура впливає на те, як описуються і виконуються бізнес-процеси. AllFusion Process Modeler 7 (BPwin) підтримує точний опис ролей, які визначають і розподіляють за категоріями завдання або роботи всередині бізнес-процесів [2].

Організаційні діаграми в AllFusion Process Modeler 7 використовують для визначення ролей, складу і відносин учасників бізнес-процесу:

- діаграми Swim Lane. Swim Lane допомагають візуалізувати посадові обов'язки, дозволяючи на одній діаграмі розглядати ролі і обов'язки ролі;
- діаграми дерева вузлів. Діаграми Node Tree показують ієрархію функцій в моделі і дозволяють розглянути всю модель цілком;
- діаграми FEO. Діаграми «тільки для експозиції» (FEO) можна використовувати для ілюстрації альтернативних точок зору, а також для відображення деталей, які порушують синтаксичні правила.

Наявність контекстно-залежною панелі інструментів, неможливість створення в моделі некоректних зв'язків, автоматична міграція граничних стрілок, можливість автоматичного відстеження дисбалансу граничних стрілок на дочірньої і батьківської діаграмах (тунелі), можливість автоматичної перевірки наявності імен стрілок і імен функціональних блоків, наявності виходів і управлінь; а також додаткові діаграми і всілякі звіти по вмісту моделі - все це допомагає автоматизувати процес побудови коректних моделей бізнес-процесів.



## 2.4 Алгоритм роботи інформаційно-аналітичної Web-системи

Інформаційно-аналітична система включає підрозділи і структури організації всіх рівнів, що підтримують бази даних оперативного доступу. Він являє собою низовий рівень генерації інформації, рівень внутрішніх і зовнішніх інформаційних джерел, виробляють «сиру» інформацію.

З джерел даних інформація переміщається на основі деякого регламенту в централізоване сховище. Ці дані зазвичай можна отримати з вихідних баз даних шляхом спеціальних перетворень, обчислень і агрегування. Перед завантаженням в сховище вся ця інформація повинна бути верифікована, щоб забезпечити цілісність і несуперечливість аналітичних даних. В сховищі зберігаються історичні дані, тобто дані за досить великий проміжок часу. В оперативній системі дані зберігаються в цілісному вигляді за обмежений проміжок, після чого вони відправляються в архів. Таким чином, завантаження даних з джерел в сховище здійснюється спеціальними процедурами, що дозволяють:

- витягувати дані з різних баз даних, текстових файлів;
- виконувати різні типи узгодження і очищення даних;
- перетворювати дані при переміщенні їх від джерел до сховища;
- завантажувати узгоджені і «очищені» дані в структури сховища.

Шар збереження даних призначений безпосередньо для зберігання значущою, перевіреної, узгодженої, несуперечливої і хронологічно цілісної інформації, яку з досить високим ступенем впевненості можна вважати достовірною.

Власне сховище даних не орієнтоване на рішення якої-небудь певній функціональній аналітичної задачі. Мета сховища даних - забезпечити цілісність і підтримувати хронологію всіляких даних, і з цієї точки зору воно нейтрально по відношенню до додатків.

Сховище даних найчастіше реалізується у вигляді реляційної бази даних, що працює під управлінням досить потужною реляційної СУБД. Така СУБД

повинна підтримувати ефективну роботу з терабайтними обсягами інформації, мати розвинені засоби обмеження доступу, забезпечувати підвищений рівень надійності і безпеки, відповідати необхідним вимогам по відновленню і архівації.

Для організації доступу до сховищ даних використовуються спеціалізовані робочі місця, підтримують необхідні технології як оперативного, так і довготривалого аналізу. Результати роботи аналітиків оформляються у вигляді звітів, графіків, рекомендацій і зберігаються як на локальному комп'ютері, так і в загальнодоступному вузлі локальної мережі.

Аналітична діяльність в рамках корпорації досить різноманітна і визначається характером вирішуваних завдань, організаційними особливостями компанії, рівнем і ступенем підготовленості аналітиків.

У зв'язку з цим сучасний підхід до інструментальних засобів аналізу не обмежується використанням якоїсь однієї технології. В даний час прийнято розрізняти наступні основні види аналітичної діяльності:

- стандартна звітність;
- нерегламентовані запити;
- аналітична обробка в режимі реального часу (OLAP);
- інтелектуальний аналіз даних (Data Mining).

Кожна з цих технологій має свої особливості, певний набір типових задач і повинна підтримуватися спеціалізованою інструментальною середовищем.

Найбільший інтерес представляють технології OLAP і Data Mining.

Аналітична обробка в режимі реального часу OLAP - технологія оперативної аналітичної обробки даних, використовує методи і засоби для збору, зберігання і аналізу багатовимірних даних з метою підтримки процесів прийняття рішень. Основне призначення OLAP-систем - підтримка аналітичної діяльності, довільних запитів користувачів.

Всі дані в багатовимірному сховище діляться на три категорії [4]:

- детальні дані - дані, що переносяться безпосередньо з OLTP підсистем. Відповідають елементарним подіям, фіксується в OLTP системах;
- агреговані (узагальнені) дані - дані, одержувані шляхом підсумовування по певним вимірам;
- метадані - дані про дані, що містяться в сховище. Дані в сховищі утворюють такі інформаційні потоки [2]:
- вхідний потік - утворюється даними, копійований з OLTP-систем, дані при цьому часто очищаються і збагачуються шляхом додавання нових атрибутів;
- потік узагальнення - утворюється агрегуванням детальних даних і їх збереженням у сховище;
- архівний потік - утворюється переміщенням детальних даних, кількість звернень до яких знизилася;
- потік метаданих - утворюється потоком інформації про дані в репозиторій даних;
- вихідний потік - утворюється даними, вилучаються користувачами;
- зворотний потік - утворюється очищеними даними, записів назад в OLTP-системи.

У процесі аналізу даних часто виникає необхідність побудови залежностей між різними параметрами, число яких може бути значним. Можливість аналізу залежностей між різними параметрами передбачає можливість подання даних у вигляді багатовимірної моделі - гіперкуба, або OLAP-куба, в осередках якого зберігаються аналізовані дані, наприклад обсяги продажів. На перетині осей вимірювань розташовуються дані, що кількісно характеризують аналізовані факти - заходи, наприклад, обсяги продажів, виражені в одиницях продукції.

Осі куба це вимірювання, за якими відкладають параметри, які стосуються аналізованої предметної області, наприклад, назви товарів і назви місяців року.



У найпростішому випадку двовимірного куба виходить таблиця, що показує значення рівнів продажів по товарах і місяцях. Подальше ускладнення моделі даних можливо з кількох напрямками [5]:

- збільшення числа вимірювань - дані про продажі не тільки по місяцях і товарам, але і по регіонах. В цьому випадку куб стає тривимірним;
- ускладнення вмісту комірки - наприклад, нас може цікавити не тільки рівень продажів, але і чистий прибуток або залишок на складі. В цьому випадку в комірці буде кілька значень;

Введення ієрархії в межах одного виміру - загальне поняття «час» пов'язане з ієрархією значень: рік складається з кварталів, квартал з місяців і т.д.

Методи інтелектуального аналізу даних (Data Mining) поділяються на три групи [6]: пошук залежностей (discovery), прогнозування (predictive modelling) і аналіз аномалій (forensic analysis). Пошук залежностей полягає в перегляді бази даних з метою автоматичного виявлення залежностей.

Проблема тут полягає у відборі дійсно важливих залежностей з величезного числа існуючих в бази даних. Прогнозування передбачає, що користувач може пред'явити системі записи з незаповненими полями і запросити відсутні значення. Система сама аналізує вміст бази і робить правдоподібне пророкування щодо цих значень. Аналіз аномалій - це процес пошуку підозрілих даних, сильно відхиляються від стійких залежностей [3].

Технології інтелектуального аналізу даних дозволяють вирішувати безліч задач із залученням методів математичної статистики і теорії ймовірності, а також методів штучного інтелекту.

Найбільшого поширення знайшли методи використання, що дозволяють вирішувати такі завдання:

- Класифікація - віднесення об'єкта (події, предмета) до одного з заздалегідь відомих класів по його характеристикам.
- Регресія - прогнозування значення будь-якого вихідного параметра об'єкта по набору вхідних параметрів.

- Кластеризація - завдання полягає в угрупованні об'єктів на кластери (Незалежних груп) по значеннями притаманних об'єктам параметрів. Рішення цього завдання допомагає краще зрозуміти дані.

Пошук асоціативних правил - виявлення закономірностей між будь-якими пов'язаними об'єктами. Вирішення цього завдання допомагає краще зрозуміти природу аналізованих даних і може служити для прогнозування появи подій.

Передбачення послідовностей - знаходження залежностей між об'єктами або подіями в формі правил, що вказують, що після якогось події А настає подія В.

Аналіз відхилень - аналіз даних на предмет входження явних нехарактерних шаблонів.

В якості основних методів Data Mining прийнято виділяти наступні [7]:

- Регресійний, дисперсійний і кореляційний аналіз.
- Нейромережеві алгоритми (вихідні параметри розглядаються як сигнали, перетворені відповідно до наявних зв'язками між вузлами мережі; результатом аналізу є відгук всій мережі на вихідні дані; структура зв'язків задається в ході спеціальної процедури, званої навчанням мережі).
- Дерева рішень (ієрархічні структури, що базуються на наборі предикатів).
- Алгоритми сегментації (застосовуються для об'єднання схожих подій в групи на підставі подібних значень декількох полів в наборі даних).
- Алгоритми обмеженого перебору, що обчислюють частоти комбінацій простих логічних подій в підгрупах даних.
- Еволюційні методи (пошук і генерація похідних алгоритмів, виражають взаємозалежність даних, на підставі спочатку заданого алгоритму, модифікує в процесі пошуку) [4].

Ефективність методів Data Mining безпосередньо пов'язана з кількістю вихідних даних. Чим більший обсяг представлена вихідна статистика, тим більше стійкі закономірності виявляються в ході аналізу. Чим в більшому обсязі представлена вихідна статистика, тим стійкіші закономірності виявляються в ході аналізу. Обидві технології можна розглядати як складові частини процесу підтримки прийняття рішень.



## РОЗДІЛ 3 Розробка інформаційно-аналітичної Web-системи оцінювання попиту на продукцію легкої промисловості

### 3.1 Проектування інформаційно-аналітичної web-системи

Відповідно до опису системи основною функцією є аналіз запитів шляхом пошуку інформації в декількох БД. Таким чином, визначається єдина робота контекстної діаграми. Далі визначаються вхідні і вихідні дані, а також механізми і управління.

Для того щоб аналізувати запити, необхідно зареєструвати його в системі, відкрити доступ до БД і обробити цей запит. В якості вхідних даних будуть використовуватися «Логін клієнта», «токен клієнта», «початкова БД», «запит контенту». Виконання запиту веде або до отримання інформації від системи, або до зміни вмісту БД (наприклад, при складанні експертних оцінок), тому вихідними даними будуть «звіти» і «змінена БД». Процес обробки запитів буде виконуватися монітором системи під контролем адміністратора (рис. 3.1).

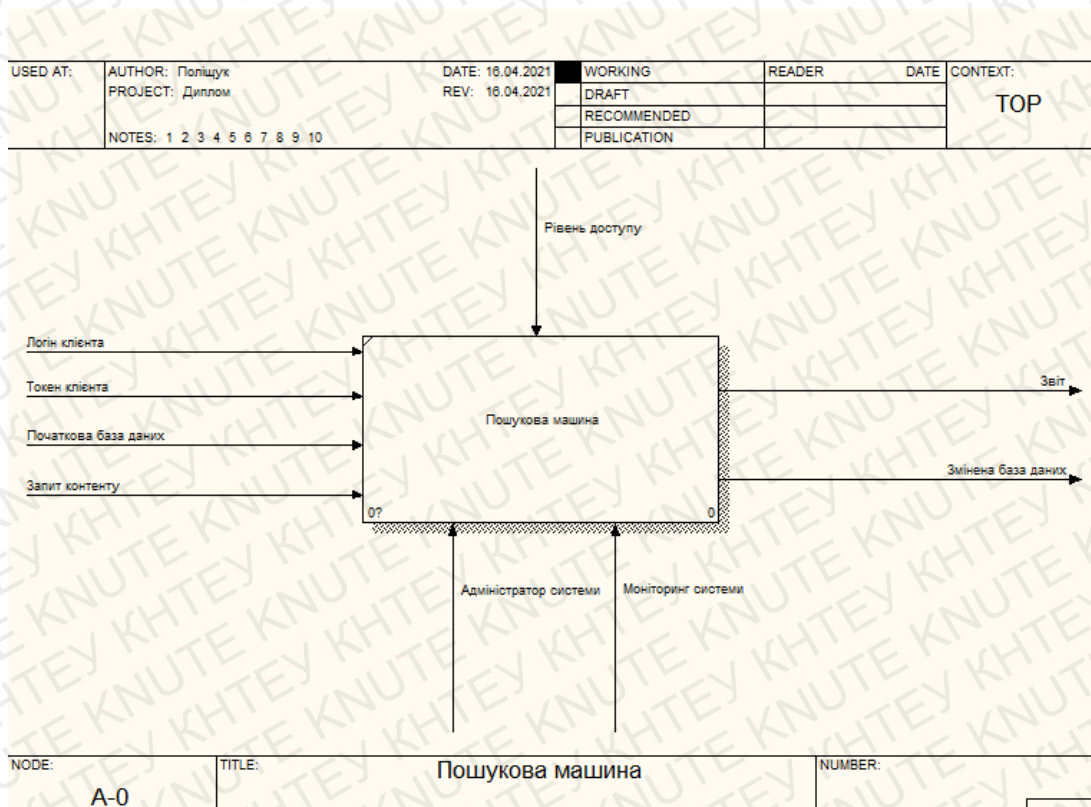


Рис. 3.1. Контекстна діаграма системи

Після проходження процедури доступу в системи, API аналізує запит клієнта, вибираючи підсистему, яка буде обробляти запит. Користувача системи не цікавлять внутрішні алгоритми її роботи. В даному випадку йому важливо, що вибір підсистеми буде проведений автоматично, без його втручання, тому декомпозиція звернення до підсистеми тільки ускладнить модель.

Декомпозована роботу «Обробка запиту клієнта», що виконується підсистемою обробки запитів, визначає категорії і повноваження користувачів. Перед здійсненням пошуку відповіді на запит необхідно відкрити БД (підключитися до неї). У загальному випадку БД може перебувати на віддаленому сервері, тому може знадобитися встановлення з'єднання з нею. Послідовність робіт:

- Відкриття БД.
- Виконання запиту.
- Генерація звітів.
- Після відкриття БД необхідно повідомити системі про встановлення з'єднання з БД, після чого виконати запит і згенерувати звіти для користувача.

Необхідно відзначити, що в «Виконання запиту» включається робота різних підсистем. Наприклад, якщо запит включає в себе тестування, то його буде виконувати підсистема професійних і психологічних тестів (рис. 3.2).



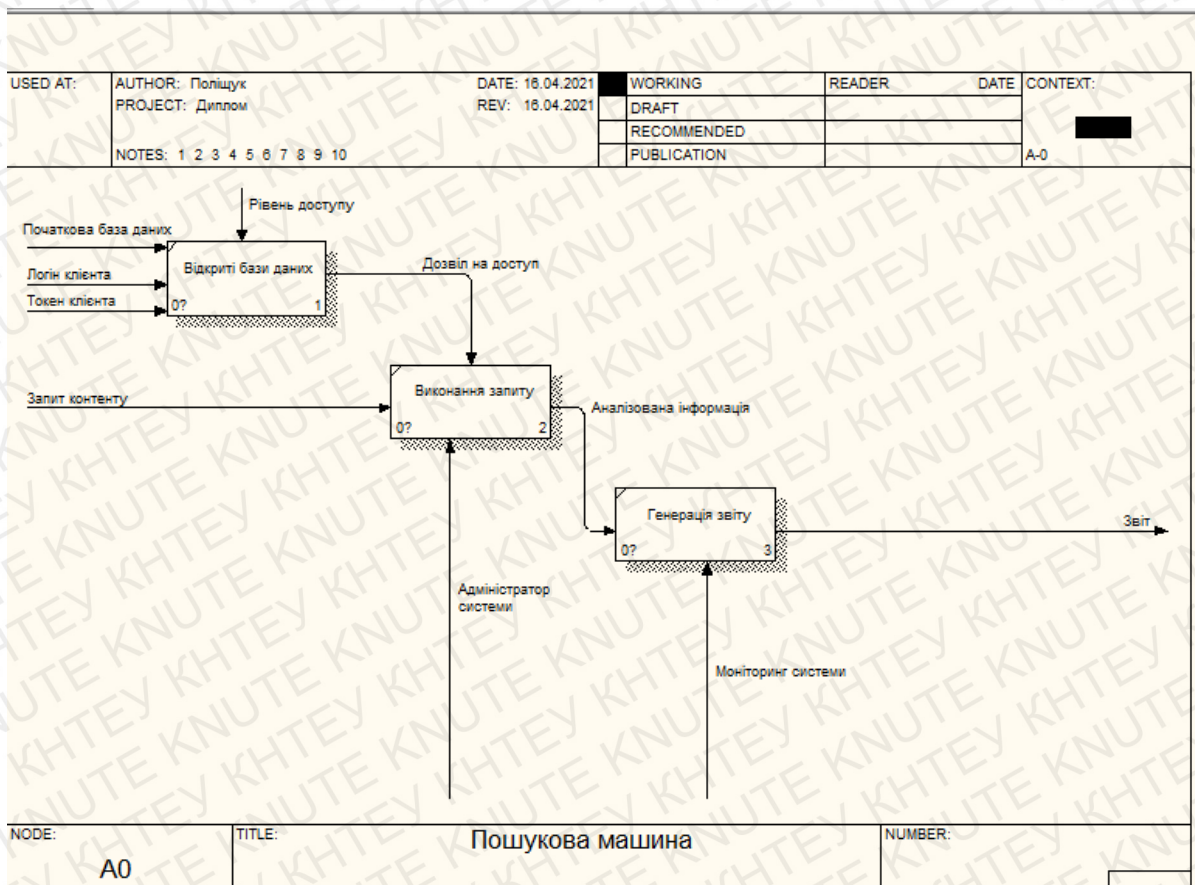


Рис. 3.2. Декомпозиція роботи «Запит клієнта»

Декомпозиції останнього блоку «Зміна БД». З точки зору клієнта, дані системи розташовуються в одній БД. Відповідно до моделі, дані що надійшли не відразу оновлюються в системі, а проходять додатковий етап обробки і контролю.

Алгоритм зміни можна сформулювати наступним чином:

- Визначається БД, в якій буде змінюватися інформація.
- Адміністратор здійснює контроль даних і вносить їх в БД.

Надавши можливість поновлення БД безпосередньо за запитом, міняючи процес контролю даних, дана система працювати коректно не буде. В цьому випадку необхідно забезпечити контроль цілісності БД для уникнення її пошкодження. У цьому випадку діаграма буде виглядати наступним чином (рис. 3.3).



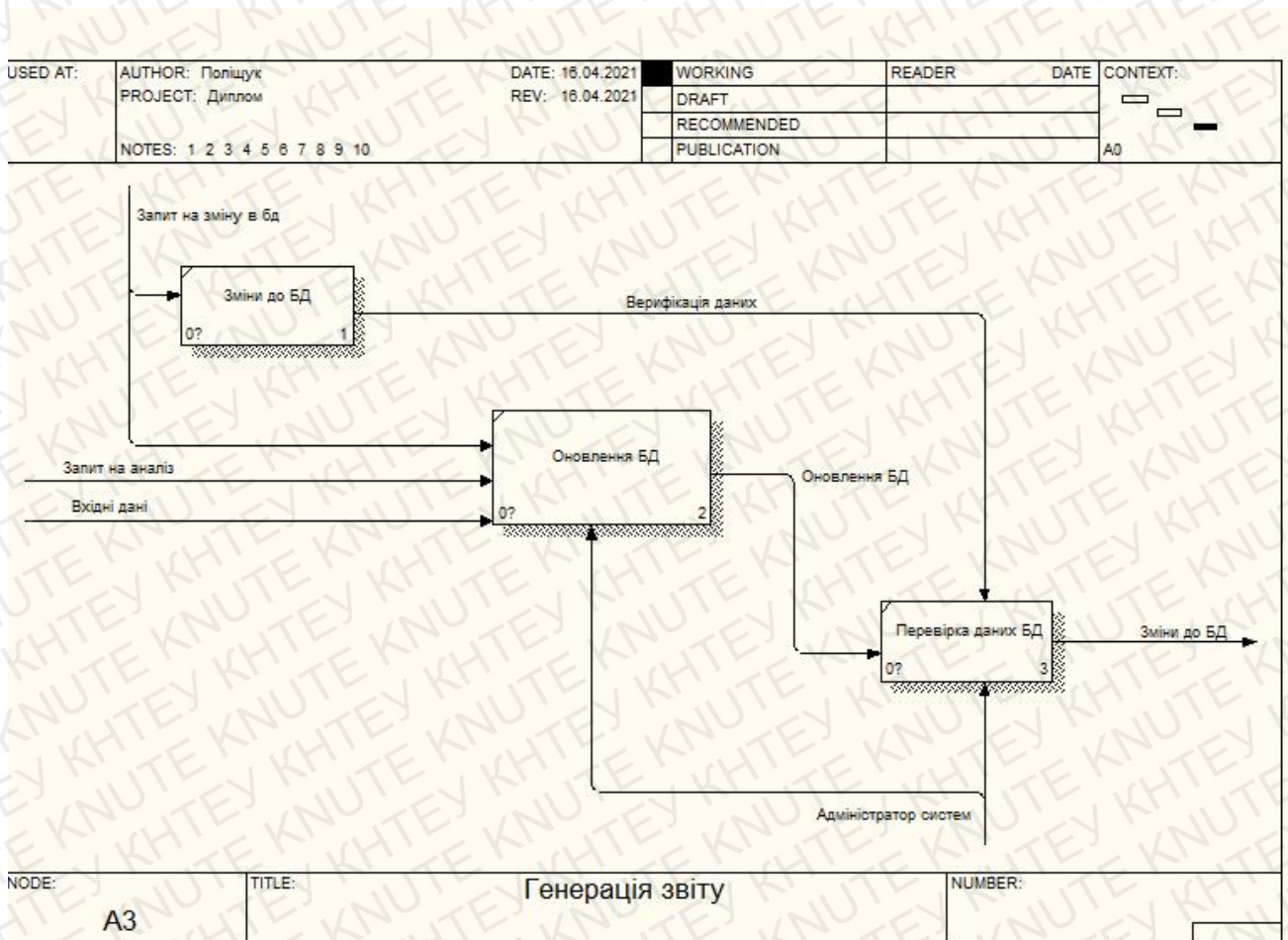


Рис. 3.3. Декомпозиція оновлення системи

### 3.2 Розробка користувацького інтерфейсу

Процес створення бази даних розпочався зі структурування інформації та розділення її на таблиці. Для того, щоб система працювала, потрібно проаналізувати джерела інформації та до аналітичної сторінки підключити всі необхідні плагіни.

Так як найпопулярнішими джерелами інформації є пошукові мережі, пошукові мережі та соціальні мережі, то логічним буде розбити блоки видачі інформації на наступні: Google Adwords, Яндекс Дірект та SEO (рис. 3.4).

Відповідно, для того, щоб дані блоки працювали, необхідно підключити API відповідних аналітичних ресурсів, що відповідають інформаційне

наповнення. А саме, до блоку Google Adwords потрібно підключити API Google Trends (дод. А)



```
1 use strict ,
2 import chai from 'chai';
3 import api from '../src/api';
4 import { getInterestResults, constructInterestObj } from '../src/utilities';
5
6 const interestHandler = {
7   processor: getInterestResults,
8   objectConstructor: constructInterestObj,
9 };
10
11 const expect = chai.expect;
12
13 describe('api', () => {
14   let count = 0;
15
16   function request(counter) {
17     return new Promise((resolve, reject) => {
18       const fakeReqObj = {
19         widgets: [{
20           request: {
21             requestOptions: {},
22           },
23           title: 'Interest over time',
24           id: 'TIMESERIES',
25           token: 'dogman',
26         }],
27       };
28     });
29   }
30 }
```

/home/uscreene/ucrate.org.ua/bogena/google-trends-api-master/google-trends-api-master/test/api.spec.js

Строка: 0    Позиция: 0

Рис. 3.4. API Google

В даному API найголовнішим файлом є google-trends-api.min.js, що відповідає за комунікацію з сервісом Google. Його перевагою є те, що, він не несе великого навантаження на сервер (рис.3.6), що є актуальним для корпоративного використання. Адже встановлений на власний сервер додаток не буде перенавантажувати сервер та дозволить використовувати сервіс на будь-якому хостингу (рис. 3.5)



```
17 /* ***** Interest over time ***** */
18
19 // googleTrends.interestOverTime({keyword: 'Valentines Day'})
20 // .then((res) => {
21 //   console.log('this is res', res);
22 // })
23 // .catch((err) => {
24 //   console.log('got the error', err);
25 //   console.log('error message', err.message);
26 //   console.log('request body', err.requestBody);
27 // });
28
29 // googleTrends.interestOverTime({
30 //   keyword: 'Valentines Day',
31 //   startTime: new Date(Date.now() - (4 * 60 * 60 * 1000)),
32 //   granularTimeResolution: true,
33 // }, function(err, results) {
34 //   if (err) console.log('oh no error!', err);
35 //   else console.log(results);
36 // });
37
38 /* ***** Interest over time - Set a custom timezone ***** */
39
40 // googleTrends.interestOverTime({
41 //   keyword: 'Valentines Day',
42 //   timezone: new Date().getTimezoneOffset() / 60,
43 // }, function(err, results) {
44 //   if (err) console.log('oh no error!', err);
45 //   else console.log(results);
46 // });
47
48 /* ***** Interest over time - Comparing multiple keywords ***** */
```

Рис. 3.5. Алгоритм пошуку інформації в API Google



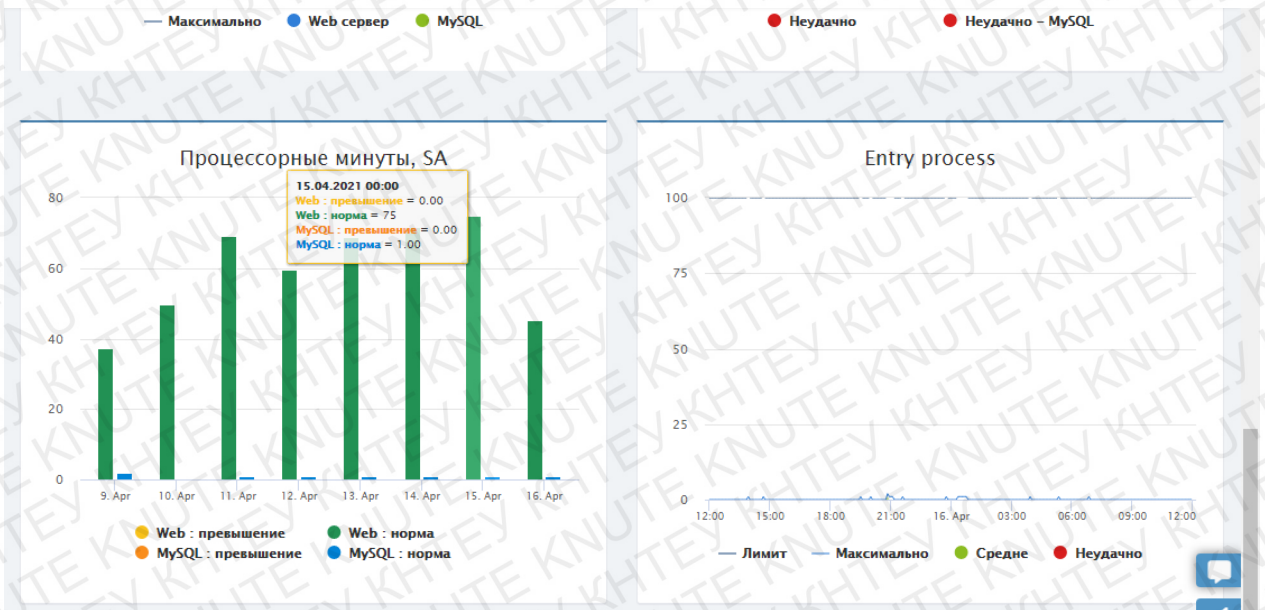


Рис. 3.6. Навантаження на сервер

Як видно на зображенні, значення max по роботі хостинга не перевищує нормальні показники та не викликає критичних помилок в роботі хостингу.

Наступним підключається Яндекс Дірект. Для його функціонування підключається API Yandex Wordstat. Зважаючи на те, що парсер Яндекс не працює без VPN на території України, сам API потребує певного доопрацювання (дод. Б) (рис. 3.7).

Так як даний проект створювався в WordPress то підключення до блоків відбувалося за рахунок згенерованих токенів, які й виконували кожену операцію окрему в корпоративному акаунті на аналізованих майданчиках (рис 3.8).

```

5 #####
6 # Вводим исходные данные для парсинга
7 #####
8
9 # Задаем URL адрес API Яндекс.Директа
10 # Адрес песочницы: url = 'https://api-sandbox.direct.yandex.ru/v4/json/'
11 # Адрес полного доступа: url = 'https://api.direct.yandex.ru/v4/json/'
12 url = 'https://api-sandbox.direct.yandex.ru/v4/json/'
13 # Указываем свой токен на доступ к API Яндекс.Директ
14 token = 'AqAAAAAX1GmLIIX9s4uEoSN1SEYjJxTsAHZ0r8w'
15 # Указываем логин своей учетной записи от Яндекс.Директ
16 userName = 'yandex.username'
17
18 # Пишем список общих минус-слов, как в примере (со знаком "-")
19 minusWords = [
20     '-купить',
21     '-пнуть',
22 ]
23
24
25 # Пишем список фраз, по которым будем парсить
26 phrases = [
27     'хліб',
28     'зерно'
29 ]
30
31 # Указываем регион, при необходимости (можно оставить пустим)
32 geo = ['uk']
33
34 #####

```

Рис. 3.7. API Yandex Wordstat

```
37
38 # Добавляем минус-слова ко всем фразам
39 data = []
40 for i in range(len(phrases)):
41     data.append(phrases[i])
42     for j in range(len(minusWords)):
43         data[i] += ' ' + minusWords[j]
44
45 # Создаем парсер
46 parser = WordstatParser(url, token, userName)
47
48 try:
49     # Запрашиваем кол-во оставшихся баллов Яндекс.Директ API
50     units = parser.getClientUnits()
51     if 'data' in units:
52         print ('>>> Баллов осталось: ', units['data'][0]['UnitsRest'])
53     else:
54         raise Exception('Не удалось получить баллы', units)
55
56 # Отправляем запрос на создание нового отчета на сервере Яндекс.Директ
57 response = parser.createReport(data, geo)
58 if 'data' in response:
59     reportID = response['data']
60     print ('>>> Создается отчет с ID = ', reportID)
61 else:
62     raise Exception('Не удалось создать отчет', response)
63
64 # Проверяем список отчетов на сервере. Должен появиться новый. Ожидаем его готовности
65 reportList = parser.getReportList()
```

Рис. 3.8. Процесс пошуку в API Yandex Wordstat

Для аналізу даних шляхом SEO необхідно підключити ресурс, який має назву `арі megaindex`. Дані, які відносяться до розділу SEO – це всі запити, які в аналітиці Google та Yandex відносяться до розділів «Refferal» та «Organic». Це всі запити, які йдуть напряму через стрічку пошуку браузера, перехід з соціальних мереж та меседжерів. Всі ці дані зберігаються в Google Аналітиці та Yandex Аналітиці.

Підключення `арі megaindex` відбувається за рахунок підключення 1 PHP файлу (рис. 3.9)



```
1 <?php
2 $array = array();
3 $array['user'] = 'megaindex-api-test@megaindex.ru';
4 $array['password'] = '123456';
5 $array['lr'] = '213';
6 $array['imp']=1;
7 $array['request'] = "Xnid";
8 $content = file_get_contents('https://api.megaindex.ru/scan_wordstat?'.http_build_query($array));
9 $json = json_decode($content);
10 if( !is_object($json) ){
11     echo 'Не удалось разобрать данные';
12     exit();
13 }
14 if($json->status!=0){
15     echo $json->err_msg;
16     exit;
17 }
18 echo $json->data . " показов в месяц. ";
19 ?>
```

Рис. 3.9. Арі megaindex

Використання цих двох ресурсів повністю замінюють ручний пошук в аналітичних системах пошукових мереж та зосереджує результати парсингу пошукових машин в одній точці. Результатом робіт є наступна аналітична сторінка (рис 3.10).

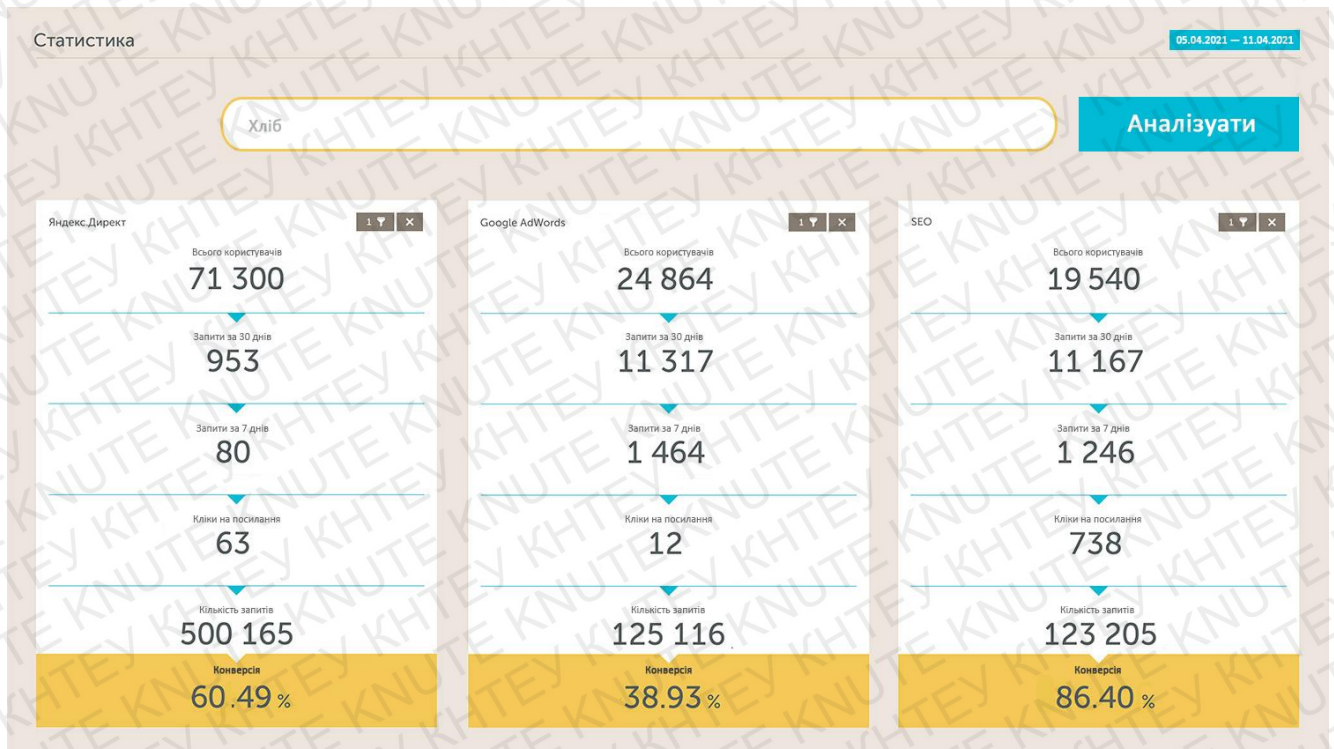


Рис. 3.10. Аналітична сторінка



### 3.3 Оцінка отриманих результатів проектування

Для коректного відображення сайту та роботи всіх функцій необхідно на пристрої мати наступне програмне забезпечення:

- Google Chrome (версія оновлення не менше 48).
- Internet Explorer (версія оновлення не менше 10).
- Mozilla Firefox (версія оновлення не менше 44).
- Adobe Flash Player.
- Вільної пам'яті на пристрої не менше 30 мб.
- Швидкість завантаження сторінки від 256 кб/с.

При використанні сайту на мобільних пристроях, функціональність сайту зберігається. Мінімальний розмір екрану для мобільного пристрою 720x480 пікселі (табл. 3.1).

Для порівняльного аналізу було використано 3 пристрої. На яких перевірено веб сайт на наявність помилок:

- коректність відображеної інформації;
- відповідність заданим параметрам;
- динамічність;
- робота PHP движка.

Таблиця 3.1

#### Відповідність критеріям перевірки

Критерії	Відповідність
Коректність відображеної інформації	Відповідає
Відповідність заданим параметрам	Відповідає
Динамічність	Відповідає

Робота PHP движка

Відповідає

## ВИСНОВКИ ТА РЕЗУЛЬТАТИ

Для вирішення поставленої в роботі мети було проведено класифікацію інформаційно-аналітичних систем, які використовуються на сучасних підприємствах. В процесі виконання випускного кваліфікаційного проекту було зроблено наступне:

1. Проведено аналіз сучасних методів розробки інформаційних систем підприємства торгівлі та визначено найактуальніші серед них.
2. Провівши аналіз технологій було визначено технології, за допомогою яких найзручніше реалізувати дану інформаційну систему.
3. Досліджені сучасні мови веб-програмування, що дало змогу порівняти структурний рівень кожної з мов, їх пристосованість до реалізації певних завдань.
4. Досліджено системи моделювання бізнес-процесів для створення імітаційної моделі аналітичної системи.
5. Розроблено імітаційну модель аналітичної веб системи.
6. В третьому розділі було налаштовано API Google Trends, API Yandex Wordstst та API Megaindex.
7. Створено інтерфейс користувача. Дружній інтерфейс створено для інтуїтивної навігації по сайту, що допомагає користувачу використовувати веб-ресурс в повному обсязі.
8. За результатами розробки отримано сайт, який можна використовувати на підприємстві торгівлі.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Спикльмайр С. Разработка Web-приложений и управление контентом для программистов / Спикльмайр С. — М.: ДМК, 2013 – С. 464
2. Горнаков, С.Г. Осваиваем популярные системы управления сайтом / Горнаков С.Г. – ДМК Пресс, 2019 – С. 336
3. Візуальне та імітаційне моделювання бізнес-процесів як найбільш ефективні методи впровадження процесно-орієнтованого підходу до управління підприємством / О. О. Андрейчіков, О. М. Гуца, О. Г. Українець // Системи обробки інформації. - 2012. - Вип. 3(1). - С. 92-95
4. Управління бізнес-процесами від моделювання до моніторингу з використанням продуктів WebSphere V6 [Текст]: Підтримка IBM Supportie / Улі Валі, Л. Лейбович, Е. Превост, і ін .; пер. з англ. А. Закис. 1-е изд. - IBM® Redbook, 2017. - 423 с.
5. IBM WebSphere Business Modeler. [Електроний ресурс]. Режим доступу [www. URL: http://www.interface.ru/home.asp?artId=16175](http://www.interface.ru/home.asp?artId=16175) – Загол. з екрана.
6. BPMI Notation Working Group. [Електроний ресурс]. Режим доступу [www. URL: http://www.bpmi.org](http://www.bpmi.org) - Загол. з екрана.
7. Управление бизнес-процессами от моделирования до мониторинга с использованием продуктов WebSphere V6 [Текст]: Поддержка IB Supportie / Ули Вали, Л. Лейбович, Э. Превост, и др.; пер
8. Сайт Інституту фізики та біофізики НАН України. – Режим доступу: <http://iarpb.kiev.ua/>. Дата доступу: 02.04.2016
9. Шаблоны Joomla!. – Режим доступу: <http://joomlaportal.ru/joomlatemplates>. – Дата доступу: 25.05.2016
10. Керівництво по PHP. [Електронний ресурс]. – Режим доступу: <https://php.net/manual/ruPHP>

11. Створення сайтів - CMS за і проти. [Електронний ресурс]. – Режим доступу: <http://orangestudio.com.ua/all/111-site-development-cms.html>
12. Опис CMS 1С-Бітрікс і оцінок. [Електронний ресурс]. – Режим доступу: <http://lemarbet.com/ua/otkrytie-internet-magazina/vybiraem-dvizhok-dlya-internet-magazina-kakaya-cms-luchshe>
13. Системи управління сайтом. [Електронний ресурс]. – Режим доступу: <https://www.bitrix.ua/products/cms/>
14. Система управління контентом CMS PrestaShop [Електронний ресурс]. – Режим доступу: [www.phpcoders.org.ua](http://www.phpcoders.org.ua) › Інтернет технології
15. Обираєм двигун для інтернет-магазину: яка CMS краще. [Електронний ресурс]. – Режим доступу: <http://lemarbet.com/otkrytie-internet-magazina/vybiraem-dvizhok-dlya-internet-magazina-kakaya-cms-luchshe>
16. Веб-сервери Інтернет — Wiki TNEU. [Електронний ресурс]. – Режим доступу: [http://wiki.tneu.edu.ua/index.php?title=1\\_Веб-сервери\\_Інтернет](http://wiki.tneu.edu.ua/index.php?title=1_Веб-сервери_Інтернет)
17. Що таке браузер Які браузери популярні на сьогоднішній день. [Електронний ресурс]. – Режим доступу: <http://vse-prosto.xn--b1ag1aeig3e.xn--p1ai/shho-take-brauzer-jaki-brauzeri-populjarni-na.html>
18. Веб-сервер (Web Server). [Електронний ресурс]. – Режим доступу: <http://hi-news.pp.ua/tehnka-tehnologyi/2337-veb-server-web-server-dlya-chogo-vn-potrben-yak-vlashtovano-yak-pracyuye.html>
19. Довідкова система. [Електронний ресурс]. – Режим доступу: <http://techtrend.com.ua/index.php?newsid=7553>
20. Огляд варіантів побудови інформаційної системи для торговельного підприємства. [Електронний ресурс]. – Режим доступу: <https://studfiles.net/preview/5485652/page:2>
21. Мови сценаріїв і програмування. [Електронний ресурс]. – Режим доступу: <http://um.co.ua/8/8-14/8-14653.html>

22.Проектування інтернету-магазину для підприємства роздрібною торгівлі. [Електронний ресурс]. – Режим доступу: <http://uawsi.com/22/220880.html>



## Код Api Google Trends

```
'use strict';
// var googleTrends = require('./lib/google-trends-api.min.js');
// googleTrends.autoComplete({keyword: 'Back to school'})
// .then((res) => {
//   console.log('this is res', res);
// })
// .catch((err) => {
//   console.log('got the error', err);
//   console.log('error message', err.message);
//   console.log('request body', err.requestBody);
// });
// googleTrends.interestOverTime({keyword: 'Valentines Day'})
// .then((res) => {
//   console.log('this is res', res);
// })
// .catch((err) => {
//   console.log('got the error', err);
//   console.log('error message', err.message);
//   console.log('request body', err.requestBody);
// });
// googleTrends.interestOverTime({
//   keyword: 'Valentines Day',
//   startTime: new Date(Date.now() - (4 * 60 * 60 * 1000)),
//   granularTimeResolution: true,
// }, function(err, results) {
//   if (err) console.log('oh no error!', err);
//   else console.log(results);
```

```

// });
// googleTrends.interestOverTime({
//   keyword: 'Valentines Day',
//   timezone: new Date().getTimezoneOffset() / 60,
// }, function(err, results) {
//   if (err) console.log('oh no error!', err);
//   else console.log(results);
// });
// googleTrends.interestOverTime({keyword: ['Valentines Day', 'Christmas Day']})
// .then((res) => {
//   console.log('this is res', res);
// })
// .catch((err) => {
//   console.log('got the error', err);
// })
// googleTrends.interestByRegion({
//   keyword: 'Donald Trump',
//   startTime: new Date('2017-02-01'),
//   endTime: new Date('2017-02-06'),
//   resolution: 'CITY',
// })
// .then((res) => {
//   console.log(res);
// })
// .catch((err) => {
//   console.log(err);
// })
// googleTrends.interestByRegion({
//   keyword: 'Donald Trump',
//   startTime: new Date('2017-02-01'),

```

```
// endTime: new Date('2017-02-06'),
// geo: 'US-CA',
// })
// .then((res) => {
//   console.log(res);
// })
// .catch((err) => {
//   console.log(err);
// })
// googleTrends.relatedQueries({keyword: 'Westminster Dog Show'})
// .then((res) => {
//   console.log(res);
// })
// .catch((err) => {
//   console.log(err);
// })
// googleTrends.relatedTopics({
//   keyword: 'Chipotle',
//   startTime: new Date('2015-01-01'),
//   endTime: new Date('2017-02-10'),
// })
// .then((res) => {
//   console.log(res);
// })
// .catch((err) => {
//   console.log(err);
// });
// googleTrends.realTimeTrends({
//   geo: 'US',
//   category: 'all',
```



```
// }, function(err, results) {  
//   if (err) console.log('oh no error!', err);  
//   else console.log(results);  
// });  
// Please note that google only keeps around T-15 days of daily trends information.  
// TrendDate designation that go too far back in the past will result in an error.  
// Note: TrendDate is optional and will default to currentDate  
// googleTrends.dailyTrends({  
//   trendDate: new Date('2019-01-10'),  
//   geo: 'US',  
// }, function(err, results) {  
//   if (err) {  
//     console.log('oh no error!', err);  
//   }else{  
//     console.log(results);  
//   }  
// });
```

## API Yandex Wordstat

```
# Підключаємо клас парсеру WordstatParser, імпортуємо необхідні модулі
from wspartner import WordstatParser

import time

# Вводимо початкові дані для парсинга
# Задаємо URL адреса API Яндекс.Директа
# Адреса пісочниці: url = 'https://api-sandbox.direct.yandex.ru/v4/json/'
# Адреса повного доступу: url = 'https://api.direct.yandex.ru/v4/json/'
url = 'https://api-sandbox.direct.yandex.ru/v4/json/'

# Вказуємо свій токен на доступ до API Яндекс.Директ
token = 'AgAAAAAX1GmLlIX9s4uEoSniSEyjjxTsAHZ0p8w'

# Вказуємо логін свого облікового запису від Яндекс.Директ
userName = 'yandex.username'

# Пишемо список загальних мінус-слів, як в прикладі (зі знаком "-")
minusWords = [
    '-Купити',
    '-Дешево',
    '-завантажити',
    '-Безкоштовно'
]

# Пишемо список фраз, за якими будемо аналізувати довільні
phrases = [
    'Фотошоп',
    'Photoshop'
]
```

```

# Вказуємо регіон, при необхідності (можна залишити порожнім)
geo = []

# Код скрипта парсинга
# Додаємо мінус-слова до всіх фразам
data = []

for i in range (len (phrases)):
    data.append (phrases [i])
    for j in range (len (minusWords)):
        data [i] += " + minusWords [j]

# Створюємо парсер
parser = WordstatParser (url, token, userName)
try:
    # Запитуваний кількість залишилися балів Яндекс.Директ API
    units = parser.getClientUnits ()
    if 'data' in units:
        print ( '>>> Балів залишилося:', units [ 'data' ] [0] [ 'UnitsRest' ])
    else:
        raise Exception ( 'Не вдалося отримати бали', units)

# Відправляємо запит на створення нового звіту на сервері Яндекс.Директ
response = parser.createReport (data, geo)
if 'data' in response:
    reportID = response [ 'data' ]
    print ( '>>> Проект звіт з ID =', reportID)
else:
    raise Exception ( 'Не вдалося створити звіт', response)

```



# Перевіряємо список звітів на сервері. Повинен з'явитися новий. Очікуємо його готовності

```
reportList = parser.getReportList ()
```

```
if 'data' in reportList:
```

```
    lastReport = reportList [ 'data' ] [len (reportList [ 'data']) - 1]
```

```
    i = 0
```

```
    while lastReport [ 'StatusReport']! = 'Done':
```

```
        print ( '>>> Підготовка звіту, чекайте ... (' + str (i) + ')')
```

```
        time.sleep (2)
```

```
        reportList = parser.getReportList ()
```

```
        lastReport = reportList [ 'data' ] [len (reportList [ 'data']) - 1]
```

```
        i += 1
```

```
    print ( '>>> Звіт ID =', lastReport [ 'ReportID'], 'отримано!')
```

```
else:
```

```
    raise Exception ( 'Не вдалося прочитати список звітів', reportList)
```

```
# Читаємо звіт
```

```
report = parser.readReport (reportID)
```

```
if 'data' in report:
```

```
    # Зберігаємо результати парсинга в файли (окремо фрази, окремо частотності).
```

```
    # Якщо rightCol == True, буде зберігатися права колонка Яндекс.Вордстат (на додаток до лівої)
```

```
    parser.saveReportToTxt (report, True)
```

```
    print ( '>>> Результати парсінга успішно збережені в файли!')
```

```
else:
```

```
    raise Exception ( 'Не вдалося прочитати звіт', report)
```

```
# Видаляємо на сервері Яндекс.Директ новий звіт (він більше не потрібен)
report = parser.deleteReport (reportID)

if 'data' in report:
    print ( '>>> Звіт з ID =', reportID, 'успішно видалено з сервера
Яндекс.Директ')
else:
    raise Exception ( 'Не вдалося видалити звіт', report)

print ( '>>> Все готово!')

# Все готово! Шукаємо файли "phrases_left.txt" і "shows_left.txt" з результати
парсинга в директорії цього скрипта
except Exception as e:
    print ( '>>> зловити виняток:', e)
```