

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерних наук

СИДОРЕНКО Олександр Віталійович

**Програмний модуль підтримки просування
інформаційного порталу по пошукових
системах/ Software for support the information
portal promotion in search systems**

напрямок підготовки: 6.050103 - Програмна інженерія
фахове спрямування - Програмне забезпечення систем

Бакалаврська дипломна робота

Виконав студент групи ПЗС-41
О. В. Сидоренко

Науковий керівник:
викладач ЛИСЕНКО О.О.

Бакалаврську дипломну роботу
допущено до захисту:

" ___ " _____ 20__ р.

Завідувач кафедри
_____ **А. В. Пукас**

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПІДТРИМКИ ПРОСУВАННЯ ІНФОРМАЦІЙНОГО ПОРТАЛУ ПО ПОШУКОВИХ СИСТЕМАХ	9
1.1. Оптимізація та просування сайтів.....	9
1.2. Методи оптимізації	11
1.3. Опис предметної області	37
1.4. Специфікація вимог до системи.....	44
Висновки до розділу 1	50
РОЗДІЛ 2 ПРОЕКТУВАННЯ СИСТЕМИ ПІДТРИМКИ ПРОСУВАННЯ ІНФОРМАЦІЙНОГО ПОРТАЛУ ПО ПОШУКОВИХ СИСТЕМАХ	51
2.1. Розроблення архітектури програмної системи.....	51
2.2. Проектування структури бази даних	58
Висновки до розділу 2	63
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПІДТРИМКИ ПРОСУВАННЯ ІНФОРМАЦІЙНОГО ПОРТАЛУ ПО ПОШУКОВИХ СИСТЕМАХ	64
3.1. Програмна реалізація системи	64
3.2. Програмна реалізація бази даних.....	67
Висновки до розділу 3	69
РОЗДІЛ 4 ТЕСТУВАННЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ.....	70
4.1. Тестування.....	70
4.2. Розгортання програмного продукту.....	77
4.3. Інструкція користувача.....	79
Висновки до розділу 4	81
ВИСНОВКИ	82
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	83
ДОДАТОК А ЛІСТИНГ ОСНОВНИХ МОДУЛІВ СИСТЕМИ.....	85

ВСТУП

Останніми роками спостерігається невпинне зростання кількості веб-ресурсів у межах глобальної мережі Інтернет, що, з одного боку, сприяє доступності інформації, а з іншого – посилює конкуренцію між сайтами. Результатом цієї ситуації є поступова втрата відвідуваності та занепад окремих інтернет-джерел. Описана ситуація виникає доволі часто і не лише через об'єктивні чинники: відсутність оновлень, застарілу інформацію чи повільний хостинг, а як наслідок втрати «цікавості» з боку пошукових систем, що сприяє розташуванню ресурсу на «віддалених» сторінках пошукового запиту [1, 2].

Позиція інтернет-ресурсу в результаті пошукового запиту є дуже важливим фактором, що на пряму впливає на кількість відвідувачів. При цьому пошукова оптимізація сприяє його відображенню у верхніх позиціях рейтингу пошукової системи. Досягнення зазначеної цілі можливе лише у випадку досконалого знання алгоритмів ранжування та вмілого застосування технологій оптимізації пошукових механізмів, що є неможливим без детального огляду проблематики цієї галузі, аналізу основних методів роботи та основних алгоритмів оцінювання.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПІДТРИМКИ ПРОСУВАННЯ ІНФОРМАЦІЙНОГО ПОРТАЛУ ПО ПОШУКОВИХ СИСТЕМАХ

1.1. Оптимізація та просування сайтів

Пошукове просування або пошукова оптимізація SEO (Search Engine Optimization) - це комплекс заходів для підвищення позицій сайту в результатах видачі пошукових систем за певними запитами користувачів. Зазвичай, чим вище позиція сайту в результатах пошуку, тим більше зацікавлених відвідувачів переходить до нього з пошукових систем.

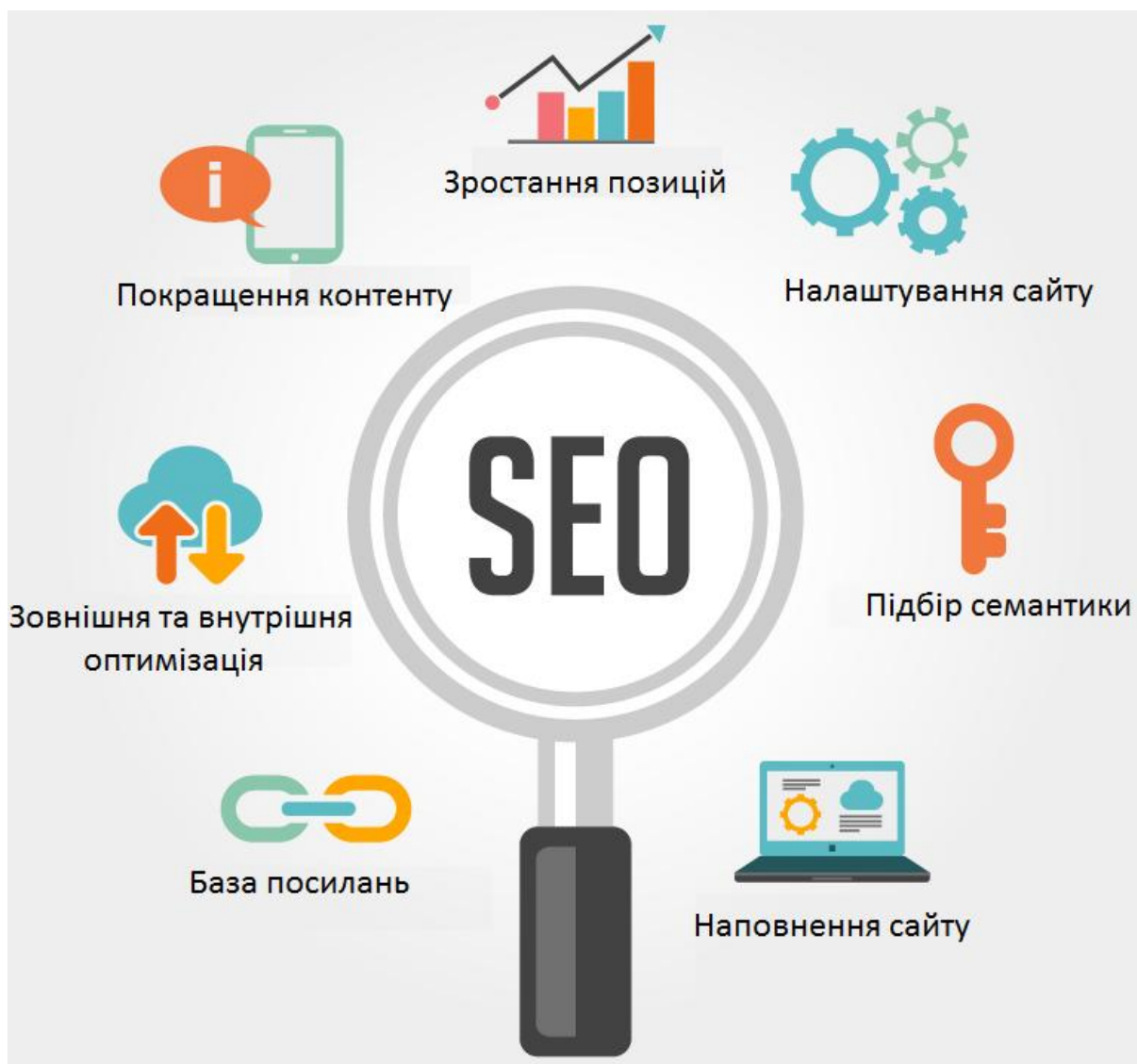


Рис. 1.1. Search Engine Optimization

Тут використовуються спеціальні заходи, які в основному, пов'язані із зміною змісту сайту і посилань, щоб вони відповідали можливим запитам, з якими звертаються до пошукових машин відвідувачі. Ідеалом пошукового просування є виведення сайту на першу (в крайньому випадку — другу) сторінку пошукової системи.

Основні параметри для оптимізації

При обчисленні ступені відповідності сайту до введеного запиту (релевантності) пошукова система враховує наступні параметри сайту:

- Щільність ключових слів. Складні алгоритми сучасних пошукових систем дозволяють застосовувати семантичний аналіз тексту, для відсіювання пошукового спаму, коли ключове слово зустрічається дуже часто.
- Індекс цитування сайту. Він залежить від кількості та авторитетності сайтів, що посилаються на даний сайт. Багато пошукових систем не враховують взаємні посилання між сайтами. Важливо, щоб посилання були з сайтів схожої тематики.

Чинники, що впливають на позицію сайту у видачі пошукової системи:

- Внутрішні, які є в підпорядкуванні власника сайту — приведення тексту і розмітки сторінок у відповідність до вибраних запитів, покращення якості контенту на сайті, стилістичне оформлення тексту (заголовки, жирний шрифт), зручна структура і навігація, використання внутрішніх посилань тощо.
- Зовнішні, які залежать від популяризації сайту в Інтернеті — обмін посиланнями, реєстрація в каталогах та інші заходи для підвищення посилань на ресурс.

Фахівця, який здійснює заходи з оптимізації сайтів, називають оптимізатором.

Одночасно з появою та розвитком пошукових систем у середині 90-х років з'являється і оптимізація сайтів. В той час пошуковики надавали велике значення тексту на сторінці, ключовим словам в мета-тегах та іншим внутрішнім чинникам, якими власники сайтів могли легко маніпулювати. Це призвело до того, що в результатах пошуку багатьох пошуковиків перші кілька

сторінок займали невідповідні сайти, що свідчило про низьку якість пошуковиків і привело багато з них до занепаду.

З появою технології PageRank більшої ваги набувають зовнішні чинники, і це допомогло Google стати лідером пошуку у світовому масштабі.

Оптимізація сайтів розвивається разом з пошуковими системами, і в сучасних результатах пошуку можна бачити все більше і більше комерційних сайтів із штучно роздутою популярністю, особливо за комерційно привабливими запитами.

1.2. Методи оптимізації

Методи оптимізації можна умовно поділити на кілька типів:

- Біла (природна) оптимізація.
- Сіра оптимізація.
- Чорна (заборонена) оптимізація.

Білою або природною оптимізацією називається підлаштування та вдосконалення коду, тексту та інших параметрів сайту під алгоритми пошукових систем з метою підвищення його позицій у результатах пошуку без застосування заборонених чи нечесних методів. Це комплекс заходів Інтернет маркетингу з підвищення відвідуваності веб-сайту, які засновано на аналізі поведінки цільових відвідувачів.

Біла оптимізація дозволяє природним чином, на підставі аналізу поведінки споживачів, спромогтися максимальної віддачі від сайту, а саме зростання цільової відвідуваності, популярності ресурсу серед користувачів Інтернет і рейтингу у пошукових системах. Природна оптимізація не містить «допінгових» методів оптимізації — розсилки спаму, фіктивного збільшення кількості банерних показів та інших заборонених методів.

Комплекс заходів

- Постійне покращення вмісту сайту, який регулярно індексують роботи пошукових систем.

- Постійне вдосконалення зручності сайту для відвідувачів — юзабіліті.
- Постійний аналіз запитів, що пов'язані з просуванням сайтом.
- Постійний пошук сайтів спорідненої тематики для створення партнерських програм.

Завдяки грамотному укладанні вмісту сайту, його підлаштування під пошукові системи, покращенню навігації сайту і постійному аналізу запитів користувачів — сайт стає більш відвідуваним, цікавим і зручним для користувача. В природній оптимізації ключову роль грає розвиток функціональності ресурсу, тобто збільшення можливостей системи і зручність використання для відвідувачів (юзабіліті).

Способи внутрішньої білої оптимізації

- Підбір і розміщення в коді сайту мета-тегів: короткого опису. Робиться це з врахуванням слів і словосполучень, за якими сайт повинен знаходитися в пошукових системах. Сторінки сайту повинні мати зрозумілу адресу - це є зручним і для людей, і для пошукових машин, які враховують тему сторінки.
- Оптимізація текстів сайту, тобто забезпечення відповідності текстів до мета-тегів. В тексті повинні зустрічатися слова, що позначені в мета-тегах, як ключові. Можна збільшити «вагу» слова в тексті за рахунок позначення його як заголовку або виділення його грубішим шрифтом. Проте, не варто забувати, що надлишок ключових слів в тексті може нашкодити. По-перше, текст буде гірше сприйматися, по-друге, пошукові системи можуть розцінити його як спам.
- Важливою частиною внутрішньої оптимізації є оптимізація файлів robots.txt і .htaccess. Це дозволяє вказати пошуковим машинам, які сторінки не потрібно індексувати, вказати правильну адресу (з www або без) тощо.

Способи зовнішньої білої оптимізації

- Додавання сайту до бази пошукових систем.

- Реєстрація сайту в авторитетних каталогах сайтів (Yandex Каталог, Ukr.net, Meta.ua).
- Розміщення прес-релізів в Інтернеті з посиланням на просувний сайт.

До сірої пошукової оптимізації можна віднести, наприклад, додавання великої кількості ключових слів в текст сторінки, часто в збиток читабельності: «масло масляне, тому що в ньому присутні масляні жири». Або автоматичне перескерування з одного веб-ресурса на інший.

Сіра оптимізація офіційно не заборонена, але її використання може бути розцінено як неприродне завищення популярності сайту. Деякі пошукові системи, наприклад, Google можуть тимчасово або назавжди заблокувати такий сайт, хоча остаточне рішення про законність методів просування приймає фахівець, а не автоматика.

Чорна оптимізація — це застосування заборонених і недобросовісних методів для підлаштування коду, тексту чи інших параметрів сайту під алгоритми пошукових систем з метою підняття його позицій у видачі результатів.

Дорвей (Doorway) (рисунок 1.2)

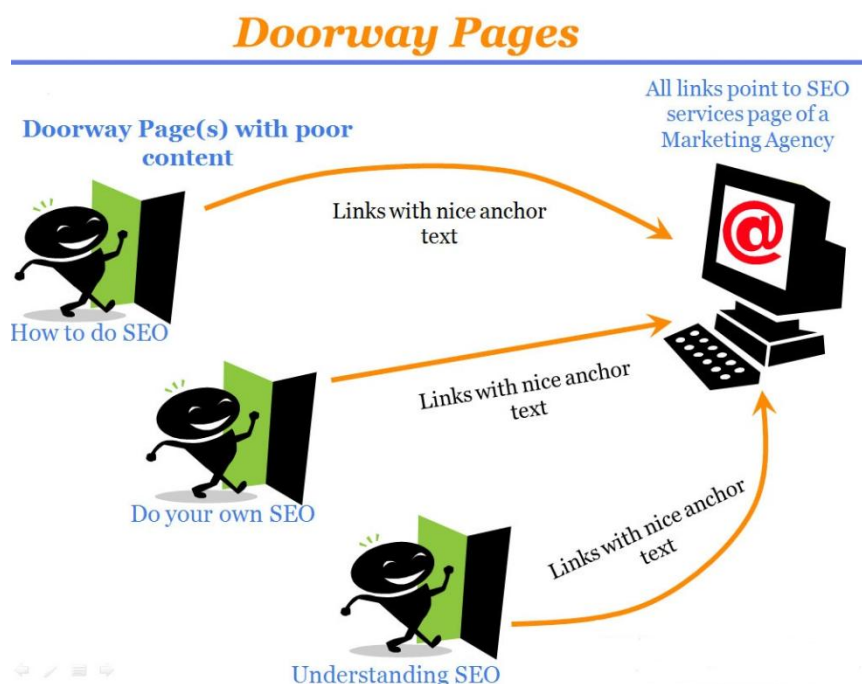


Рис. 1.2. Doorway сторінки

До чорної оптимізації відноситься використання дорвеїв — сторінок, які створено (часто автоматично) спеціально для роботів пошукових систем. В теорії людина не повинна читати текст, який розміщено на дорвеї, оскільки він є безглуздим набором ключових слів, що використовуються в можливих пошукових запитах. Як правило, дорвеї розраховані на залучення з пошукових систем відвідувачів, що використовують певні пошукові запити.

Такий підхід надає можливість попадання дорвею в результати пошуку за конкурентними запитами, але з іншого боку не дозволяє дорвею існувати довгий час без санкцій з боку пошукової системи. Подібні методи популяризації — це пряме порушення правил пошукових систем та безладний спам по чужих ресурсах (гостьових книгах, форумах, каталогах, блогах, вікі-сайтах тощо), де може залишитися посилання на дорвей.

Клоакінг (рисунок 1.3).



Рис. 1.3. Клоакінг

Відвідувачу надається одна сторінка, читабельна, а пошуковому роботу — інша, яку оптимізовано під певні запити. Схожим способом є негайна заміна тексту на сторінці відразу ж після її індексації пошуковою системою.

Інші способи.

Використання прихованого тексту, зазвичай в колір фону сторінки. Користувачі такого тексту не бачать, проте, пошукові роботи його індексують. В прихованому тексті містяться ключові слова для додавання вагомості сторінки, що оптимізується.

Однопиксельні посилання. Використання графічних зображень-посилань розміром 1*1 піксел, які відвідувач не бачить, але в тегах до картинки прописані ключові слова. Це також сприймається пошуковими системами як спроба обману і може привести до блокування сайту.

SEO-текст (рисунк 1.4).

Пошукові системи в різні способи враховують застосовані до сайту методи оптимізації. Одні надають перевагу внутрішнім чинникам (наприклад, Google), інші схильні більше брати до уваги зовнішні чинники, покладаючись на вагомість сайту серед інших ресурсів Інтернету (наприклад, Яндекс).



Рис. 1.4. Клоакінг

Проте, самими суттєвими є три чинники:

- Текст.
- HTML-код.

- Рівень авторитетності сайту, що виражається в кількості та вагомості зовнішніх посилань.

Розміщений на сайті текст є базовим чинником, що відповідає за пошукові позиції сайту.

SEO-текст — це текст, який з певною щільністю збагачено ключовими словами з семантичного ядра пошукових запитів сайту. Збагачення проводиться не довільним чином, а згідно строгих правил, які постійно вдосконалюють оптимізатори. SEO-текст є першою необхідною умовою для успішного просування сайту, його створюють спеціально для підвищення оптимізаційних характеристик ресурсу

Наповнення сайту SEO-текстом є єдиною гарантованою можливістю отримання високого рейтингу у внутрішньому ранжируванні пошукових машин. Для ретельно складеного SEO-тексту решта оптимізаційних методів є лише допоміжними засобами.

Семантичним ядром називають сукупність ключових пошукових запитів, за якими доцільно проводити пошукову оптимізацію, щоб привернути на сайт цільову аудиторію.

Принцип контентної оптимізації свідчить: любий текст, що опубліковано на сайті, повинен бути оптимізованим під пошукові системи.

Доцільний текст для сайту

Завдання пошукової системи полягає в структуризації інформаційного простору, який постійно розширюється і стає менш впорядкованим. Основною складовою інформаційного простору є текст. Тому, структуризація Інтернету — це структуризація текстів. Пошукової системи впорядковують наявні в Інтернеті тексти за релевантністю запитів користувачів.

Доцільний текст — це текст, який шукає користувач пошукової системи і який в ідеалі повинна надати пошукова система.

Пишемо для свого відвідувача

Щоб правильно створювати тексти для цільових відвідувачів, необхідно знати їх інтереси, враховувати їх культурний рівень, запити і обмеження.

Інформація повинна бути відфільтрованою і перетвореною у зрозумілу для відвідувача форму.

Не забувайте, що ви пишете для звичайних людей, тому слід уникати використання вузько професійних або технічних термінів.

Пишіть звичною мовою

Прагніть викладати на сайті прості і короткі тексти. Пишіть так, щоб тексти відповідали рівню ваших користувачів і забезпечувалася максимальна зручність читання текстів.

Не пишіть зайвого. Зайві слова примушують людей докладати більше зусиль для пошуку потрібної інформації, а високі вирази, складні чи задовгі слова відлякують відвідувачів.

Уникайте дуже активної реклами

Не варто нав'язувати користувачам товари або послуги. Люди віддають перевагу фактам і уникають нав'язливого або перебільшено хорошого. Довіра в Інтернеті є вкрай важливою, і організаціям доводиться багато працювати, щоб її завоювати та зберегти.

Нав'язливі хвалебні заяви здаються відвідувачам саморекламою, тому часто їх ігнорують. Висловлюйте лише факти і надайте людям можливість робити висновки самостійно. Хороший вміст говорить сам за себе.

Якщо організація має гідні нагороди, варто їх згадати, особливо якщо компанія є не дуже відомою. Проте, не зайвими будуть інші похвали, наприклад, нагороди за дизайн сайту (за винятком сайтів дизайнерських агентств).

Розміщуйте рекламні гасла за межами ділянок тексту з корпоративною чи іншою інформацією, в яких люди чекають фактичної інформації, наприклад, в розділі «Інформація для замовника».

Виділення основних акцентів і видалення зайвого

Користувачі Інтернету, зазвичай, не затримуються надовго на одній сторінці і переходять з одного розділу на інший. Замість того, щоб закидати відвідувачів сайту величезним об'ємом матеріалу, варто поділити інформацію

на кілька підрозділів. Почніть з основних речей, а згодом надайте інформацію, яка дозволяє без зусиль зрозуміти суть питання.

Залишайте тісно пов'язану за сенсом інформацію на одній сторінці, не примушуйте користувачів «стрибати» від однієї неповної частини відомостей до іншої. Якщо на сторінці міститься взаємозв'язана інформація, користувачі не матимуть нічого проти прогортання тексту.

Збагачення тексту ключовими словами .

Великий об'єм тексту на сторінці — це добре, але якщо він не містить основних ключових слів і фраз, які можуть індексуватися пошуковими машинами, то користі від такого тексту небагато, оскільки сайт не буде знайдено по відповідних запитах.

Поширеною помилкою є текст, який не пов'язаний з пропонованою продукцією або складається з загальних фраз. Інтернет переповнено сайтами, що пропонують різну продукцію, яка жодного разу не згадується в тексті.

Щоб пошукова машина вважала сайт релевантним до запиту, вона повинна знайти у ньому текст запиту. Найпростішим способом допомогти їй в цьому буде розміщення відповідних ключових слів в основному тексті сторінки, а також в тегах title і meta.

Визначити, які слова насправді використовуються при пошуку, можна за допомогою спеціальних сервісів пошукових машин - служби Wordstat.Яндекс, Adstat.Rambler, MAIL.RU, Adwords.Google.

Після визначення популярних слів в запитах відвідувачів, їх можна пристосувати до товарів та послуг і відповідно втілити в основний текст. При недостатньому дослідженні ключових слів з'являється тенденція до використання загальних виразів або кричущої реклами. В цьому випадку не виграє ані вміст, ані ранг сторінки в пошуковій машині.

Ретельно підібрані ключові слова, які розумно розміщено в основному тексті сторінки, підвищують вірогідність високого ранжирування за спорідненими запитами. Зрозуміло, що недостатньо просто наповнювати цими

словами сторінку. Текст потрібно скласти так, щоб око читача не «чіплялося» за повторення ключових слів.

Головна сторінка є дуже важливою, тому, тут слід уникати графічних написів. Якщо пошукова машина не знайде на сторінці тексту, тоді вона звернеться до тегів title і meta. Але, маючи такі обмежені дані, вона навряд чи визнає таку сторінку релевантною до запиту. Щоб виправити ситуацію, слід поміщати на кожному сторінку, що оптимізується, не менше 250 слів видимого друкарського тексту.

SEO-копірайтинг

При написанні текстів, не забувайте, що ви звертаєтеся до відвідувача. Насичення тексту ключовими словами є важливим, але якщо від цього страждає стиль, «читабельність», то увага аудиторії швидко втрачається. При дослідженні ключових слів варто уявити себе на місці відвідувача. Чого він прагне? Що шукає? Як товари або послуги йому потрібні? Чи варто їх купувати?



Рис. 1.5. SEO-копірайтинг

Описуйте товари і послуги емоційно. Опишіть, як людина зможе виглядати або відчувати себе за допомогою даного товару (послуги), як покращиться його життя, скільки звільниться часу. Використовуйте слова, що знаходять у людей відгук, наприклад, «вільний», «безкоштовний», «успіх», «Ви», «гроші» і подібні.

Відвідувачі швидше знайдуть потрібне, якщо на сайті буде звичною мовою ясно висловлено основні думки. Уникайте великих фрагментів суцільного тексту. Вони легше читатимуться, якщо розбивати їх за допомогою абзаців, графіки, списків, підзаголовків.

Наводьте приклади, що ілюструють переваги вашого продукту і уникайте професійного жаргону, який може відштовхнути відвідувача. Хочете, щоб відвідувач зробив якусь дію? Напишіть чіткі інструкції, наприклад «Зробіть замовлення прямо зараз», «Підпишіться на безкоштовну розсилку» тощо. Такі звернення називають заклик до дії.

Основою тексту є ключові слова. Ви завжди повинні будувати текст, ґрунтуючись на ключових словах, а не навпаки. Якщо ви хочете, щоб сайт було знайдено за ключовими для сайту фразами, — згадайте в тексті всі ці фрази. Але, краще розподілити різні товари (послуги) до різних сторінках. Це дозволить не розпорозувати, а конкретизувати ключові слова для пошукової машини.

Кожна сторінка повинна бути сфокусована не більше ніж на двох або трьох ключових словах. Є кілька причин обмеження кількості ключових слів на сторінці. Найбільш важливі ключові слова повинні бути поміщені в `тег title` сторінки. Оскільки він повинен містити не більше шістдесяти знаків, це обмежує число ключових слів.

Якщо сторінка оптимізується для кількох ключових слів одночасно, пошукові системи не зможуть виділити з них одне як основне. Щоб зайняти високе місце в рейтингу пошукової системи, кожна сторінка сайту повинна бути сфокусована на певному ключовому слові. Якщо потрібно популяризувати сайт за кількома ключовими словами, варто збільшити число сторінок.

Високе ранжирування для одного ключового слова може допомогти рейтингу сторінки для решти релевантних ключових слів. Наприклад, якщо сторінка займає високе місце за словами «css меню», то, ймовірно, вона матиме хороші позиції за словами «вертикальне меню на CSS» або «створити своє меню».

Форматування тексту

Ретельно продумані методи форматування текстів для Інтернету можуть створити візуальні підказки, які значно покращать зручність читання текстів на сайті і допоможуть користувачам швидко переглядати інформацію та знаходити потрібні розділи.

До стандартних ефективних методів форматування відносять:

- Виділення ключових слів.
- Короткі і зрозумілі заголовки.
- Маркіровані і нумеровані списки.
- Короткі абзаци.
- Найбільш важливу фразу на сторінці розташовують у перших двох

рядках тексту.

Виділення ключових слів.

Виділення ретельно вибраних ключових слів за допомогою грубого чи колірного шрифту може привернути увагу користувачів до певних фрагментів сторінки, зробити потрібний акцент і привернути увагу до важливих елементів.

Виділення цілих абзаців або довгих фраз ускладнює читання, тому, слід виділяти лише ті слова і фрази, які позначають основні терміни і вирази. Виділення надмірно великої кількості елементів швидше призведе до негативних наслідків: сторінка виглядає переобтяженою і губиться увага читача.

Елементи дизайну часто асоціюють з оформленням гіперпосилань (наприклад, підкреслені надписи голубим кольором, грубіший шрифт), які дозволяють автоматично ідентифікувати подібні елементи на сторінці. В певних випадках такого форматування буде достатньо і тому, не потрібно нічого додатково виділяти.

Короткі і зрозумілі заголовки

Заголовок на головній сторінці сайту має особливе значення, оскільки він вже є посиланням на сторінку в результатах пошукових машин. Користувачі дуже швидко проглядають виведений перелік ресурсів у вікні пошукової

машини, тому, заголовок повинен передавати призначення сайту в кількох словах.

Дослідження доводять, що користувачі часто читають лише перші слова в наданих списках пошукових систем, тому, абсолютно неправильно використовувати ключові слова, які описують призначення вашого сайту, в кінці заголовка.

На конкретній сторінці ефективні заголовки та підзаголовки можуть привернути увагу користувачів і анонсувати подальший вміст. Вони служать як покажчики структури тексту, розбиваючи його на зручні розділи і забезпечують комфортність читання та розуміння викладеного матеріалу.

Основний сенс заголовків — вказати читачам на відомості, що їх цікавлять, тому, заголовки повинні бути короткими (максимум 60 символів) і зрозумілими.

Маркіровані і нумеровані списки

Для опису послідовності дій або переліку певних елементів використовуйте маркіровані і нумеровані списки. Маркірований список застосовують, коли перелічені елементи можна розташовувати в будь-якому порядку, а нумерований є зручним для опису покрокових інструкцій.

При використанні вертикальних списків з'являється більше вільного простору біля кожного з елементів списку, що дозволяє помічати кожен з елементів. Такі списки є більш ефективними, ніж перелік елементів, що розміщено в один рядок.

Поради щодо створення списків:

1. Використовуйте вертикальні списки за наявності чотирьох або більше елементів у списку.
2. Перед списком додайте пояснення з загальним описом елементів списку. Воно не повинне бути таким же об'ємним, як опис окремих елементів.
3. Кожен пункт списку повинен містити в собі відповідь на питання, що поставлене у відповідному реченні до цього списку.

4. Використовуйте помітні ідентифікатори для виділення елементів списку.
5. Використовуйте схожу будову фраз для кожного пункту списку.
6. Не застосовуйте списки дуже часто. Інакше вони втрачають свою ефективність.

Короткі абзаци.

Ще одним методом покращення зручності є розділення матеріалу сторінки на невеликі частини. Короткі абзаци, навколо яких залишається вільний простір, виглядають привабливіше, ніж суцільний блок тексту. Інформацію, яку поділено на невеликі фрагменти, набагато простіше сприймати.

Не існує певних правил відносно того, наскільки великими можуть бути абзаци. Бажано обмежувати абзац п'ятьма реченнями.

В кожному абзаці повинно бути одне головне речення, що присвячене одній темі. Більша кількість тем в одному абзаці збільшує його складність і ускладнює пошук ключової думки. Відвідувач повинен мати можливість прочитати головне речення і вирішити, чи буде він далі читати абзац.

Інтернет реклама

Інтернет реклама на сьогоднішній день є важливим чинником у популяризації та розвитку сайту і має ряд переваг над традиційними видами реклами в ЗМІ.

- Інтернет реклама працює відразу у двох напрямках — залучення відвідувача на сайт і одночасне просування іміджу сайту чи компанії.
- Виготовлення та демонстрація презентації сайту, компанії, продукту чи послуги в короткі терміни.
- Широкі можливості для проведення ефективних рекламних кампаній і залучення потенційних клієнтів.

Рекламну кампанію в Інтернеті, можна провести самостійно, але краще скористатися послугами професіоналів - як рекламних агентств повного циклу, так і компаній, що спеціалізуються на рекламі в Інтернет.

В Інтернеті можна застосувати наступні види реклами:

Банерна реклама (рисунок 1.6). Розміщення банерів на різних ресурсах, не обов'язково тематичних. Банерна реклама є ефективнішою, якщо вона здійснюється в масштабі цілої рекламної кампанії. Тоді, складові бренду компанії: логотип, фірмові кольори чи яскраві слогани будуть підсвідомо запам'ятовуватися і викликати в подальшому зацікавленість відвідувачів.

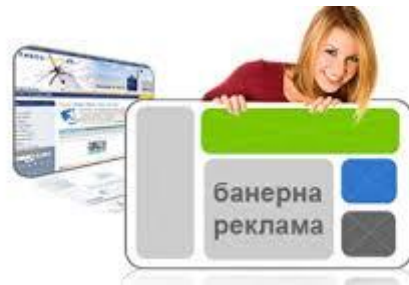


Рис. 1.6. Банерна реклама

Контекстна реклама (рисунок 1.7). Розміщення рекламної інформації на відповідних тематичних ресурсах: сторінках результатів пошуку, інформаційних порталах чи великих каталогах. Це платна послуга, що пропонують пошукові системи чи рекламні компанії і яка надає практично миттєвий результат. Контекстна реклама є особливо результативною, коли вона прив'язується до сторінки пошуку пошукової машини і висвічується при конкретних пошукових запитах. Так відвідувач отримує інформацію не лише про знайдені сайти, але і про ті ресурси, які просуваються за допомогою контекстної реклами. Цей вид реклами є самим ефективним, хоча й найдорожчим.



Рис. 1.7. Контекстна реклама

Поштові розсилки. Потенційний клієнт підписується на тематичну розсилку і в міру накопичення новин отримує необхідну рекламну інформацію. Така реклама є особливо дієвою, якщо будуть розсилатися якісні матеріали, в яких зацікавлений дописувач.

Спам. Заборонений вид реклами, який полягає в розсилці оголошень електронною поштою всупереч бажанню власників поштових скриньок. Традиційно, такий спосіб реклами вважається сумнівним з моральної точки зору, оскільки нав'язує користувачам електронної пошти інформацію, в якій вони, скоріше, не зацікавлені.

Банерна реклама

Одним з найпоширеніших видів реклами в мережі Інтернет є банерна реклама. Банер – це привабливий графічний блок, який може бути як статичним зображенням, так і анімованим міні-фільмом, який розповідає про послугу чи товар в яскравій формі. Зазвичай, банери є невід'ємною складовою фірмового стилю та іміджу компанії.

Банер є доступним 24 години на добу, 7 днів на тиждень і може охопити значну аудиторію. Один банер можна розмістити на тисячах сайтів, які щодня відвідують десятки тисяч людей.

Якісно виготовлені і розміщені на вдало підібраних тематичних ресурсах банери обов'язково привертають до себе увагу. Влучний слоган чи цікавий графічний зміст підсвідомо збуджує цікавість користувача відвідати пропонований сайт.

Ефективність банера визначається відношенням кількості показів банера до кількості «кліків» або натиснень на цей банер. Відсотковий вираз такого співвідношення позначається аббревіатурою CTR (Click through ratio). Чим вище відсоток CTR, тим ефективніше вважається банер, і тим більше відвідувачів натиснуло на ньому. Банери з низьким CTR є нерентабельними, їх необхідно або замінити або розмістити на інших, більш наближених до цільової аудиторії ресурсах.

Банери можна розміщувати:

- На окремо вибраних сайтах з чітким позиціонуванням до тематики.
- В банерній мережі, що охоплює велику кількість сайтів, які є учасниками з обміну банерною рекламою. На цих сайтах, у відведених місцях сторінки щораз будуть показуватися банери від інших учасників.

Цільові різновиди банерів:

- Цільовий банер. Його завданням є залучення на сайт потенційних клієнтів, які зацікавлені в інформації, яку надає даний Інтернет-ресурс або в запропонованих на сайті товарах чи послугах.
- Інформаційний банер. Завданням є збудити інтерес користувача і викликати бажання дізнатися більше про рекламований продукт.
- Брендний банер. Метою є збільшення індексу популярності і впізнання торгової марки рекламодавця серед користувачів Інтернету.
- Внутрішній банер. Метою є привертання уваги відвідувачів до певної новини на сайті, або повідомлення про нові послуги чи товари.

Зазвичай, на банерах вказують:

- Рекламні слогани з відповідними зображеннями.
- Оголошення про знижки, розпродаж, сезонні акції.
- Адресу чи телефон компанії.

Типи банерів

Текстові банери

Текстові банери є рекламним текстовим блоком без використання фотографій, малюнків або анімованих об'єктів. Завантажуються практично миттєво і мають найменший дратівливий чинник в порівнянні з іншими видами банерів. Цей вид банерів широко використовується компанією Google у контекстній рекламі.

Графічні банери

Графічні банери є красивими і зручними, але з обмеженими графічними ефектами.

- Јрег-банер. Статичний банер з нерухомим зображенням.

- Gif-банер. Може бути статичним або анімаційним, як правило на кілька кадрів.

Flash-банери

Flash-банери на сьогоднішній день — це один з найефективніших способів подачі рекламного повідомлення. Вони не перевантажують сайт зайвою інформацією, в них можна використовувати різноманітні ефекти: як анімаційні, так і звукові. Звичайно, використовувати ефекти слід помірно, щоб вони не ставали дратівливим чинником.

- Класичні flash-банери різних розмірів. В українських банерних мережах популярними розмірами є 468x60, 240x350, 730x90, 180x300 пікселів. Як правило, такий банер є зацикленним анімаційним роликом, тобто сюжет повторюється «по колу».

- Інтерактивні банери. Один з найефективніших видів банерів. Спонукає користувача до певної дії при наведенні мишки, тому, має високий коефіцієнт CTR. Його виконують у вигляді гри або невеликого мультфільму за участю користувача.

- Rich-media або pop-up банери розміщуються поверх сторінки, можуть бути довільної форми, обов'язково є кнопка для закриття такого банера.

- Scroll Skyscraper є pop-up банером, але при гортанні сторінки залишається на тому ж самому місці. Примушує відвідувача певний час спостерігати банер, який закриває відомості на сторінці. Закривається або відповідною кнопкою або лише при натисненні на ньому

- Screen-Glade банери. При наведенні вказівника мишки на банер розгортається додатковий рекламний простір. Наприклад, у звичайному стані є звичайним банером 468*60 пікселів, після наведення вказівника збільшується до 468*120, а після забирання вказівника з банера, повертається до первинних розмірів.

- Peel-away банер виконано у вигляді відігнутого кутика сторінки. Після натиснення на ньому користувач може побачити розгорнутий повноекранний ролик.

Java-банери

Java банери є звичайними аплетами Java, які прописано в код сторінки. Вони виконують нескладні функції: проста маніпуляція об'єктами, редагування окремих елементів, чутливість до поведінки користувачеві – натиснення мишки чи клавіші на клавіатурі. Зазвичай, Java банери є на ігрову та розважальну тематику, що підвищує інтерес Інтернет аудиторії до мережної реклами.

Коректне відображення Java банерів підтримує плагін, що втілено у всі сучасні браузері. Проте, їх завантаження відбувається дещо повільно, тому, іноді користувачі попросту вимикають опцію підтримки Java аплетів в браузері.

Ігрові банери

Ігровий банер (casual banner, interactive banner ads) - це невелика flash-гра, яка розташована на сторінці і по закінченні якої відбувається перехід за рекламним посиланням. Тобто, це нагорода користувачу за отримане задоволення від гри та піднесений настрій. Саме тому, графіка та ідеї повинні бути дотепними і зрозумілими.

Відео-банери та відео-помічники на сайті

Це нова ера в розвитку технологій Інтернет реклами. Тепер кожного відвідувача сайту буде зустрічати особистий гід по сайту, який якнайкраще виконає свою роль помічника та провідника по сайту.

Відео помічником може бути образ як живої людини так і намальований персонаж, який буде вітати кожного відвідувача сайту, розповідати йому про компанію, проводити екскурсії по сайту, просто і дохідливо розповідати про товари в Інтернет магазині. Відео помічник може з'являтися автоматично при заході на сторінку сайту або за бажанням відвідувача, подібно до програм пакету MS Office.

Компанії витрачають сотні і тисячі доларів на залучення трафіку на свої сайти, але цього у теперішній час є не достатнім. Відвідувачів потрібно

привернути на сайт, потрібно захопити їх увагу і донести до них своє повідомлення, тому, втілення відео-помічника має беззаперечні переваги:

- Різко відрізняється і виділяється серед існуючих банерів і миттєво захоплює увагу відвідувача.
- Спілкується з кожним відвідувачем на персональному рівні і привертає увагу на важливі деталі сайту.
- Є ідеальним для проведення екскурсій по сайту, промоакцій, оголошень та вітань.
- Сприяє поширенню інформації про сайт і збільшує число повторних заходів відвідувачів на сайт.

Підходи до збільшення ефективності банерних рекламних повідомлень

- Ключовим моментом ефективності є вибір сайту для розміщення банера. Цільова аудиторія сайту повинна складатися з потенційних клієнтів — люди схильні помічати відповідну рекламу там, де вони чекають її побачити.
- Велике значення в сприйнятті банера має його оформлення. Банер повинен бути простим і ненав'язливим, з плавними переходами. Колірна гамма повинна бути максимально контрастною, але в межах гармонії кольорів. Переобтяження анімацією, великою кількістю кольорів, і динамікою створює своєрідний «інформаційний шум». Золотим правилом всіх рекламних розробок є «чим простіше, тим краще», оскільки просте запам'ятовується швидше і краще, ніж складне і заплутане.
- Не можна перевантажувати банер зайвою інформацією. Найефективніший варіант — кілька слів, що виділені крупним шрифтом, в яких міститься достатньо інформації про пропонований продукт, послугу або тематичний матеріал. Бажано, щоб сенс використаної графіки і тексту не дублювався, а доповнювався.
- Наявність в тексті фраз, що спонукають до негайної дії: «Натисни!», «Знайти», «Купи прямо зараз», «Заходь зараз!», «Терміново!». Подібні фрази діють на підсвідомому рівні, стимулюючи відвідувача сайту не замислюючись

зробити запропоновану дію. Проте, в деяких випадках, подібні фрази можуть викликати протилежний, відштовхуючий ефект нав'язування.

На сьогоднішній день підставою для ефективності банера є концепція спрощеності. Спостерігається повернення до ненав'язливих текстових форм, максимальної інформативності при мінімумі динаміки і кольорів.

Контекстна реклама

Контекстна реклама на сьогоднішній день це достатньо ефективний спосіб збільшення продажів товарів, продукції або послуг. Оскільки контекстна реклама показується лише для користувачів, що цілеспрямовано шукають інформацію з теми запиту, її ефективність є набагато вищою від звичайної реклами. Відсоток цільових переходів за контекстною рекламою є значно вищим, ніж в банерній рекламі.

За способом показів контекстна реклама поділяється:

- Пошукова реклама показується в результатах пошуку по Інтернету, по сайту або групі сайтів. Обов'язковою умовою показу пошукової реклами є відповідність до пошукового запиту користувача. Користувачі мережі Інтернет при пошуку товару або послуги набирають певні ключові слова в рядках пошукових систем. При виведенні результатів пошуку на першій сторінці відобразатиметься саме Ваша реклама. Вона буде показана всім, хто здійснює пошук за цим словом.
- Тематична реклама показується на сторінці сайту, що входить до рекламної мережі сервісу, якщо тематика реклами відповідає інтересам користувача. Тематична реклама показується як додаткова інформація до змісту сторінок, які проглядає користувач, тим самим, привертаючи набагато більшу цільову аудиторію.

Збільшення ефективності рекламного оголошення:

- Потрібно ретельно підготувати текст рекламного оголошення, що буде містити ключові слова, які найбільше цікавлять потенційну клієнтуру. Для того, щоб оцінити попит та підібрати синоніми чи близькі з тематики слова-

асоціації продавці контекстної реклами надають статистику запитів конкретних слів і фраз за останній місяць.

- Відсіяти можливі помилки в ключових словах або загальні фрази. Слід враховувати особливості реалізації функції морфології на різних пошукових системах. Якщо вказати одне ключове слово, то контекстна реклама може показуватися по всіх запитах, що містять це слово або лише за конкретним запитом. Краще використовувати словосполучення, а не окремі слова. Наприклад, Ви хочете продати цуценят і як ключове слово вибрали слово «цуценя» або «песик». Так, це може бути дуже популярний запит, але за ним знайдуть як відвідувачі які хочуть придбати песика, так і відвідувачі, які хочуть переглянути фотографії чи відправити листівку із зображенням песика. Для контекстної реклами краще використати словосполучення «купити песика» або «купити цуценя дога». Таким чином, оголошення потрапить до потрібної аудиторії, яка готова купити товар, а не просто ознайомитися з ним.

Переваги контекстної реклами

- Контекстна реклама привертає на сайт лише цільову аудиторію. Можна рекламувати товар або послугу незалежно від сезонних коливань, оскільки оголошення демонструється лише для користувачів, які потребують цього товару або послуги в даний момент.

- Створює враження турботи про покупця. Рекламне повідомлення сприймається користувачем не як нав'язлива реклама, а як додаткова корисна інформація, що допомагає зорієнтуватися серед величезної кількості однотипних пропозицій.

- Підвищує лояльність покупців, люди, які поставили питання про товар або послугу, вибрали ваш сайт як відповідь. Вони розглядають можливість купити саме у вас те, що шукали, тобто, налаштовані до покупки саме вашого товару або послуги.

- Ефективність рекламної компанії можна чітко порахувати. Ви витрачаєте на рекламну компанію стільки, скільки хочете. Ви самі задаєте денний бюджет. Контекстні рекламні кампанії можуть проводитися при

низьких бюджетах. Це єдина реклама в Інтернеті, яка працює при бюджетах від 50\$ в місяць.

- Контекстна Інтернет реклама має найвищий рівень ефективності у порівнянні з іншими видами реклами. Середнє співвідношення переходів (CTR) за контекстною рекламою – 50-70 чоловік з 1000. Це в десятки разів вище, ніж в традиційній банерній рекламі. Отримання швидких результатів протягом найближчої доби.

Розміщення контекстної реклами

Контекстну рекламу необхідно розміщувати на тих пошукових системах, які домінують на ринках, на які орієнтовано Ваш сайт. При цьому вибір не повинен зупинятися на одній системі контекстної реклами, слід охоплювати максимально велику аудиторію, отже контекстну рекламу необхідно розміщувати так, щоб охопити хоча б 60-70% цільової аудиторії.

Якщо сайт орієнтовано на міжнародний ринок, то основними гравцями цього ринку контекстної реклами є Google і Yahoo.

Рекламні оголошення в Google розміщуються через його систему контекстної реклами Google AdWords (adwords.google.com). Google має ширший обхват аудиторії, і може націлювати рекламу на користувачів певної країни, області або міста, а також на запити, що введено певними мовами.

Якщо сайт орієнтовано на внутрішній ринок, то основними пошуковими системами, які пропонують розміщення контекстної реклами в Україні є Google, Yandex, BigMir і META.

Рекламні оголошення в Google AdWords і Yandex виводяться праворуч від результатів пошуку, в BigMir і META перед результатами пошуку.

Адресні електронні розсилки

Адресні електронні розсилки - один з найпотужніших способів просування свого Інтернет ресурсу. Розсилки можуть бути одноразовими і регулярними.

Одноразові розсилки

Одноразові розсилки є запрошенням для першого відвідування сайту. Для них потрібно підготувати ретельно складене коротке повідомлення з анотацією рекламованого ресурсу і зібрати певну кількість поштових адрес зацікавленої аудиторії.

Зміст листа обов'язково повинен відповідати інтересам вибраної групи людей, інакше його буде розцінено як небажану рекламу - спам. Відправляти листи потрібно з адреси автора, оскільки листи, які надходять від систем, що автоматично генерують розсилку, часто блокуються поштовими системами.

Рекомендації для проведення одноразових розсилок:

- Не використовуйте часто списки, не частіше двох разів на півроку.
- Не додавайте вибачення з приводу непрошеної пошти до тексту повідомлення.
- Якщо певний адресат буде обурюватися, тоді варто вибачитись.

Регулярні розсилки

Регулярні розсилки є ефективними для сповіщення великого числа підписчиків і забезпечення повторного відвідування сайту. Їх можна проводити лише для людей, які добровільно підписалися на розсилку.

Працювати з кожним підписчиком треба акуратно, блискавично реагувати на його побажання і ніколи не зловживати його увагою. Відправляти листи слід точно за розкладом і лише того змісту, який було заявлено при підписці.

Корисно розсилати іменні листи. Людям подобається, коли до них звертаються за іменем. Не варто відправляти листи списком - хоча б тому, що любий підписчик зможе використати адреси зі списку для власних потреб.

Регулярні поштові розсилки є хорошим способом підтримки відвідуваності. Розсилка допоможе своєчасно донести новини сайту і анонсувати події. Корисно дублювати листи спеціальною сторінкою типу «Новини сайту» і вести архів розсилок.

Способи реалізації

Існують два варіанти розсилок, які цілком можна поєднувати. У першому випадку можна самостійно займатися розсилкою за допомогою спеціальних програм. Основною перевагою цього методу є повний контроль над змістом анонсів. При другому варіанті вся технічна сторона: зберігання адрес, ведення статистики і архіву, анонсування в призначений час перекладається на платні або безкоштовні сервіси поштових розсилок.

Основним недоліком безкоштовних сервісів є надмірна кількість реклами. До плюсів можна віднести можливість отримати величезну аудиторію. Так, наприклад, оформивши розсилку на Subscribe.Ru, ви дістаєте зручний доступ до ведення підписки і архіву. Реклама про вашу розсилку поміщається на головну сторінку сервера і йде в новини інформаційного каналу, які отримують близько 6 мільйонів чоловік.

В текстах розсилок варто дотримуватися певних правил мережного етикету:

- Дотримуватися певного стилю повідомлень, що дозволить відразу зрозуміти суть і авторство.
- Писати лаконічно, але зрозуміло. Довгі листи рідко дочитуються до кінця. Якщо ж лист малозрозумілий - це просто дратує.
- Перевіряти свої листи на наявність граматичних і орфографічних помилок.
- Ввічливе звертання до підписчика.

Розсилання спам-повідомлень

Спам (spam, spamming) - це масова розсилка комерційної, політичної та іншої реклами або іншого виду повідомлень особам, що не висловлювали бажання їх отримувати. Легальність масової розсилки деяких видів повідомлень, для яких не потрібна згода одержувачів, може бути закріплена в законодавстві країни, зокрема в законі про рекламу.

Реклама компанії і продукції

Деякі компанії, що займаються легальним бізнесом рекламують свої товари або послуги за допомогою спаму. Привабливістю такої реклами є низька вартість і великий обхват потенційних клієнтів. Втім, така реклама може мати і зворотний ефект, викликаючи відторгнення у одержувачів.

Реклама незаконної продукції

За допомогою спаму рекламують продукцію, про яку не можна повідомити іншими способами наприклад порнографію чи контрафактні товари (підробки, конфіскації), лікарські засоби з обмеженням обігу, незаконно отриману закриту інформацію (бази даних), неліцензійне програмне забезпечення. Сюди ж відноситься і реклама самих послуг розсилки спаму.

Антиреклама

Імітація спам розсилки може застосовуватися для дискредитації товару, який в ній рекламується та оператора зв'язку, з адреси якого вона (нібито або дійсно) проводиться, тобто спам може використовуватися в недобросовісній конкуренції і «чорному» піарі.

SMS-спам

Спам може поширюватися не лише через Інтернет. Рекламні повідомлення, що присилаються на мобільні телефони за допомогою SMS-повідомлень, особливо неприємні, бо від них важче захиститися.

Найбільш поширені види спаму

За даними Лабораторії Касперського спам в Рунеті і Уанеті розподіляється наступним чином (таблиця 1.1):

Збір адрес електронної пошти

Спамери з'ясовують поштові адреси користувачів в різні способи:

1. Адресу поштової скриньки користувача опубліковано на загальнодоступному сайті, найчастіше в гостьовій книзі, форумі, особистій картці, де його може виявити спеціальна програма (харвестер), що сканує сайти на зразок робота пошукових систем.

Таблиця 1.1

Найбільш поширені види спаму

18,9% - освіта.	4,1% - реклама спамерських послуг.
15,7% - відпочинок і подорожі.	2,7% - для дорослих.
15,5% - медикаменти, товари чи послуги для здоров'я.	2,2% - нерухомість.
9,2% - комп'ютерне шахрайство.	2,2% - юридичні послуги.
6,5% - комп'ютери і Інтернет.	1,9% - особисті фінанси.
5,2% - елітні товари.	1,4% - поліграфія.

2. Якщо адреса поштової скриньки не є унікальним словом, а, наприклад, поширеним іменем або назвою, то воно може бути підібране за словником. В спамерів є спеціальні словники, що містять прості слова, імена людей, назви географічних об'єктів і популярні поєднання мереженого сленгу. Комбінуючи цей словник і перелік адрес поштових серверів, спамери отримують список адрес для розсилки.

3. Якщо комп'ютер людини, з якою переписувався власник даної скриньки, було заражено троянською програмою, то всі адреси з його адресної книги можуть потрапити до спамера.

4. База даних поштових адрес клієнтів певної компанії може бути вкрадена або продана спамерам самими співробітниками компанії.

Заподіяна шкода

Масова розсилка спаму є привабливою для відправника через низьку собівартість. Проте, величезна кількість даремних повідомлень наносить очевидну шкоду для одержувачів. В першу чергу мова йде про час, який витрачено на відсіювання непотрібної пошти і вишукуванні серед неї окремих потрібних листів.

Існує думка, що спам може бути вигідним для провайдерів, оскільки приводить до підвищеного трафіку. Насправді, провайдери також несуть додаткові витрати із-за підвищення зайвого навантаження на канали і

устаткування. Саме провайдерам доводиться витратити ресурси на надмірне устаткування і системи захисту від спаму.

Згідно загальнодоступної статистики спам складає не менше 80% від всіх листів, що пересилаються. За даними деяких досліджень на одного Інтернет користувача сьогодні доводиться до 70 спамових повідомлень за добу. Більша частина його відсікається поштовими серверами під час отримання, але навіть того, що залишилося, достатньо для ускладнення життя користувачів. Провайдери несуть додаткові витрати із-за постійної необхідності боротьби із спамерами (надмірне устаткування, надмірна ємкість каналів, спеціальне програмне забезпечення для розпізнавання спаму).

1.3. Опис предметної області

Внаслідок проведеного дослідження вдалося сформулювати методологію із просування інтернет-ресурсів, що складається з таких етапів: складання семантичного ядра, підготовка текстів сайту (копірайтинг), оптимізація коду, розміщення посилань, моніторинг і аналіз результатів.

Складання семантичного ядра полягає у відборі пошукових фраз, за якими просуватиметься інтернет-ресурс та розподілятиметься по сторінках сайту. В загальному випадку це повний набір слів і словосполучень, що описують певний предмет та його характеристики. Правильне складання семантичного ядра має дуже важливе значення для пошукової оптимізації, тому що саме на його основі будується пошуковий механізм, без якого не можливе просування сайту. Семантичне ядро визначає стратегію створення контенту на сайті і може або звужуватися, або розширюватися, доповнюватися синонімами, новими поняттями. Залежно від типу ресурсу і цілей просування розмір семантичного ядра може складатися як з десятка, так і з тисяч ключових фраз. Існує багато різних програм і сервісів, що автоматизують цю процедуру, проте найкраще скласти семантичне ядро

вручну, використовуючи можливості пошукових систем, зокрема таких, як сервіс Wordstat в Яндексі або сервіс підбору ключових слів від компанії Google. Так можна повністю проконтролювати процес із усвідомленням виконуваних дій.

Під час цієї процедури виділяють три групи запитів: високочастотні, середньочастотні і низькочастотні. Належність запиту до певної групи визначається за відповідною кількістю запитів, зроблених користувачами протягом місяця. Крім частоти введення, необхідно враховувати градації запитів: висококонкурентні, середньоконкурентні і низькоконкурентні. Визначення конкурентності запиту є важкорезалізовуваним і довшим у часі, оскільки вимагає детального “ручного” аналізу відповідних ресурсів. Зокрема процес підбору ключових слів із використанням сервісу Wordstat (Wordstat.Yandex.ua) полягає у введенні міста (регіону), для якого здійснюється аналіз, та відповідного ключового слова. У результаті отримується статистика у різних варіаціях (рисунок 1.8).

Цифри поруч з кожним запитом у результатах підбору надають відомості щодо попередніх прогнозів кількості показів на місяць, які можна отримати, обравши запит як ключове слово. Потім з отриманого списку підібраних запитів потрібно обрати цільові запити (запити, що стосуються тематичного наповнення ресурсу) та після формування останніх завершити роботу над створенням семантичного ядра.

Підготовка текстів сайту (копірайтинг) полягає в підготовці текстів, привабливих для відвідувачів сайту і одночасно оптимізованих під пошукові системи. Існує кілька видів копірайтингу: простий та складний, при цьому їх часто називають рерайтингом. Простий рерайтинг полягає в переробленні однієї статті в іншу, в якій описується все те саме, але іншими словами. На виході формується “унікальний контент”, що не порушує авторських прав джерела. При цьому складний рерайтинг здійснюється аналогічно, але на вході існує множина джерел. У результаті цього процесу створюється оптимізований текст щодо запитів пошукових систем.

Яндекс підбір слів

просування сайту Підібрати

● За словами ○ За регіонами ● Історія запитів

Всього показів за фразу «просування сайту»: 1 288 Список Карта

Все	Регіони	Міста	Показів на місяць	Регіональна популярність
	Київ		709	1 602%
	Харків		33	380%
	Львів		31	1 512%
	Рівне		15	2 161%
	Полтава		13	684%
	Санкт-Петербург		6	8%
	Донецьк		5	49%
	Івано-Франківськ		4	697%
	Одеса		3	45%
	Москва		3	1%
	Єкатеринбург		2	6%
	Дніпропетровськ		1	16%
	Челябінськ		1	6%
	Мюнхен		1	472%

Рис. 1.8. Підбір слова за допомогою сервісу Wordstat

Під оптимізацією коду сайту розуміють технічний процес щодо зменшення розміру коду, який передається користувачам і пошуковим системам під час завантаження. Оптимізація коду має значні переваги, зокрема: зменшення розміру сторінок сайту і прискорення їх завантаження; зниження навантаження на сервер хостингу, що позитивно позначається на його стабільності (особливо актуально для проектів з високим рівнем відвідуваності) [4].

Розміщення посилань належить до “зовнішніх чинників” оптимізації – без їх врахування неможливе успішне просування сайту в конкурентних тематиках. Причому ці роботи необхідно здійснювати постійно, створюючи зовнішнє “посилальне оточення” [2] та відображаючи необхідний текст навколо посилань, тобто текст, що відповідає тематиці товарів.

Моніторинг і аналіз результатів полягає в постійному контролі позицій інтернет-ресурсу в результатах пошуку за всіма відібраними запитамі і аналізі змін, що відбуваються. Ця процедура дає змогу оперативно реагувати на зміни

в алгоритмах ранжування пошукових машин, оцінювати і своєчасно вносити необхідні корективи.

Схема процесу SEO-оптимізації. Що стосується розглянутого процесу Seo-оптимізації із залученням відповідних фахівців, то вона має вигляд, як на рисунку 1.8. Тобто процес аналізу та просування веб-сайту складається з таких етапів:

1. Аналіз інтернет-ресурсу та створення нового проекту. При надходженні проекту в роботу за ним закріплюється відповідальний фахівець – проект-менеджер. На всіх етапах робіт із просування з клієнтом взаємодіє закріплений за проектом акаунт-менеджер. На початковому етапі на підставі побажань клієнта визначають мету та пріоритети проекту. Проект-менеджер аналізує сайт й ресурси конкурентів. Після обліку отриманих акаунт-менеджером під час діалогу із клієнтом даних проект-менеджер розподіляє цілі сторінками ресурсу.

2. Створення плану робіт. На підставі отриманих даних (у результаті аналізу сайту та ресурсів конкурентів з врахуванням рівня складності кожного із ключових запитів) проект-менеджер за допомогою співробітників відділу просування формує план робіт за проектом.

3. Просування сайту складається з внутрішньої та зовнішньої оптимізації.

3.1. Внутрішня оптимізація інтернет-ресурсу

3.1.1. Програмісти технічного відділу за підготовленим проект-менеджером технічним завданням вносять необхідні для просування ресурсу виправлення в html-код. У випадку відсутності повного доступу до сайту акаунт-менеджер передає проект-менеджеру підготовлене технічне завдання на внесення необхідних коректив в html-код сайту технічним фахівцям з боку клієнта. Після внесення коректив проект-менеджер перевіряє коректність виконання технічного завдання і в разі потреби готує нове завдання із необхідними уточненнями й доповненнями.

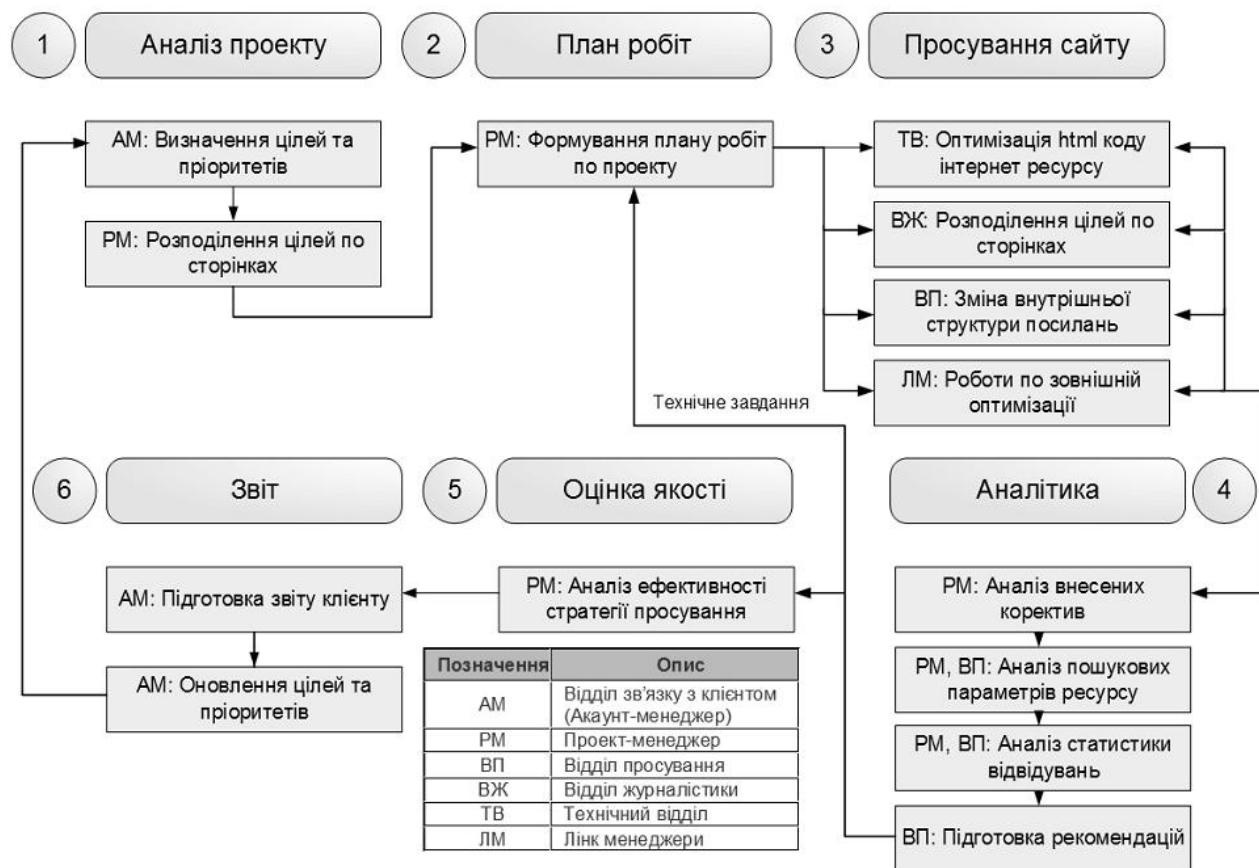


Рис. 1.9. Схема процесу SEO-оптимізації

3.1.2. Копірайтери відділу журналістики на підставі підготовленого проект-менеджером технічного завдання готують необхідні для просування сайту корективи тексту та статті. Акаунт-менеджер погоджує нові текстові матеріали із клієнтом і за необхідності копірайтери вносять повторні зміни в текстовий матеріал на підставі отриманої від клієнта інформації. Потім технічний відділ розміщує погоджені варіанти текстів.

3.1.3. Фахівці відділу просування готують технічне завдання на коригування зовнішньої “посилальної структури” для оптимізації передавання інформації між сторінками ресурсу. Технічний відділ вносить відповідні корективи в ресурс.

3.2. Зовнішня оптимізація сайту. На цьому етапі проводиться робота над зовнішніми факторами ранжування ресурсу. Згідно з поставленим завданням лінк-менеджери розміщують у ресурсі посилання із сайтів,

близьких по тематиці (для збільшення ТІЦ), а також високо ранжованих сайтів (для збільшення загальної важливості інтернет-ресурсу) [2].

4. Аналітика. На цьому етапі аналізуються основні внутрішні й зовнішні параметри інтернет-ресурсу, який просувається, розглядається динаміка позицій сайту в різних пошукових системах за кожним із ключових запитів та моделюється статистика відвідуваності сайту. На підставі отриманих даних відділ просування формує список рекомендацій з подальшої роботи.

5. Оцінка якості. На підставі даних, отриманих на попередньому етапі, проект-менеджер оцінює основні параметри просування сайту з врахуванням мети і пріоритетів, сформульованих на першому етапі. Потім проект-менеджер збирає дані, необхідні для формування звіту клієнтові.

6. Звіт клієнтові. На підставі наданих проект-менеджером даних акаунт-менеджер формує звіт для клієнта. Після аналізу звіту клієнтом у випадку корегування вхідних даних за цілями і пріоритетами акаунт-менеджер передає відповідну інформацію проект-менеджеру для зміни стратегії просування інтернет-ресурсу. У випадку одержання нових цілей або коригування пріоритетів роботи проводяться, починаючи з другого етапу.

Якщо список цілей і пріоритетів залишився незмінним, роботи тривають, починаючи з третього етапу. У зв'язку з тим, що в алгоритм ранжування сайтів роботами пошукових систем періодично вносяться корективи, необхідною умовою просування є корегування внутрішніх параметрів сторінок, які впливають на ранжування.

Етап зовнішньої оптимізації сайту також є циклічним, тобто роботи з нарощування й корегування зовнішнього “посилального оточення” сайту, що просувається, необхідно проводити безупинно для підтримки й поліпшення досягнутих позицій. З огляду на зазначені етапи аналізу та просування інтернет-ресурсу було розроблено структурну схему модульної системи автоматизації процесу Seo-оптимізації (рисунок 1.10).

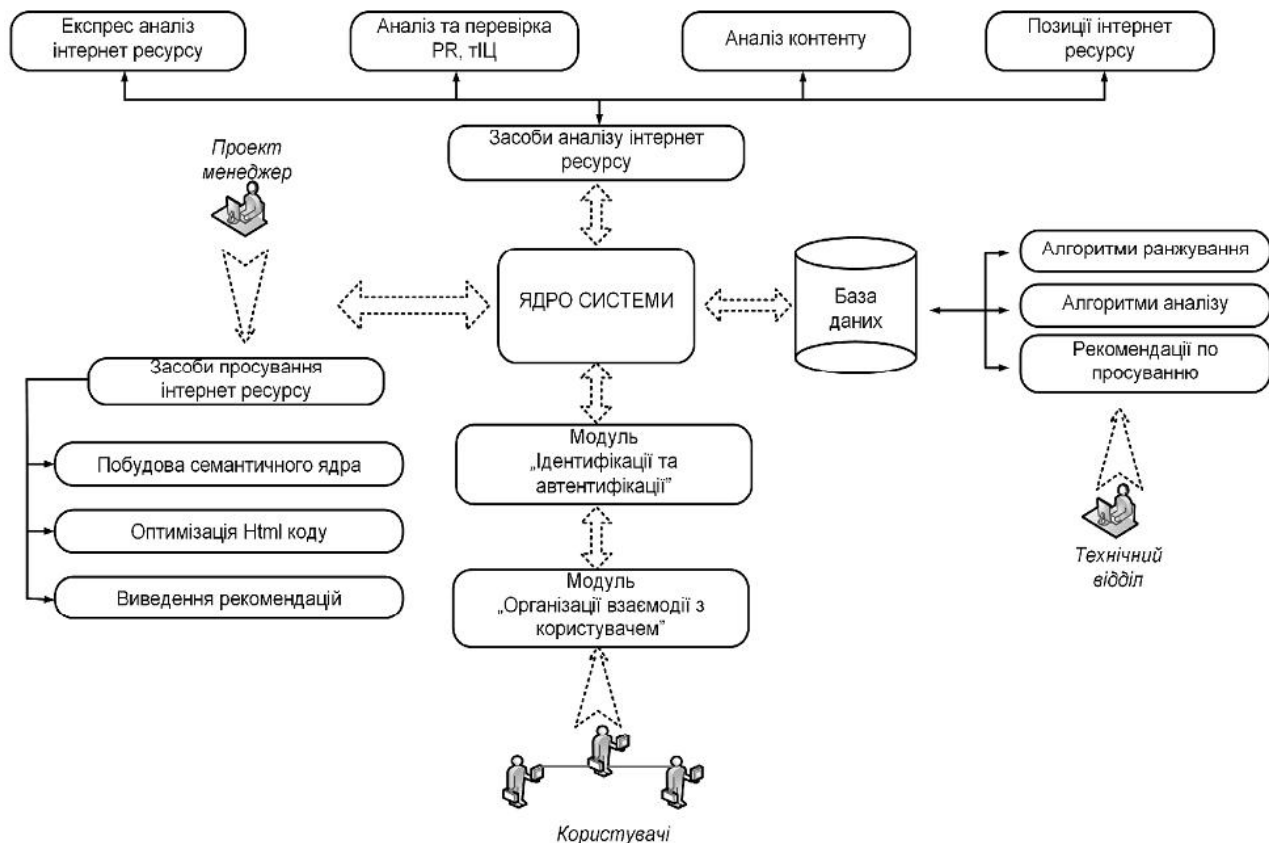


Рис. 1.10. Структурна схема системи автоматизації процесу Seo-оптимізації

Основними модулями системи є: засоби аналізу та просування інтернет-ресурсу, ядро, модуль організації взаємодії з користувачем та модуль ідентифікації та автентифікації. Система надає два рівні доступу: на рівні менеджера (адміністратора) та на рівні користувача. Менеджер (адміністратор), володіючи повними правами, може: активувати та видаляти користувачів, змінювати інформацію бази даних тощо, а права користувача обмежуються аналізом інтернет-ресурсу та виведенням рекомендацій щодо його просування. Важливою операцією, яку виконує модуль адміністрування, є опрацювання граничних ситуацій, що реалізується визначенням реакції кожного об'єкта й всієї системи на певні зовнішні дії: ініціалізацію, термінацію й обвал.

Зокрема процес ініціалізації полягає в приведенні системи до фіксованого початкового стану: оголошення всіх початкових змінних та параметрів задачі та формування ієрархії об'єктів. Процес термінації полягає

у вивільненні всіх апаратних та програмних ресурсів, зайнятих під задачі системи. Процес обвалу являє собою незаплановану термінацію системи [6].

Ядро системи – основний модуль системи, в якому опрацьовуються всі операції та роботи з її базою даних. Ядро системи координує роботу решти модулів з метою формування цілісної системи.

Особливою функцією зазначеного модуля є організація процесу введення вхідної інформації, її опрацювання та збереження.

Модуль організації взаємодії з користувачем призначений для забезпечення особливостей роботи залежно від прав користувача. Реалізується через графічний інтерфейс користувача, що об'єднує всі елементи та компоненти програми, які впливають на взаємодію з користувачем.

Засоби аналізу інтернет-ресурсу містять команди, що реалізують: експрес-аналіз, аналіз та перевірку PR та тІЦ, аналіз контенту та визначають позицію ресурсу в пошукових системах. Засоби просування реалізують: побудову семантичного ядра, оптимізацію HTML-коду та виведення рекомендацій щодо просування ресурсу.

1.4. Специфікація вимог до системи

Специфікація вимог до програмної системи – це специфікація окремого програмного продукту, програми або набору програм, які виконують деякі дії в деякому середовищі. Тобто – це повний опис поведінки системи що розробляється [3].

В загальному випадку специфікація включає наступне:

- глосарій проекту;
- опис варіантів використання.

Наведемо список основних термінів та понять в області розробки програмної системи підтримки просування інформаційного порталу по пошукових системах – глосарій. Глосарій – список понять в специфічній

області знання з їх визначеннями [3]. Ці поняття та визначення подано у таблиці 1.2.

Таблиця 1.2

Глосарій

Термін	Опис терміну
1. Основні поняття та категорії предметної області та проекту	
Сайт	сукупність веб-сторінок, доступних у мережі Інтернет, які об'єднані як за змістом, так і за навігацією. Фізично сайт може розміщуватися як на одному, так і на кількох серверах.
SEO-оптимізація	процес коригування HTML-коду, текстового наповнення (контенту), структури сайту, контроль зовнішніх чинників для відповідності вимогам алгоритму пошукових систем, з метою підняття позиції сайту в результатах пошуку в цих системах за певними запитами користувачів. Чим вище позиція сайту в результатах пошуку, тим більша ймовірність, що відвідувач перейде на нього з пошукових систем, оскільки люди зазвичай йдуть за першими посиланнями.
Інтернет-реклама	реклама, що розміщується в мережі Інтернет, переважно на добре зарекомендованих і популярних веб-сайтах
Банер	графічне зображення рекламного характеру
Програмне забезпечення	сукупність програм системи обробки інформації і програмних документів, необхідних для експлуатації цих програм [3].
Програмний продукт	програмний засіб, програмне забезпечення, які призначені для постачання користувачів (покупцеві, замовникові) [3].
Бізнес-процес	будь-яка діяльність, що має вхідний продукт, додає вартість до нього, та забезпечує вихідний продукт для внутрішнього або зовнішнього споживача [3].
База даних	логічно впорядкований та взаємопов'язаний набір даних, що використовуються спільно та призначені для задоволення інформаційних потреб користувачів.

Продовження таблиці 1.2

2. Користувачі системи	
Користувач	Особа, яка займається просування власного сайту
Проект-менеджер (Адміністратор)	Особа, яка займається безпосереднім управлінням системою підтримки просування інформаційного порталу по пошукових системах.
3. Вхідні та вихідні документи	
База даних	логічно впорядкований та взаємопов'язаний набір даних, що використовуються спільно та призначені для задоволення інформаційних потреб користувачів.
Список рекомендацій	Список пропозиції щодо дій, які, можливо, будуть потрібними у результаті зроблених висновків.

У таблицях 1.3 – 1.8 наведено опис варіантів використання, що реалізують основну функціональність програмної системи. Діаграма варіантів використання представлена на рисунку 1.11.

Таблиця 1.3

Варіант використання «Аналіз інтернет ресурсу»

Контекст використання	Управління просуванням сайту.
Дійові особи	Користувач
Передумова	Користувач аутентифікований та авторизований.
Тригер	Натиснення кнопки «Додати адресу сайту».
Сценарій	1. Заповнення полів. 2. Перевірка правильності введених даних. 3. Натиснення кнопки «Зберегти».
Постумова	Відображення доданого для аналізу сайту.

Таблиця 1.4

Варіант використання «Отримання рекомендацій по просуванню сайту»

Контекст використання	Управління просуванням сайту.
Дійові особи	Користувач
Передумова	Користувач аутентифікований та авторизований.
Тригер	Вибір сайту.
Сценарій	-
Постумова	Відображення рекомендацій.



Рис. 1.11. Діаграми варіантів використання

Таблиця 1.5

Варіант використання «Модифікація правил та алгоритмів»

Контекст використання	Управління посадами.
Дійові особи	Проект-менеджер (Адміністратор)
Передумова	Користувач аутентифікований та авторизований.
Тригер	Вибір правил та алгоритмів.
Сценарій	<ol style="list-style-type: none"> 1. Вибір правила. 2. Визначення змін. 3. Натиснення кнопки «Зберегти».
Постумова	Відображення списку правил та алгоритмів

Таблиця 1.6

Варіант використання «Реєстрація користувача»

Контекст використання	Управління користувачами.
Дійові особи	Проект-менеджер (Адміністратор)
Передумова	Користувач аутентифікований та авторизований.
Тригер	Відкриття розділу «Користувачі».
Сценарій	<ol style="list-style-type: none"> 1. Вибір користувача. 2. Заповнення необхідних полів. 3. Перевірка введених даних. 4. Натиснення кнопки «Зберегти».
Постумова	Відображення списку користувачів.

Таблиця 1.7

Варіант використання «Редагування інформації про користувача»

Контекст використання	Управління користувачами.
Дійові особи	Проект-менеджер (Адміністратор)
Передумова	Користувач аутентифікований та авторизований.
Тригер	Відкриття розділу «Користувачі».
Сценарій	<ol style="list-style-type: none"> 1. Вибір користувача. 2. Натиснення кнопки «Редагувати». 3. Заповнення необхідних полів. 4. Перевірка введених даних. 5. Натиснення кнопки «Зберегти».
Постумова	Відображення списку користувачів.

Таблиця 1.8

Варіант використання «Видалення користувача»

Контекст використання	Управління користувачами.
Дійові особи	Проект-менеджер (Адміністратор)
Передумова	Користувач аутентифікований та авторизований.
Тригер	Відкриття розділу «Користувачі».
Сценарій	<ol style="list-style-type: none"> 1. Вибір користувача. 2. Натиснення кнопки «Видалити». 3. Підтвердження операції видалення.
Постумова	Відображення списку користувачів.

В таблиці 1.9 наведено специфікацію функціональних вимог програмної системи.

Таблиця 1.9

Специфікація функціональних вимог

Ідентифікатор вимоги	Назва вимоги	Атрибути вимоги		
		Пріоритет	Складність	Контакт
1	Аналіз інтернет ресурсу	рекомендоване	середня	Користувач
2	Отримання рекомендацій по просуванню сайту	обов'язкове	висока	Користувач
3	Модифікація правил та алгоритмів	обов'язкове	висока	Проект-менеджер (Адміністратор)
4	Реєстрація користувача	обов'язкове	висока	Проект-менеджер (Адміністратор)
5	Редагування інформації про користувача	обов'язкове	висока	Проект-менеджер (Адміністратор)
6	Видалення користувача	обов'язкове	висока	Проект-менеджер (Адміністратор)

Специфікацію нефункціональних вимог наведено в таблиці 1.10.

Наведемо специфікацію суттєвих для проекту нефункціональних вимог:

1. Застосовність:
 - мінімальний час для навчання звичайних і досвідчених користувачів;
 - відповідність стандартам графічного інтерфейсу.
2. Надійність:
 - постійна безвідмовна робота;
 - пропускна здатність каналу зв'язку 100 Mb/s;
 - забезпечення можливості віддаленого доступу до комп'ютера, на якому буде встановлена система;
 - доступність – 5%.
3. Робочі характеристики:
 - швидкість завантаження інтернет-ресурсу: 0,1 – 1 с;
 - число транзакцій: 100 / 1 с;
 - використання ресурсів: від 1 Gb, в залежності від кількості студентів.
4. Проектні обмеження:
 - Операційна система Microsoft Windows 7/8;
 - MySQL;
 - PHP 5.5;
 - Yii Framework.
5. Вимоги до документації
 - наявність інтерактивної довідки.
6. Інтерфейси:
 - інтерфейс користувача – веб-інтерфейс.

Таблиця 1.10

Специфікація нефункціональних вимог

Ідентифікатор вимоги	Назва вимоги	Атрибути вимог		
		Пріоритет	Складність	Контакт
Застосовність				
1.1	Час, необхідний для навчання звичайних і досвідчених користувачів	Рекомендована	Низька	Адміністратор
1.2	Основні вимоги застосовності нової системи відносно інших систем, які знають користувачі	Опційна	Низька	Адміністратор
1.3	Вимоги по відповідальності стандартам графічного інтерфейсу користувача	Рекомендована	Низька	Адміністратор
Надійність				
2.1	Доступність	Обов'язкова	Середня	Адміністратор
2.2	Середній час безвідмовної роботи	Рекомендована	Середня	Адміністратор
2.3	Точність	Обов'язкова	Середня	Адміністратор
Робочі характеристики				
3.1	Використання ресурсів	Рекомендована	Середня	Адміністратор
4.1	Вимоги до технології програмування	Рекомендована	Середня	Адміністратор

Висновки до розділу 1

Здійснено опис предметної області, напрями діяльності. Визначено склад функцій, що входять до бізнес-процесу на основі яких розроблено схему управління бізнес-процесом. Проведено аналіз відомих методів оптимізації SEO. Здійснено аналіз вимог до програмної системи.

РОЗДІЛ 2

ПРОЕКТУВАННЯ СИСТЕМИ ПІДТРИМКИ ПРОСУВАННЯ ІНФОРМАЦІЙНОГО ПОРТАЛУ ПО ПОШУКОВИХ СИСТЕМАХ

2.1. Розроблення архітектури програмної системи

Для проектування системи підтримки просування інформаційного порталу по пошукових системах важливою вимогою є надання рекомендацій, які ґрунтуються на принципах побудови систем підтримки прийняття рішень та методології системного аналізу. Серед них можна виділити:

- принцип остаточної мети – визначає глобальну мету системи та має абсолютний пріоритет, тобто визначається основна ціль системи – аналіз та просування інтернет-ресурсу, тобто надання рекомендацій користувачам щодо підвищення релевантності веб-сайту;

- принцип єдності – передбачає розгляд системи як одного цілого, тобто всі елементи системи є взаємопов'язані. У нашому випадку це є: підсистема, що забезпечує внесення нової інформації у програму (нових фактів); підсистема надання рекомендацій на основі збережених знань (фактів), машина логічного виведення; підсистема, яка дає змогу вивести рекомендації у відповідній формі (система виведення);

- принцип зв'язності – містить сформовані рекомендації, що ґрунтуються на внесених та збережених знаннях (фактах) у базі даних;

- принцип модульності – передбачає конструювання системи із сукупності модулів. Проектовану систему можна поділити на такі модулі: модуль внесення даних користувача системи, модуль внесення нових правил, головний модуль логічного виведення, модуль виведення сформованої системи рекомендацій (рисунки 2.1);



Рис. 2.1. Модульна структура системи

- принцип функціональності – забезпечує загальну функціональність системи, а саме, функціональність внесення даних користувачів, внесення нових знань й формування рекомендацій, збереження звітів користувачів;
- принцип розвитку – полягає у внесенні нових знань протягом функціонування системи.

Наведені принципи дають змогу відобразити особливості побудови системи та врахувати важливі взаємні зв'язки між її елементами. Опис сутності системи, яка досліджується, розглянемо відповідно до мети, яка визначає призначення системи та конкретизується за допомогою дерева цілей (рисунок 2.2).

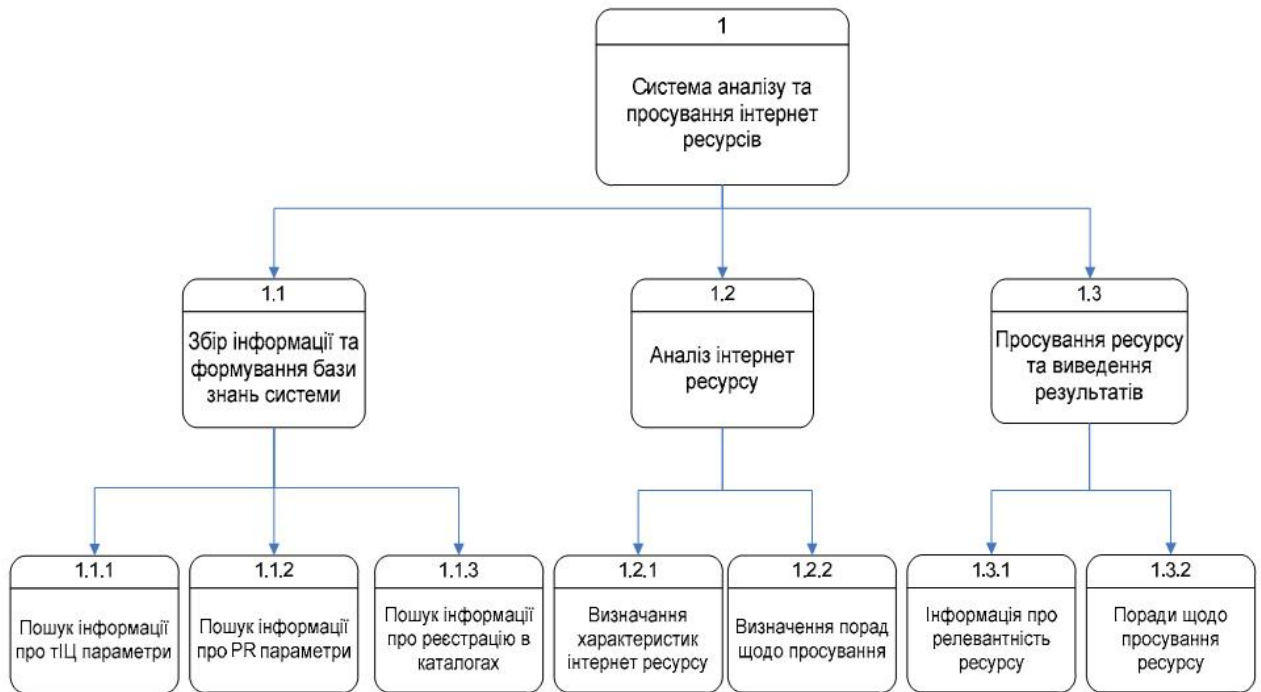


Рис. 2.2. Дерево цілей системи

Згідно з методологією системного аналізу головною ціллю є створення системи підтримки просування інформаційного порталу по пошукових системах. Це завдання можливо виконати тільки після виконання всіх підцілей.

Від вершини дерева цілей відходять три гілки, які поділяють головну ціль на три підцілі. Першою підціллю є “Збирання інформації та формування бази знань”. На цьому етапі необхідно виділити необхідні характеристики та визначити структуру знань, які необхідні для роботи системи. Зазначена ціль поділяється на три підцілі, метою яких є створення логічної моделі бази знань, вибір методів подання знань, проектування засобів автоматизованого опитування експертів, що дасть змогу наповнювати та розширювати базу знань автоматизовано.

Другою підціллю є “Аналіз інтернет-ресурсу”. Метою цієї гілки є збирання та відображення інформації, що стосується вибраного інтернет-ресурсу, а також методів та засобів аналізу цих даних. Ця підціль поділяється на дві підцілі: визначення характеристик інтернет-ресурсу (здійснює визначення поточної релевантності ресурсу) та визначення порад

щодо просування (відобразатиме за допомогою продукційної моделі можливі варіанти підвищення позиції в пошукових видачах).

Третя підціль є “Просування ресурсу та виведення результатів”. Під час виконання цієї підцільі будуть реалізовані спеціалізовані засоби відображення інформації щодо релевантності сайту та поради (рекомендації) щодо його просування.

Під час проектування прикладної програми важливими завданнями, які необхідно вирішити, є визначення всіх функцій розроблюваного продукту, інтерфейсу зв'язку між складовими модулями та набір проектних рішень для забезпечення безвідмовної роботи програми. Архітектура програми забезпечує поєднання розроблених компонентів, які утворюють модулі та задає контекст, у межах якого приймаються наступні проектні рішення. Розроблення архітектури вимагає прийняття таких проектних рішень: створення програми з модулів та керування базами даних.

Створення програми з модулів визначає виділення набору окремих компонентів, що взаємозалежні і мають добре визначені інтерфейсні зв'язки з іншими компонентами програми. Вони визначають форму всіх взаємодій між ними, проте не містять з'єднань на їх внутрішню структуру. З огляду на те, проектувана програма повинна складатись зі своєрідних блоків, які зображуються за допомогою контекстної діаграми.

Керування базами даних здійснюватиметься у двох аспектах: керування внутрішніми та зовнішніми базами. Під внутрішніми базами даних у розроблюваній системі розумітимемо зберігання внутрішньої структури програми в основній пам'яті комп'ютера. При цьому використання файлів дасть змогу звертатись до них зі сторони багатьох користувачів та легко переносити їх на різні платформи. Своєю чергою, зовнішні бази даних (зберігаються в зовнішній пам'яті комп'ютера) використовуватимуться для збереження даних користувачів, оскільки зазначені типи даних погано піддаються структуризації та мають низьку інформаційну щільність.

Формулювання методологічного підґрунття проектування системи аналізу та просування інтернет-ресурсів неможливе без побудови контекстної діаграми, яка відображає взаємодію системи із зовнішнім середовищем. Представлена контекстна діаграма (рисунок 2.3) описує вхідні та вихідні потоки системи та дає змогу систематизувати та автоматизувати етапи розроблення програмного забезпечення.



Рис. 2.3. Контекстна діаграма потоків даних

Застосування контекстної діаграми є обмеженим, оскільки вона відображає лише основні сутності розроблюваної системи, не роблячи наголосу на їх внутрішню побудову. З огляду на те відображається діаграма декомпозиції потоків даних першого рівня, яка показує основні процеси системи аналізу та просування інтернет-ресурсів (рисунок 2.4). Основними процесами є:

- аналіз параметрів інтернет-ресурсу полягає у комплексному аналізі параметрів веб-сайту, його Google PR та Яндекс тІЦ показників, наявність в каталогах (DMOZ) та кількість проіндексованих сторінок;

- вибір рекомендацій щодо просування полягає у збиранні проаналізованої інформації та виборі рекомендацій щодо оптимізації веб-сайту;
- виведення результатів полягає у поєднанні отриманих результатів та виведенні інформації про сайт із рекомендаціями щодо просування.



Рис. 2.4. Декомпозиція діаграми потоків даних

Деталізований процес аналізу параметрів інтернет-ресурсу відображено на рисунку 2.5.

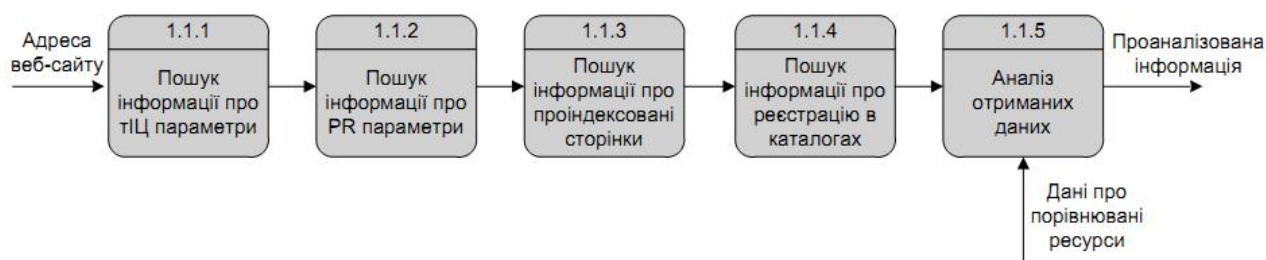


Рис. 2.5. Деталізація процесу «Аналіз параметрів ресурсу»

Деталізація процесу демонструє процеси пошуку інформації щодо параметрів веб-сайту та є складовою частиною системи підтримки просування інформаційного порталу по пошукових системах. На рисунку 2.6 представлено діаграму станів для додавання нових даних для SEO.

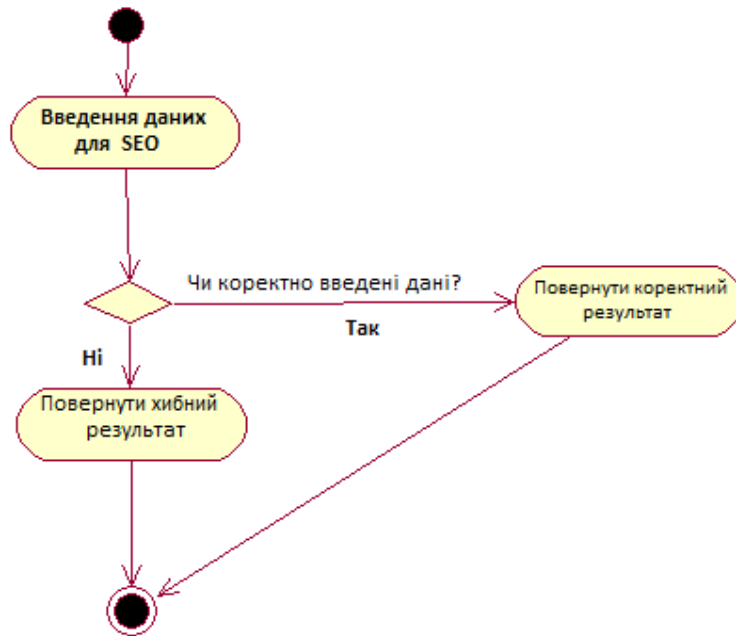


Рис. 2.6. Діаграма станів для додавання нових даних для SEO

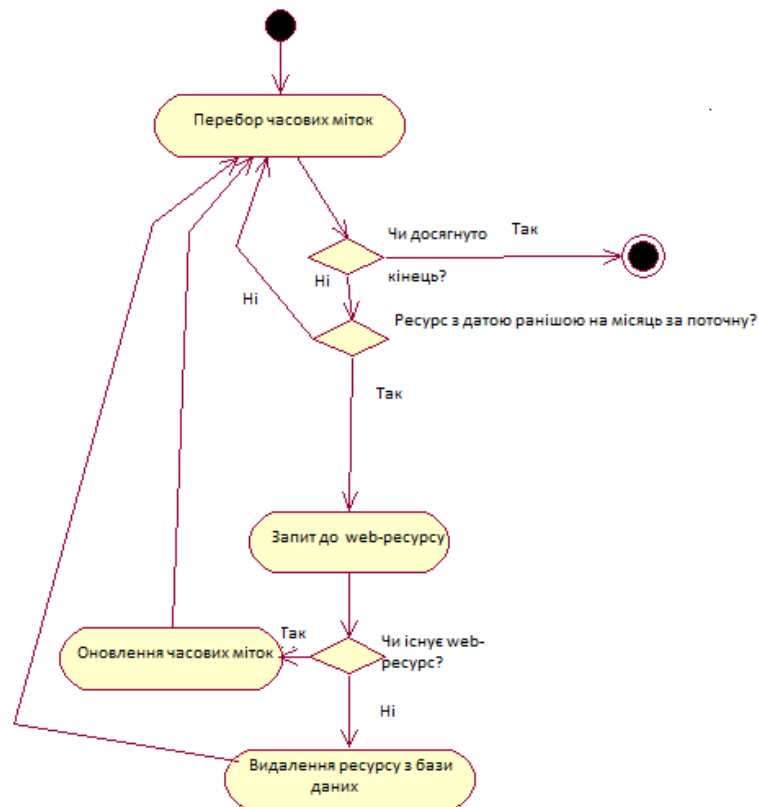


Рис. 2.7. Діаграма станів для отримання часових характеристик SEO

На рисунку 2.8. представлено діаграму розгортання системи.

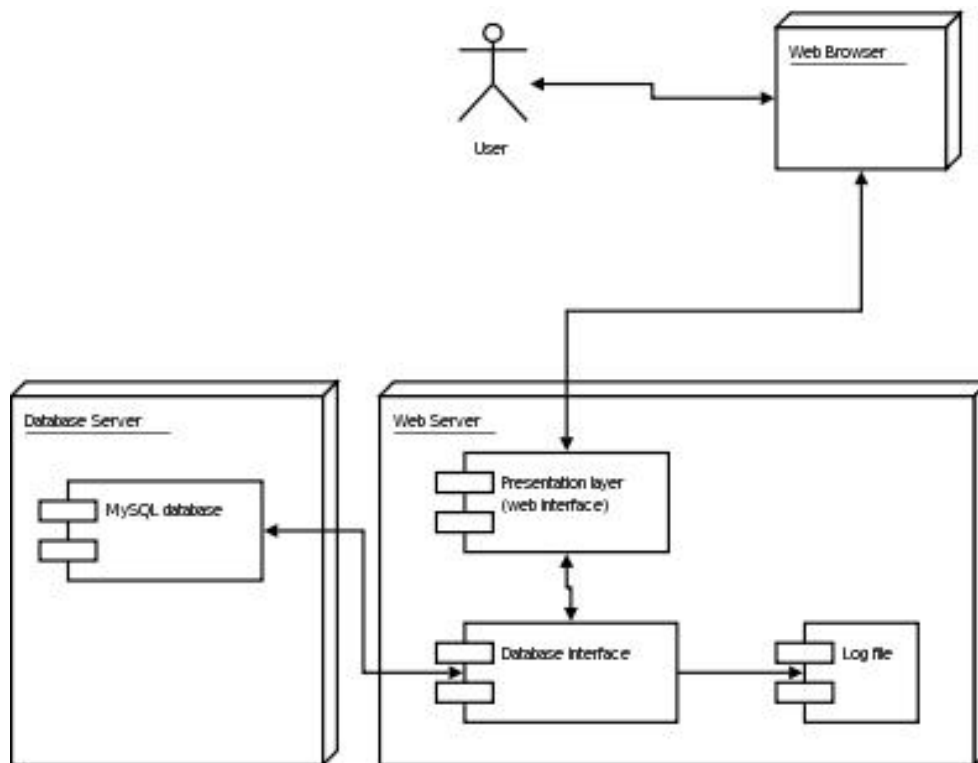


Рис. 2.8. Діаграма розгортання

2.2. Проектування структури бази даних

Розглянемо структурну модель бази даних, отриману в результаті дослідження предметної області і побудови функціональної діаграми.

Спершу визначимо два основні завдання проектованої системи :

- зберігати інформацію про сайти, SEO і користувачів.
- додавати інформацію про сайти, SEO і користувачів.

Із завдань проектованої системи були виявлені такі сутності як: «Сайти», «Користувачі», «Групи користувачів», «Категорія».

Основними є сутності «Сайти» та «Користувачі», які використовуються для формування списку сайтів для просування і списку користувачів відповідно.

Структурна модель бази даних системи підтримки просування інформаційного порталу по пошукових системах зображена на рисунку 2.9.

На основі функціональної моделі була створена фізична модель бази даних. Кожна сутність структурної моделі є таблицею у фізичній моделі даних.

Для створення бази даних і таблиць був використаний локальний сервер, а також СУБД MySQL. База даних і таблиці були створені шляхом виконання запитів на мові SQL. Створена база даних була збережена під назвою «prog_shop».

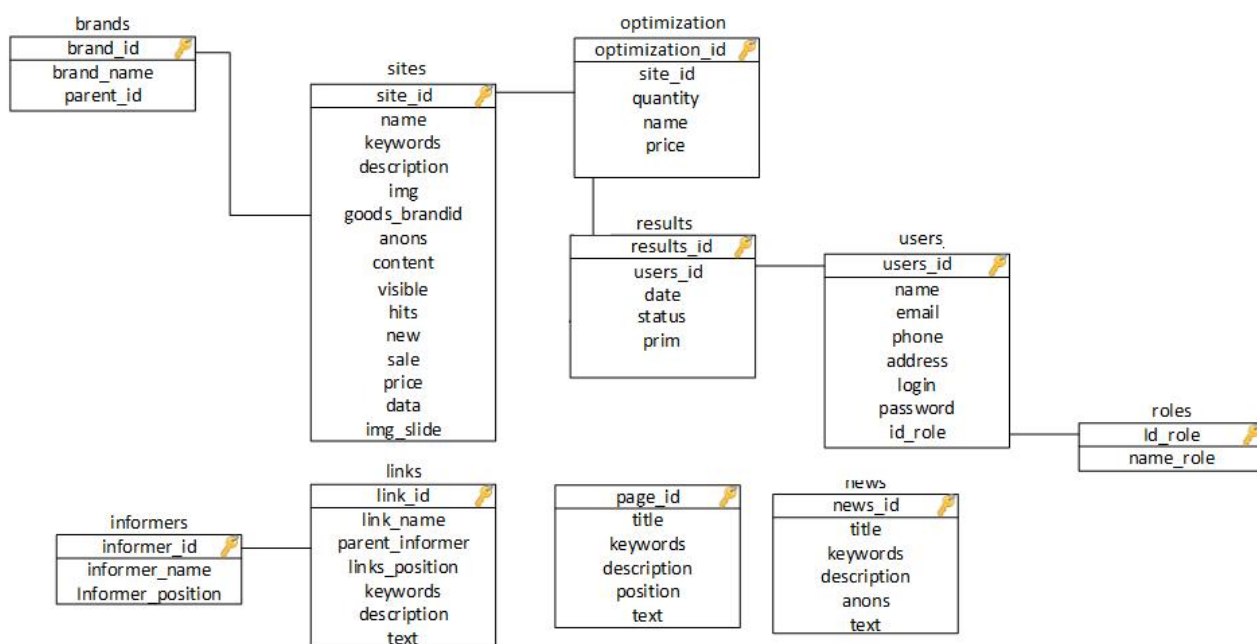


Рис. 2.9. Структурна модель бази даних

Для кожної таблиці були визначені поля і їх типи. Для зв'язку між таблицями використовувалися первинні ключі, які були визначені для кожної таблиці.

Поля таблиці «brands» та їх атрибути приведені в таблиці 2.1.

Таблиця 2.1

Опис атрибутів відношення brands

Назва атрибутів	Тип полів	Розмір полів, байт	Ключове поле
brand_id	tinyint	3	Так
brand_name	varchar	255	Ні
parent_id	tinyint	3	Ні

Поля таблиці «sites» і їх атрибути приведені в таблиці 2.2.

Таблиця 2.2

Опис атрибутів відношення sites

Назва атрибутів	Тип полів	Розмір полів, байт	Ключове поле
sites_id	int	10	Так
name	varchar	255	Ні
keywords	varchar	255	Ні
description	varchar	255	Ні
img	varchar	255	Ні
goods_brandid	tinyint	3	Ні
anons	text		Ні
content	text		Ні
visible	enum('0', '1')		Ні
hits	enum('0', '1')		Ні
new	enum('0', '1')		Ні
sale	enum('0', '1')		Ні
price	float		Ні
data	date		Ні
img_slide	varchar	255	Ні

Таблиця 2.3

Опис атрибутів відношення users

Назва атрибутів	Тип полів	Розмір полів, байт	Ключове поле
users_id	int	10	Так
name	varchar	255	Ні
email	varchar	255	Ні
phone	varchar	255	Ні
address	varchar	255	Ні
login	varchar	255	Ні
password	varchar	255	Ні
id_role	enum('0', '1')		Ні

Поля таблиці «roles» і їх атрибути приведені в таблиці 2.4.

Таблиця 2.4

Опис атрибутів відношення roles

Назва атрибутів	Тип полів	Розмір полів, байт	Ключове поле
id_role	tinyint	3	Так
name_role	varchar	255	Ні

Таблиця 2.5

Опис атрибутів відношення links

Назва атрибутів	Тип полів	Розмір полів, байт	Ключове поле
link_id	tinyint	3	Так
link_name	varchar	255	Ні
parent_informer	tinyint	3	Ні
links_position	tinyint	3	Ні
keywords	varchar	255	Ні
description	varchar	255	Ні
text	text		Ні

Поля таблиці «informers» і їх атрибути приведені в таблиці 2.6.

Таблиця 2.6

Опис атрибутів відношення informers

Назва атрибутів	Тип полів	Розмір полів, байт	Ключове поле
informer_id	tinyint	3	Так
informer_name	varchar	255	Ні
Informer_position	tinyint	3	Ні

Поля таблиці «page» і їх атрибути приведені в таблиці 2.7.

Таблиця 2.7

Опис атрибутів відношення page

Назва атрибутів	Тип полів	Розмір полів, байт	Ключове поле
page_id	tinyint	3	Так
title	varchar	255	Ні
keywords	varchar	255	Ні
description	varchar	255	Ні
position	tinyint	3	Ні
text			Ні

Поля таблиці «news» і їх атрибути приведені в таблиці 2.8.

Таблиця 2.8

Опис атрибутів відношення page

Назва атрибутів	Тип полів	Розмір полів, байт	Ключове поле
news_id	int	10	Так
title	varchar	255	Ні
keywords	varchar	255	Ні
description	varchar	255	Ні
anons	text		Ні
text	text		Ні
date	date		Ні

Висновки до розділу 2

У цьому розділі розроблено архітектуру програмного додатку, що дозволить краще зрозуміти функції основних його частин. Створено та описано структурну схему, основними компонентами якої є: рівень клієнта, рівень бізнес-логіки та рівень даних. Описано функціональну структуру системи та її основних елементів – модулів обробки даних. Визначено основні елементи бази даних та встановлено зв'язки між ними. Спроектовано структуру бази даних.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ПІДТРИМКИ ПРОСУВАННЯ ІНФОРМАЦІЙНОГО ПОРТАЛУ ПО ПОШУКОВИХ СИСТЕМАХ

3.1. Програмна реалізація системи

Після проведення аналізу доцільності розробки системи з'явилася необхідність вирішити, яке програмне забезпечення нам знадобиться.

Створення будь-якого сайту починається з розробки дизайну, який повинен бути унікальним і відповідати обраній тематиці. Для виконання цієї частини завдання не кожен графічний засіб підійде. Наступним етапом є верстка сторінок і окремих частин сайту, які будуть описані HTML кодом з використанням скриптів на мові програмування PHP. Оскільки сховищем даних, яке буде працювати з сайтом є засіб MySQL, а система адміністрування, написана мовою PHP, для налагодження скриптів і модулів сайту було встановлено локальний web – сервер.

Ефективність є виключно важливим чинником при програмуванні для розрахованих на багато користувачів середовищ, до числа яких належить і web. Дуже важлива перевага PHP полягає в його «движку». «Движок» PHP не є ні компілятором, ні інтерпретатором. Він є транслуючим інтерпретатором. Такий пристрій «движка» PHP дозволяє обробляти сценарії з достатньо високою швидкістю.

За деякими оцінками, більшість PHP-сценаріїв (особливо не дуже великих розмірів) обробляються швидше аналогічних їм програм, написаних на Perl. Проте, якою б не була робота з PHP, виконувани відкомпільовані файли будуть працювати значно швидше - в десятки, а іноді і в сотні разів. Але продуктивність PHP цілком достатня для створення цілком серйозних web-додатків.

Головним чинником мови PHP є практичність. PHP надає програмісту інструмент для швидкого і ефективного вирішення поставлених завдань. Практичний характер PHP обумовлений п'ятьма важливими характеристиками:

- традиційністю
- простотою
- ефективністю
- безпекою
- гнучкістю

PHP надає в розпорядження розробників і адміністраторів гнучкі та ефективні засоби безпеки, які умовно діляться на дві категорії: засоби системного рівня і засоби рівня програми .

Оскільки PHP є вбудованою мовою, він відрізняється винятковою гнучкістю по відношенню до потреб розробника. Хоча PHP зазвичай рекомендується використовувати у поєднанні з HTML, він з таким ж успіхом інтегрується і в JavaScript, WML, XML та інші мови. Крім того, добре структуровані додатки PHP легко розширюються в міру необхідності.

Немає проблем і з залежністю від браузерів, оскільки перед відправкою клієнту сценарії PHP повністю компілюються на стороні сервера. По суті, сценарії PHP можуть передаватися будь-яких пристроїв з браузерами, включаючи стільникові телефони, електронні записні книжки, пейджери і портативні комп'ютери, не кажучи вже про традиційні ПК.

Існує ще одна «характеристика», яка робить PHP особливо привабливим: він розповсюджується безкоштовно, до того ж, з відкритими вихідними кодами.

Стратегія Open Source, і розповсюдження початкових текстів програм в масах, зробило безсумнівно позитивний вплив на багато проектів, в першу чергу – Linux. Сказане відноситься і до історії створення PHP, оскільки підтримка користувачів з усього світу виявилася дуже важливим чинником в розвитку проекту PHP.

Прийняття стратегії Open Source і безкоштовне розповсюдження початкових текстів PHP надало неоціниму послугу користувачам. До того ж,

чуйне співтовариство користувачів PHP є свого роду «колективної службою підтримки», і в популярних електронних конференціях можна знайти відповіді навіть на найскладніші питання

Для написання даної програмної web-системи як вже було сказано вище була обрана мова програмування PHP, а системою управління бази даних MySQL Server. Взаємодія сценарію на мові PHP з MySQL можна зобразити схемою :

Сценарій PHP -> DBI -> DBD MySQL -> Сервер MySQL.

Таке поєднання забезпечує єдиний інтерфейс взаємодії з різними системами управління базами даних. А DBD пов'язує цей інтерфейс безпосередньо з конкретною СУБД. Для того, щоб програма на PHP могла спілкуватися і працювати з базою даних, необхідно підключити модуль DBI і драйвер DBD.

Розглянемо декілька реалізованих методів.

Метод «avtorisation» який відповідає за процес авторизації адміністратора в системі. Отримавши на вхід дані користувача, а саме логін та пароль, функція перевіряє його наявність в базі даних, якщо дані збігаються, здійснюється вхід в системі, в протилежному випадку виводиться повідомлення про невірне введення пароль або логін. Код цього методі поданий нижче:

```
function avtorisation ()
{
    $sek {
        $login = $_POST['login'];
        $password = $_POST['password'];

        session_start();
        $a = $_SESSION['a'];
        $x = strlen($_POST['password']);
        $rez_func = (float) $_POST['rez_func'];
        $rez = mysql_query("SELECT * FROM user WHERE login='".$_login.'");
        $row = mysql_fetch_assoc($rez); or = array();
        if ($_POST['login'] != "" && $_POST['password'] != "" && $_POST['r_f'] != "")
        if (mysql_num_rows($rez) == 1)
        {
            if (md5(md5($password).$row['salt']) == $row['password'])
            {
                if ($aaa == $rez_func)
                {
                    setcookie ("login", $row['login'], time() + 70000, '/');
                    setcookie ("password", md5($row['login'].$row['password']), time() + 7000, '/');
                    $_SESSION['id'] = $row['id'];
                }
            }
        }
    }
}
```

```

        $_SESSION['login'] = $row['login'];
        $sid = $row['id'];
        $dtm = time();
else
    {
        $error[] = "Невірний логін та пароль";
        return $error;
    }
}

```

Метод «add_sites» здійснює запис даних про відомих виробників програмного забезпечення.

```

function add_sites ($action)
{
    $action1 = "";
    if ($action == "regg") {$action1 = "Додати виробника програмного забезпечення "; $ins =
mysql_query("INSERT INTO sites (name, length) VALUES ('".$name."','".$length."");"}

```

Код програмної системи подано у додатку А.

3.2. Програмна реалізація бази даних

У якості системи керування базами даних (далі - СКБД) була обрана MySQL. Встановлення даної СКБД не вимагає будь-яких налаштувань, оскільки сервер, на якому встановлений пакет LAMP, автоматично передбачає зв'язок між елементами Apache, PHP та MySQL за допомогою спеціальних розширень. Для того, щоб працювати з базою даних засобами PHP, необхідно попередньо її створити. Зробити це можна або за допомогою утиліти mysql командного інтерпретатора, або за допомогою веб-інтерфейсів, використовуючи програмний засіб phpMyAdmin [13], що входить до пакету LAMP. При реалізації проекту даної дипломної роботи використано останній спосіб.

Для створення бази даних було обрано пункт меню “Бази даних” phpMyAdmin, після чого в поле “Створити базу даних” було введено ім'я бази (рисунок 3.1).

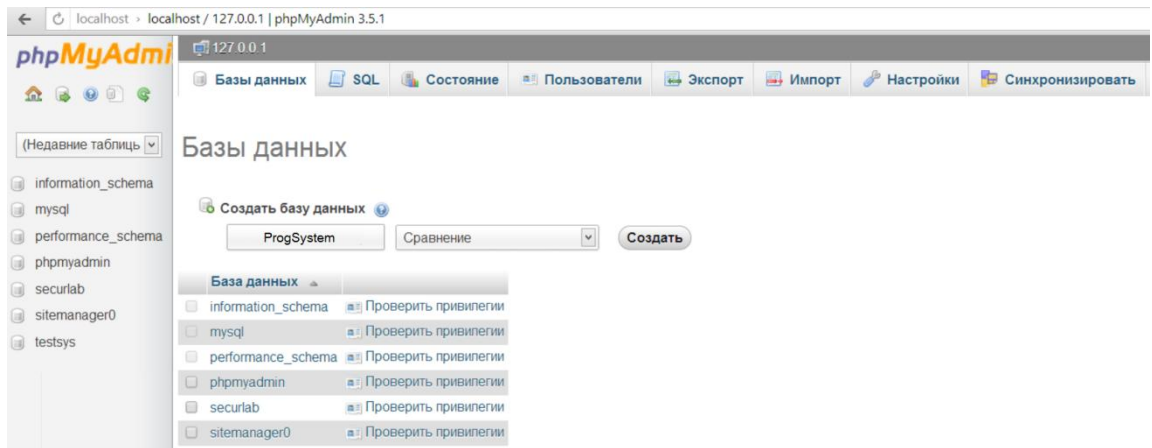


Рис. 3.1. Сторінка створення бази даних

Після вводу імені натиснено кнопку “Створити”, після чого користувача переадресує на сторінку перегляду таблиць бази даних. Для створення таблиць використано кнопку “Створити таблицю” лівої бокової панелі, після чого відкривається сторінка створення нової таблиці бази даних (рисунок 3.2).

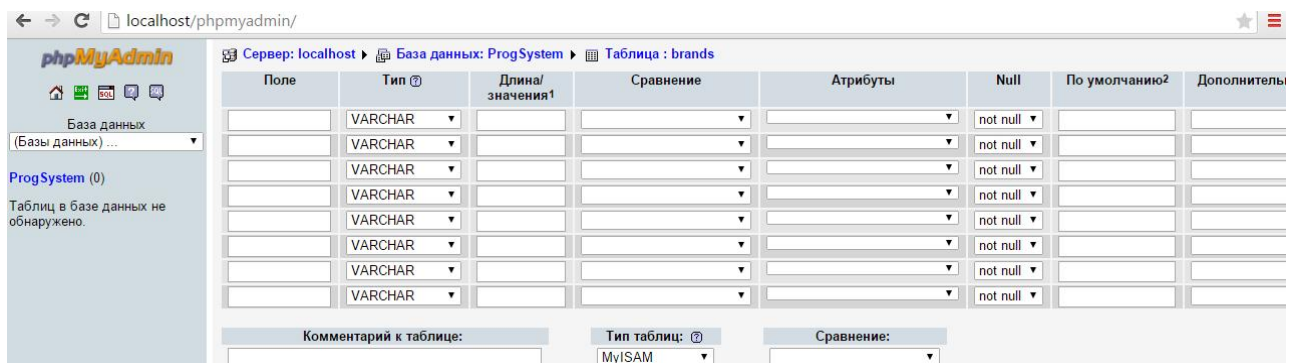


Рис. 3.2. Сторінка створення таблиці бази даних

```
CREATE TABLE IF NOT EXISTS `mydb`.`users` (
  `ID` INT NOT NULL ,
  `user_id` INT NULL ,
  `first_name` VARCHAR(255) NULL ,
  `last_name` VARCHAR(255) NULL ,
  `middle_name` VARCHAR(255) NULL ,
  `nick_name` VARCHAR(255) NULL ,
  `avatar` VARCHAR(255) NULL ,
  `resource_id` INT NULL ,
  PRIMARY KEY (`ID`) )
ENGINE = InnoDB
```

На сторінці створення таблиці необхідно вказати ім'я таблиці, а також заповнити інформацію про поля створюваної таблиці. Імена таблиць та їх властивості задаються відповідно даних діаграми “Суть-зв’язок”, розглянутої в другому розділі даної дипломної роботи. Для ідентифікаторів та цілочисельних полів був використаний тип `int`. Для стрічкових полів використано тип даних `varchar` та кодування `utf8_general_ci` для забезпечення підтримки мультіязичного вводу даних та вводу спеціальних символів.

Висновки до розділу 3

У даному озділі обґрунтовано технологію, мову програмування та розроблено програмну систему. Обґрунтовано засоби розробки бази даних та створено програмну реалізацію бази даних програмної системи за допомогою механізму збережених процедур.

РОЗДІЛ 4

ТЕСТУВАННЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ

4.1. Тестування

Існує кілька методів тестування програмного забезпечення, розглянемо і зробимо їх аналіз, для вибору найбільш оптимального. Ручне тестування - сутність методу полягає в перегляді вихідних текстів програми і правильності алгоритму роботи. Для великого обсягу вихідного коду і через його складність використання цього методу не дуже ефективно.

Автоматизоване тестування, яке діляться на метод «білого ящика» і метод «чорного ящика».

Метод «білого ящика» - фактично перевіряє відповідність програми розробленим раніше алгоритмам. Цей метод не виявляє алгоритмічні помилки і тому використовується тільки як доповнення до методу «чорного ящика». Вихідною інформацією для цього методу є граф-схема алгоритму, яка потім перетвориться в граф управління. Далі розробляється набір текстів, з тим або іншим ступенем повноти покривають множинк дуг або маршрутів в графі управління. Оскільки для складної програмної системи керуючий граф також виходить вельми складним, областю застосування цього методу фактично є модулі і невеликі підсистеми.

Метод «чорного ящика» - реалізує тестування програмного комплексу по входу/виходу. Метод перевіряє відповідність функціонування системи початкової специфікації на програмний засію. Він може виявляти помилки будь-яких класів і придатний на практиці: для тестування великих систем.

Тестування системи клієнта. Перехід по пункту меню

1. Натискаємо на пункт меню (Просування сайту і на головну).
2. У середині сторінки відображається відповідне значення, а саме: посилання на сайт та запити розкритки, та інформація головної сторінки.

3 . Всі перелічені пункти виконуються.

Запити для розкручування.

1. Вводимо параметр пошуку, наприклад, факультет.

2. Натискаємо додати.

3. Відкривається вікно з результатом, де відображений список запитів, у яких містяться всі запити для розкручування.

4. Всі параметри відображаються, тобто тест виконується.

Авторизація користувачів.

1. Натискаємо вхід

2. Введення логіна і пароля.

3. Натискання кнопку Вхід.

4. Потрапляємо у відповідне для користувача місце.

5. Виконання всіх пунктів.

Перехід вибору регіону просування порталу.

1. Натискаємо кнопку переходу в регіон.

2. Вибираємо регіон.

3. Вибираємо місто.

4. Вибираємо мову.

5. Всі значення додані - виконується .

У даній системі всі проведені функціональні тести виконані успішно, тобто всі вимоги були виконані.

Крім функціонального тестування також було проведено нагрузочне тестування, яке застосовується для аналізу роботи інформаційних систем на різних рівнях навантаження. Основним поняттям нагрузочного тестування є "віртуальний користувач". Керуючи числом віртуальних користувачів, тестувальник управляє навантаженням на систему. Віртуальний користувач виконує типові операції в системі шляхом відтворення трафіку, який відправляється клієнтським додатком на сервер. Іншими словами, віртуальний користувач виконує скрипти, які посилають на сервер пакети у форматі чинного протоколу, наприклад, http, odbs, NCA та ін.

Основні показники продуктивності інформаційної системи, які вимірюються в ході тестування навантаження:

- час відгуку (час виконання операції);
- число операцій виконуваних в одиницю часу (один з варіантів: TPS - transactions per second).

Одним з термінів навантажувального тестування є "крива деградації" - графік, якої показує залежність продуктивності системи (наприклад, в одиницях часу відгуку) від робочого навантаження (наприклад, від числа віртуальних користувачів).

У дипломній роботі розглянуті два способи проведення навантажувального тестування:

- за допомогою спеціалізованого веб-сервісу loadimpact.com;
- за допомогою спеціалізованого ПЗ Apache JMeter [7].

Обидва способи використовують загальний критерій виміру стійкості веб-додатків - залежність часу відгуку системи від кількості одночасно працюючих з сайтом користувачів.



Веб-сервіс loadimpact.com у безкоштовній версії здійснює множинні запити тільки на головну сторінку сайту, при цьому можливо тільки кілька способів візуалізації результатів тестування. Використання loadimpact.com виправдано у випадку необхідності проведення навантажувального тесту для проекту з низькою та середньою відвідуваністю і в разі відсутності часу на налаштування і проведення повноцінних нагрузочних випробувань.



Apache JMeter є професійним програмним комплексом проведення нагрузочного тестування та пропонує велику кількість інструментів для

здійснення практично будь-яких сценаріїв проходження тесту. Також присутня велика кількість способів агрегації і візуалізації отриманих в ході тестування даних.

Результати тестування. У безкоштовної версії loadimpact.com проводиться 5 тестів при навантаженні на сервер еквівалентий одночасній роботі 10, 20, 30, 40 або 50 користувачів. На рисунку 4.1 зображені результати проходження повного циклу тесту. Результати теста вийшли наступними:

- тест проводився протягом 10 хвилин;
- середній час відгуку сайту складає:
- при 10 віртуальних користувачах - 586 мс;
- при 20 - 536 мс;
- при 30 - 541 мс;
- при 40 - 478 мс;



Рис. 4.1. Результати навантажувального тестування за допомогою loadimpact.com

Як видно, очікувана тенденція падіння продуктивності при збільшенні навантаження не виправдалася, і отримані результати виявилися прямо протилежними. Така поведінку можна пояснити наступними факторами:

1. Наявність систем кешування запитів веб-сервера і сервера СУБД. Це означає, що чим популярніша запитувана інформація, тим більша вірогідність її переносу в оперативну пам'ять сервера. У разі СУБД, одного разу отримана відповідна інформація на певний запит зберігається в кеші, і при проходженні всіх подальших запитів відбувається звернення до збереженого результату в оперативній пам'яті, а не звернення до БД.

2. Вбудована система збільшення споживаних ресурсів у веб-сервері Apache.

3. Використання «хмарних» обчислень. Всі використовувані ноди є частиною обчислювального комплексу, при необхідності на певні проміжки часу система може виділити більше встановленого ліміту ресурсів нода, тим самим покриваючи дефіцит продуктивності сервера.

В основі JMeter лежить система тестових планів. У кожного плану може бути декілька сценаріїв проведення тестування. На рисунку 4.2 наведено приклад інтерфейсу JMeter.

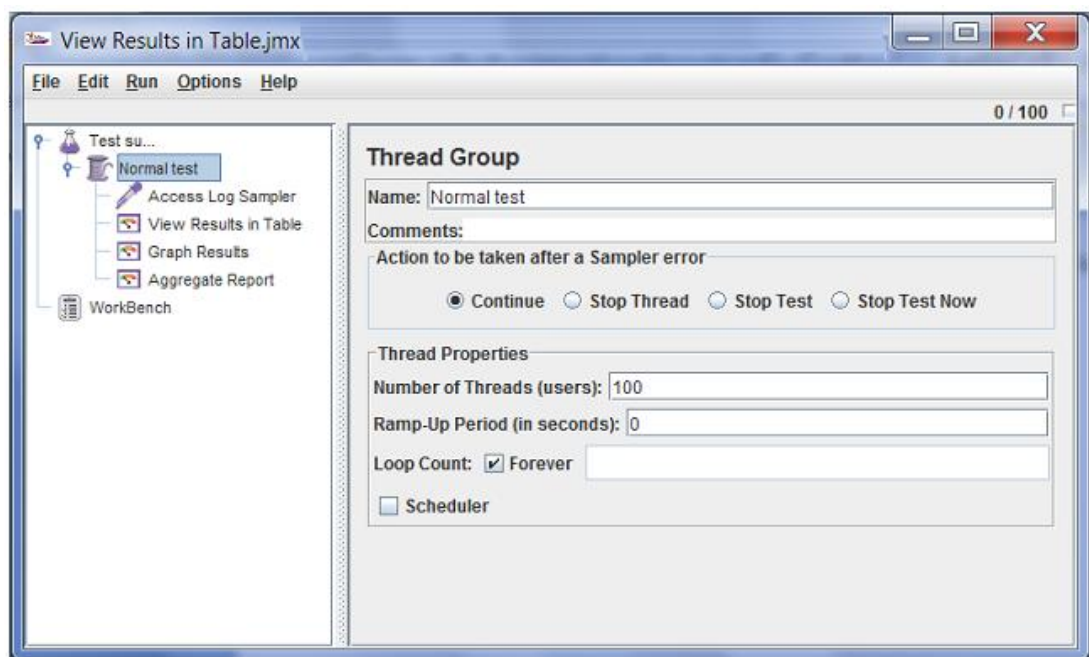


Рис. 4.2. Інтерфейс програми JMeter

Починаємо ми з створення нового тестового плану і додавання до нього сценарію. Найбільш правдоподібний сценарій користувальницьких дій не обов'язково писати самим - можна скористатися логами відвідування сайту, який веде веб-сервер. Ці логи, як правило, знаходяться в папці користувача.

Імпортувавши файл з логами в програму ми отримали сценарій проходження користувацьких запитів для одного клієнта. Далі, необхідно додати способи збору даних. Основними будуть:

- View Results in Table - перегляд результатів проходження всіх запитів;
- Graph Results - графічне відображення результатів проходження запитів;
- Aggregate Report - оброблена інформація по всіх запитах.

План тесту наступний:

1. Створюється 200 віртуальних користувачів, які рівномірно будуть додаватися протягом 10 секунд;
2. Сценарій запитів складається з 50 різних запитів, взятих з логів веб-серверів (тобто запити будуть максимально близькі до природних);
3. Кожен користувач повинен здійснити 2500 запитів (50 циклів по 50 запитів) на сервер при цьому між кожним запитом буде здійснюватися випадкова затримка в рамках 1000 мілісекунд.

При одночасній роботі 180 клієнтів (3480 запитів в хвилину) сервер завис і не відповідав на запити, що позначилося на результатах тесту, в яких сильно були завищені середні показники. Тому тест був зупинений, і віддалено був перезавантажений вебсервер.

В силу того, що сервер перестав відповідати на запити, повноцінний тест не був закінчений.

Щоб виміряти робочі показники сервера при високому навантаженні змінимо сценарій проходження тесту:

1. Створюється 100 віртуальних користувачів, які рівномірно будуть додаватися протягом 10 секунд;
2. Сценарій запитів складається з 50 різних запитів, взятих з логів веб-серверів (тобто запити будуть максимально близькі до природних);

3. Кожен користувач повинен здійснити 600 запитів (20 циклів по 50 запитів) на сервер при цьому між кожним запитом буде здійснюватися випадкова затримка в рамках 1000 мілісекунд.

Тобто зменшимо в два рази кількість користувачів, збільшимо швидкість їх появи і зменшимо кількість запитів.

Результати тестування наведені на рисунку 4.3.

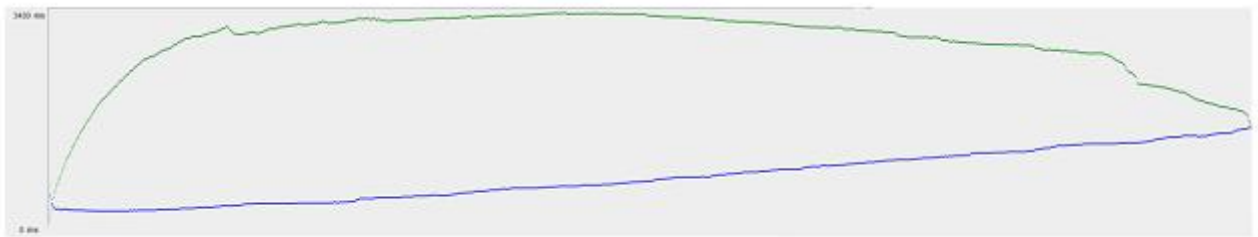


Рис. 4.3. Графік з результатами навантажувального тестування JMeter

Верхня крива означає пропускну здатність, виражену в запитах на хвилину. Спочатку, коли вводяться всі 100 потоків, пропускну здатність різко зростає. Потім, кожен з потоків виконує свої 600 запитів, у цей момент пропускну здатність ність доходить до 3500 запитів на хвилину. Далі, деякі потоки завершують свою роботу і пропускну спроможність падає.

Верхня межа графіка - 3430 мс. Нижня крива позначає середній час відгуку сайту на запити, що надходять. Крива плавно зростає до максимуму в 1.5 секунди. Інтерпретуючи отримані дані, робимо висновок, що сервіс може працювати при одночасній роботі 100 користувачів, які в сумі породжують до 3500 запитів в хвилину, отже, в день можливо обробити до 5 мільйонів запитів день.

Насправді кожен клієнт в середньому породжує 50 запитів, нерівномірно розподілених протягом 5-10 хвилин, це означає, що сервер витримає відвідуваність в 6500 користувачів в день.

Дані по середньому числу запитів від користувача і по середньому перебуванню користувача на сайта взяті з статистики відвідуваності сайту, зібраної за 3 місяці роботи системи.

Дані отримані в тестуванні свідчать про стабільну роботу сервера. Навіть при великих навантаженнях не відбувається різкого збільшення часу відгуку системи.

4.2. Розгортання програмного продукту

Для функціонування розробленого програмного продукту необхідні наступні апаратні та програмні засоби на серверній стороні:

Система: Linux;

СУБД MySQL версії не нижче 5.1;

PHP версії не нижче 5.3;

Apache 2.x.

Мінімальні вимоги до апаратного забезпечення:

Процесор: 1,0 ГГц;

Оперативна пам'ять: 128 Мбайт;

Відеокарта: ATI Radeon 9400 128 Мбайт;

Жорсткий диск: 4 Гбайт вільного простору;

Мережеве підключення: 512 Кбіт/с.

Рекомендовані вимоги до апаратного забезпечення:

Процесор: Intel pentium 4 2,4 ГГц;

Оперативна пам'ять: 512 Мбайт;

Відеокарта: ATI Radeon 9550 128 Мбайт;

Жорсткий диск: 4 Гбайт вільного простору;

Мережеве підключення: 1 Мбіт/с.

Для того, щоб інсталювати програмний засіб, необхідно виконати наступні кроки:

1. Створити директорію для веб-проекту на сервері.
2. Скопіювати всі файли проекту до директорії веб-проекту.
3. Використовуючи phpMyAdmin, здійснити імпорт файлу дампу бази даних. Файл має ім'я prog_shop.sql і знаходиться в кореневій директорії проекту.
4. Після імпорту файлу бази даних необхідно видалити файл bakalavrsys.sql.
5. Перейти в директорію includes проекту, в файлі conf.php задати параметри підключення до бази даних.

Система передбачає можливість конфігурування підключення бази даних в двох варіаціях: конфігурування підключення на сервері для розробки та тестуванні, та конфігурування підключення до бази даних на production-сервері. Для того, щоб визначити тип сервера, необхідно для константи DEBUG в файлі conf.php задати значення:

- true, за умови, якщо проект розгортається на сервері розробки;
- false, за умови, якщо проект розгортається на production-сервері.

Після встановлення типу сервера, необхідно задати параметри для підключення до бази даних:

- змінна \$dbname визначає ім'я бази даних;
- змінна \$dbuser визначає ім'я користувача бази даних;
- змінна \$dbpasswd визначає пароль для вказаного користувача бази.

Для роботи з системою підтримки просування інформаційного порталу по пошукових системах користувач повинен попередньо встановити один із наступних браузерів:

- Internet Explorer версії не нижче 11;
- Opera версії не нижче 12;
- Mozilla Firefox версії не нижче 29;
- Yandex Browser версії не нижче 14;
- Safari версії не нижче 7.

Для того, щоб користуватись системою, користувач повинен попередньо звернутись до адміністратора системи для створення облікового запису користувача.

4.3. Інструкція користувача

Робота із системою підтримки просування інформаційного порталу по пошукових системах починається із авторизації користувача. На рисунку 4.4 представлено форму авторизації в системі.

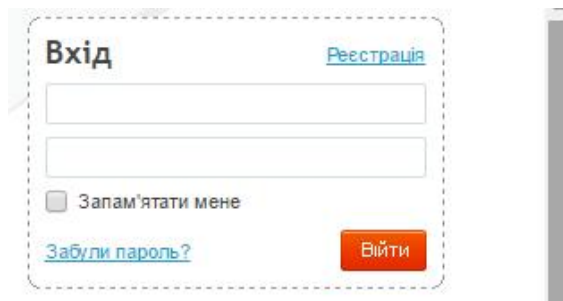
The image shows a login form titled "Вхід" (Login). It features two input fields for username and password. Below the password field is a checkbox labeled "Запам'ятати мене" (Remember me). At the bottom left, there is a link "Забули пароль?" (Forgot password?). At the bottom right, there is a red button labeled "Війти" (Login). A blue link "Реєстрація" (Registration) is located at the top right of the form area.

Рис. 4.4. Сторінка авторизації

Якщо користувач ще не зареєструвався в системі, то у нього є можливість це зробити за допомогою відповідної форми.

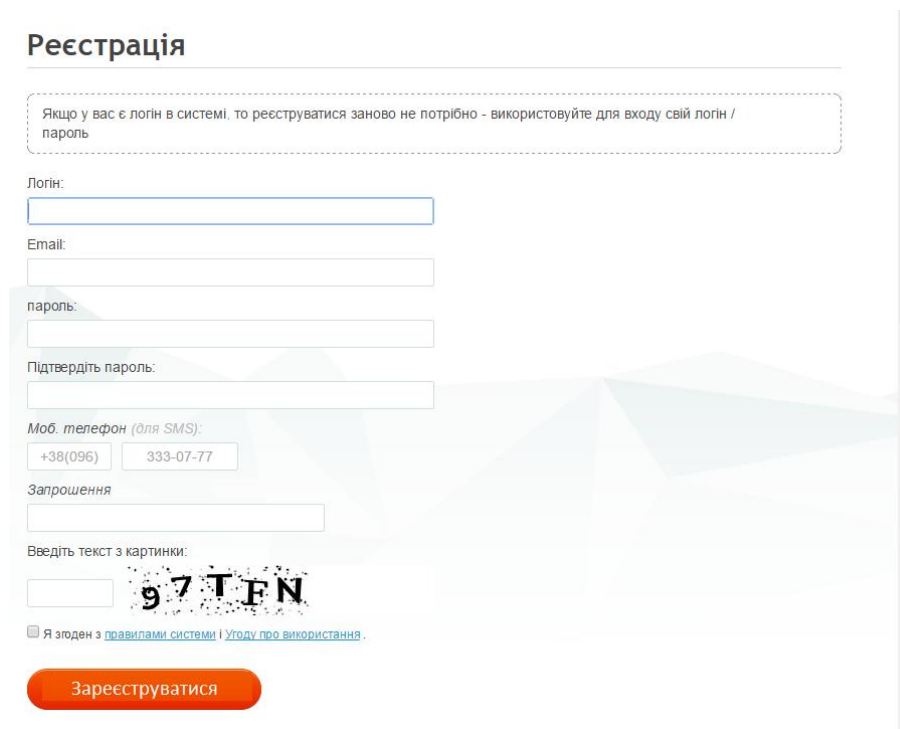

The image shows a registration form titled "Реєстрація" (Registration). It includes a message: "Якщо у вас є логін в системі, то реєструватися знову не потрібно - використовуйте для входу свій логін / пароль" (If you already have a login in the system, you do not need to register again - use your login / password for login). The form contains several input fields: "Логін:" (Login), "Email:", "пароль:" (password), "Підтвердіть пароль:" (Confirm password), "Моб. телефон (для SMS):" (Mobile phone (for SMS)) with a dropdown for country code (+38(096)) and a text field for the number (333-07-77), "Запрошення" (Invitation code), and "Введіть текст з картинки:" (Enter text from the image) with a CAPTCHA image showing "97TFN". At the bottom, there is a checkbox "Я згоден з правилами системи і Угоду про використання" (I agree with the system rules and the Terms of Use) and a red button "Зареєструватися" (Register).

Рис. 4.5. Сторінка реєстрації

Після успішної авторизації, у користувача додається можливість для додавання власного сайту для просування (рисунок 4.6).



ВІДПРАВИТИ ЗАПИТ

Ваше ім'я: (*)

Адрес сайту:

Телефон: (*)

E-mail: (*)

SEO просування


Коментар

Перевірка на людяність. Скільки буде два плюс три:

Відправити

Рис. 4.6. Форма для додавання власного сайту для просування

На рисунку 4.7 представлено сторінку для задавання початкових параметрів для просування власного інформаційного порталу.



1 Просуваємо сайт

http://

2 Регіон

Google Яндекс

Україна Тернопіль

на українській мові

3 Просуваємо запити

Факультет комп'ютерних інформаційних технологій

ДОБАВИТИ **ПІДБРАТИ**

4

Регіон	Запит	Сторінка
1	Україна	Факультет комп'ютерних інформаційних технологій
	262.33	fci.tne.u.edu.ua

бюджет: \$262,33

Просунути сайт Ресстрація

ОЧИСТИТИ

Рис. 4.7. Сторінка формування початкових вимог для просування порталу

Висновки до розділу 4

Здійснено опис процедур тестування та їхніх результатів, описані тест-вимоги до програмного забезпечення, а також виявлені дефекти. Розкрито питання встановлення та налаштування програмного забезпечення на сервері, а також вказані вимоги, дотримання яких необхідно для користування системою, описана інструкція користувача для роботи із системою.

ВИСНОВКИ

Запропоновано принципи побудови системи аналізу та просування інтернет-ресурсів. Спроектвані контексні діаграми та приведена їх декомпозиція надає необхідний апарат із побудови зазначеної системи, що дасть змогу здійснювати підтримку прийняття рішень у галузі оптимізації інтернет-ресурсів. Зазначений підхід дасть змогу: застосовувати нові адаптивні алгоритми визначення релевантності ресурсу; використовувати в створеній системі інтерактивні елементи; спростити адміністрування інтернет-ресурсів; підвищити інформативність та конкурентоздатність ресурсу. Подальші дослідження будуть скеровані на вивчення методів організації та проектування кожної з компонент системи аналізу та просування інтернет-ресурсів.

В ході виконання бакалаврської роботи частково реалізована підсистема підтримки просування інформаційного порталу по пошукових системах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Федорова Д.Э., Семенов Ю.Д., Чижик К.Н. CASE-технологии. - М.: Горячая линия Телеком, Радио и связь, 2005. – 160 с.
2. Дин Леффингуэлл, Дон Уидриг. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. - М.: Вильямс, 2012. – 448 с.
3. Gurdy Leete, Mary Leete. Microsoft Expression Blend Bible. - John Wiley & Sons, 2007 - 734 p.
4. Тарасов О. В., Федько В. В, Лосев М. Ю. Організація баз даних та знань. Проектування баз даних : навч.-практ. Посіб. - Х. : ХНЕУ, 2011. – 200 с.
5. Берко А. Ю., Верес О. М., Пасічник В. В. Системи баз даних і знань. Книга 1. Організація баз даних і знань : навч. посіб. Львів : Магнолія, 2011. – 456 с.
6. Э. Вигли., Д. Мот., П. Фут. Microsoft Mobile и .Net Compact Framework. Руководство разработчика. - СПб. : Питер, 2009. – 699 с.
7. К. Пейтон. PHP 5 и MySQL. - М. : Бином-Пресс, 2011. - 366 с.
8. Білас О. Є. Якість програмного забезпечення та тестування : навч. посібн. - Львів : Видавництво Львівської політехніки, 2013. - 216 с.
9. Кузнецов М. В., Симдянов И. В., Гольшев С. В. PHP 5. Практика разработки Web-сайтов (СПб.: БХВ-Петербург: 2005).
10. Пелещишин А.М., Буров Є.В. Принципи формального моделювання інформаційних систем на базі Web-технологій (Львів: НУ "Львівська Політехніка": 1999).
11. Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. Алгоритмы: построение и анализ. - 2-е изд. - М.: «Вильямс», 2006. - 1296 с.
12. Кен Хендерсон Професійне керівництво з SQL Server: структура та реалізація. — М.: Издательский дом «Вильямс», 2006. — С. 1056. ISBN 5-8459-0912-0

13. Чураков Михаил. Муравьиные алгоритмы [Электронный ресурс] / Михаил Чураков, Андрей Якушев. // Режим доступа: <http://rain.ifmo.ru/cat/data/theory/unordered/ant-algo-2006/article.pdf>.
14. Бормашов Д. А. “Кластерный анализ текстов”: Дипломная работа [Электронный ресурс] / Д. А. Бормашов. Режим доступа: <http://inf.tsu.ru/library/DiplomaWorks/CompScience/2006/bormashov/diplom.pdf>.
15. Чубукова И.А. Data Mining БИНОМ. Лаборатория знаний, Интернет-университет информационных технологий - ИНТУИТ.ру, 2006.
16. Дюк В.А., Самойленко А.П. Data Mining: учебный курс. – СПб.: Питер, 2001.
17. Барсегян А. А., Куприянов М. С., Степаненко В. В., Холод И. И. Методы и модели анализа данных: OLAP и Data Mining. – СПб.: БХВ-Петербург, 2004. – 336 с.
18. Дивак М. П., Шпінталь М.Я., Козак О.Л., Струбицька І.П., Спільчук В.М., Піговський Ю.Р. Методичні рекомендації до виконання дипломної роботи освітньо кваліфікаційного рівня «бакалавр» студентам усіх форм навчання для напряму підготовки 6.050103 – «Програмна інженерія» // Тернопіль : ФОП Шпак П.П. - 2014. - 54 с.

ДОДАТОК А

ЛІСТИНГ ОСНОВНИХ МОДУЛІВ СИСТЕМИ

```
// Модуль головної сторінки системи просування сайтів
```

```
include_once("includes/sp-load.php");
checkLoggedIn();
include_once(SP_CTRLPATH."/siteauditor.ctrl.php");
$controller = New SiteAuditorController();
$controller->view->menu = 'seotools';
$controller->layout = 'ajax';
$controller->set('spTextTools', $controller->getLanguageTexts('seotools',
$_SESSION['lang_code']));
$controller->set('spTextPanel', $controller->getLanguageTexts('panel', $_SESSION['lang_code']));
$controller->spTextSA = $controller->getLanguageTexts('siteauditor', $_SESSION['lang_code']);
$controller->set('spTextSA', $controller->spTextSA);

if($_SERVER['REQUEST_METHOD'] == 'POST'){

    switch($_POST['sec']){

        case "create":
            $controller->set('spTextSettings', $controller->getLanguageTexts('settings',
$_SESSION['lang_code']));
            $controller->createProject($_POST);
            break;

        case "update":
            $controller->set('spTextSettings', $controller->getLanguageTexts('settings',
$_SESSION['lang_code']));
            $controller->updateProject($_POST);
            break;

        case "activateall":
            if (!empty($_POST['ids'])) {
                foreach($_POST['ids'] as $id) {
                    $controller->__changeStatus($id, 1);
                }
            }
            $controller->showAuditorProjects($_POST);
            break;

        case "inactivateall":
            if (!empty($_POST['ids'])) {
                foreach($_POST['ids'] as $id) {
                    $controller->__changeStatus($id, 0);
                }
            }
            $controller->showAuditorProjects($_POST);
            break;
    }
}
```



```

    case "deleteall":
        if (!empty($_POST['ids'])) {
            foreach($_POST['ids'] as $id) {
                $controller->__deleteProject($id);
            }
        }
        $controller->showAuditorProjects($_POST);
        break;

    case "showreport":
        $controller->showProjectReport($_POST);
        break;

    case "importlinks":
        $controller->importLinks($_POST);
        break;

    default:
        $controller->showAuditorProjects($_POST);
        break;
}

}else{
    switch($_GET['sec']){

        case "Activate":
            $controller->__changeStatus($_GET['project_id'], 1);
            $controller->showAuditorProjects($_GET);
            break;

        case "Inactivate":
            $controller->__changeStatus($_GET['project_id'], 0);
            $controller->showAuditorProjects($_GET);
            break;

        case "delete":
            $controller->__deleteProject($_GET['project_id']);
            $controller->showAuditorProjects($_GET);
            break;

        case "new":
            $controller->set('spTextSettings', $controller->getLanguageTexts('settings',
$_SESSION['lang_code']));
            $controller->newProject($_GET);
            break;

        case "edit":
            $controller->set('spTextSettings', $controller->getLanguageTexts('settings',
$_SESSION['lang_code']));
            $controller->editProject($_GET['project_id']);
            break;
    }
}

```

```

case "showrunproject":
    $controller->showRunProject($_GET['project_id']);
    break;

case "runproject":
    $controller->runProject($_GET['project_id']);
    break;

case "viewreports":
    $controller->viewReports($_GET);
    break;

case "pagedetails":
    $controller->viewPageDetails($_GET);
    break;

case "recheckreport":
    $controller->recheckReportPages($_GET['project_id']);
    $controller->showRunProject($_GET['project_id']);
    break;

case "showreport":
    $controller->showProjectReport($_GET);
    break;

case "checkscore":
    $controller->checkPageScore($_GET);
    break;

case "deletepage":
    $controller->__deleteReportPage($_GET['report_id']);
    $controller->loadReportsPage($_GET);
    break;

case "croncommand":
    $controller->showCronCommand();
    break;

case "showsettings":
    $settingCtrler = $controller->createController('Settings');
    $settingCtrler->set('spTextPanel', $controller->getLanguageTexts('panel',
$_SESSION['lang_code']));
    $settingCtrler->spTextSettings = $controller->getLanguageTexts('settings',
$_SESSION['lang_code']);
    $settingCtrler->set('spTextSettings', $settingCtrler->spTextSettings);
    $settingCtrler->set('headLabel', $controller->spTextSA['Site Auditor Settings']);
    $settingCtrler->showSystemSettings('siteauditor');
    break;

case "importlinks":
    $controller->showImportProjectLinks($_GET);
    break;

```

```

        default:
            $controller->showAuditorProjects($_GET);
            break;
    }
}

?>
include_once("includes/sp-load.php");
checkLoggedIn();
include_once(SP_CTRLPATH."/website.ctrl.php");
include_once(SP_CTRLPATH."/keyword.ctrl.php");
include_once(SP_CTRLPATH."/user-type.ctrl.php");
$controller = New WebsiteController();
$controller->view->menu = 'seotools';
$controller->layout = 'ajax';
$controller->spTextPanel = $controller->getLanguageTexts('panel', $_SESSION['lang_code']);
$controller->set('spTextPanel', $controller->spTextPanel);
$controller->spTextWeb = $controller->getLanguageTexts('website', $_SESSION['lang_code']);
$controller->set('spTextWeb', $controller->spTextWeb);

if($_SERVER['REQUEST_METHOD'] == 'POST'){

    switch($_POST['sec']){
        case "create":
            $controller->createWebsite($_POST);
            break;

        case "update":
            $controller->updateWebsite($_POST);
            break;

        case "activateall":
            if (!empty($_POST['ids'])) {
                foreach($_POST['ids'] as $id) {
                    $controller->__changeStatus($id, 1);
                }
            }
            $controller->listWebsites($_POST);
            break;

        case "inactivateall":
            if (!empty($_POST['ids'])) {
                foreach($_POST['ids'] as $id) {
                    $controller->__changeStatus($id, 0);
                }
            }
            $controller->listWebsites($_POST);
            break;

        case "deleteall":
            if (!empty($_POST['ids'])) {

```

```

        foreach($_POST['ids'] as $id) {
            $controller->__deleteWebsite($id);
        }
    }

    $controller->listWebsites($_POST);
    break;

case "crawlmeta":
    $keyInput = empty($_POST['keyinput']) ? "" : $_POST['keyinput'];
    $controller->crawlMetaData(urldecode($_POST['url']), $keyInput);
    break;

case "import":
    $controller->importWebsiteFromCsv($_POST);
    break;

default:
    $controller->listWebsites($_POST);
    break;
}

}else{
    switch($_GET['sec']){

        case "Activate":
            $controller->__changeStatus($_GET['websiteId'], 1);
            $controller->listWebsites($_GET);
            break;

        case "Inactivate":
            $controller->__changeStatus($_GET['websiteId'], 0);
            $controller->listWebsites($_GET);
            break;

        case "delete":
            $controller->__deleteWebsite($_GET['websiteId']);
            $controller->listWebsites($_GET);
            break;

        case "edit":
            $controller->editWebsite($_GET['websiteId']);
            break;

        case "new":
        case "newweb":
            $controller->newWebsite($_GET);
            break;

        case "import":
            $controller->showImportWebsites($_GET);
            break;
    }
}

```

```

        case "crawlmeta":
            $keyInput = empty($_GET['keyinput']) ? "" : $_GET['keyinput'];
            $controller->crawlMeta(urldecode($_GET['url']), $keyInput);
            break;

        case "updateurl":
            $websiteId = intval($_GET['website_id']);
            $websiteInfo = $controller->__getWebsiteInfo($websiteId);
            print '<input type="hidden" name="weburl" id="weburl"
value="'. $websiteInfo['url']. "'>';
            break;

        default:
            $controller->listWebsites($_GET);
            break;
    }
}
?>

// Модуль Users
include_once("includes/sp-load.php");

// check the sections can accessed by user
$userIncludeList = array("my-profile", "myprofile", "edit-profile", "renew-profile",
"updatemyprofile", "update-subscription");
if ( in_array($_GET['sec'], $userIncludeList) || in_array($_POST['sec'], $userIncludeList) ) {
    isLoggedIn();
} else {
    checkAdminLoggedIn();
}

include_once(SP_CTRLPATH."/user.ctrl.php");
include_once(SP_CTRLPATH."/website.ctrl.php");
include_once(SP_CTRLPATH."/keyword.ctrl.php");
include_once(SP_CTRLPATH."/user-type.ctrl.php");
include_once(SP_CTRLPATH."/seoplugins.ctrl.php");
$controller = New UserController();
$controller->view->menu = 'users';
$controller->layout = 'ajax';
$controller->spTextPanel = $controller->getLanguageTexts('panel', $_SESSION['lang_code']);
$controller->set('spTextPanel', $controller->spTextPanel);
$controller->spTextUser = $controller->getLanguageTexts('user', $_SESSION['lang_code']);
$controller->set('spTextUser', $controller->spTextUser);

if($_SERVER['REQUEST_METHOD'] == 'POST'){

    switch($_POST['sec']){
        case "create":
            $controller->createUser($_POST);
            break;

        case "update":

```

```

        $controller->updateUser($_POST);
        break;

    case "updatemyprofile":
        $controller->updateMyProfile($_POST);
        break;

    case "update-subscription":
        $controller->updateSubscription($_POST);
        break;

    case "activateall":
        if (!empty($_POST['ids'])) {
            foreach($_POST['ids'] as $id) {
                $controller->__changeStatus($id, 1);
            }
        }
        $controller->listUsers($_POST);
        break;

    case "inactivateall":
        if (!empty($_POST['ids'])) {
            foreach($_POST['ids'] as $id) {
                $controller->__changeStatus($id, 0);
            }
        }
        $controller->listUsers($_POST);
        break;

    case "deleteall":
        if (!empty($_POST['ids'])) {
            foreach($_POST['ids'] as $id) {
                $controller->__deleteUser($id);
            }
        }
        $controller->listUsers($_POST);
        break;

    default:
        $controller->listUsers($_POST);
        break;
}

}else{
    switch($_GET['sec']){

        case "Activate":
            $controller->__changeStatus($_GET['userId'], 1);
            $controller->listUsers($_GET);
            break;

        case "Inactivate":

```

```

        $controller->__changeStatus($_GET['userId'], 0);
        $controller->listUsers($_GET);
        break;

    case "delete":
        $controller->__deleteUser($_GET['userId']);
        $controller->listUsers($_GET);
        break;

    case "edit":
        $controller->editUser($_GET['userId']);
        break;

    case "new":
        $controller->newUser();
        break;

    case "my-profile":
    case "myprofile":
        $controller->showMyProfile($_GET);
        break;

    case "edit-profile":
        $controller->editMyProfile();
        break;

    case "renew-profile":
        $controller->renewMyProfile();
        break;

    default:
        $controller->listUsers($_GET);
        break;
    }
}

if(!empty($_SERVER['REQUEST_METHOD'])){

    # to section to generate report from admin area
    include_once("includes/sp-load.php");
    checkAdminLoggedIn();
    include_once(SP_CTRLPATH."/cron.ctrl.php");
    $controller = New CronController();
    $controller->timeStamp = mktime(0, 0, 0, date('m'), date('d'), date('Y'));
    $controller->set('spTextTools', $controller->getLanguageTexts('seotools',
$_SESSION['lang_code']));
    $controller->set('spTextPanel', $controller->getLanguageTexts('panel',
$_SESSION['lang_code']));
    $controller->spTextKeyword = $controller->getLanguageTexts('keyword',
$_SESSION['lang_code']);
    $controller->set('spTextKeyword', $controller->spTextKeyword);
}

```

```

if($_SERVER['REQUEST_METHOD'] == 'POST'){

    switch($_POST['sec']){

        case "generate":
            $controller->executeReportGenerationScript($_POST);
            break;
        }

    }else{
        switch($_GET['sec']){

            case "generate":
                $controller->routeCronJob($_GET['website_id'], $_GET['repTools']);
                break;

            case "croncommand":
                $controller->showCronCommand();
                break;

            default:
                $controller->showReportGenerationManager();
                break;
        }
    }

}

}

# the section for generate reports using system cron job
include_once("includes/sp-load.php");
include_once(SP_CTRLPATH."/cron.ctrl.php");
include_once(SP_CTRLPATH."/report.ctrl.php");
include_once(SP_CTRLPATH."/searchengine.ctrl.php");
include_once(SP_CTRLPATH."/keyword.ctrl.php");
$controller = New CronController();
$controller->timeStamp = mktime(0, 0, 0, date('m'), date('d'), date('Y'));

$includeList = array(); // the only included seo tools id
$controller->executeCron($includeList);

// delete crawl logs before 2 months
include_once(SP_CTRLPATH."/crawllog.ctrl.php");
$crawlLog = new CrawlLogController();
$crawlLog->clearCrawlLog(SP_CRAWL_LOG_CLEAR_TIME);
echo "Clearing crawl logs before " . SP_CRAWL_LOG_CLEAR_TIME . " days";

}
?>

```



```

include_once("includes/sp-load.php");
checkAdminLoggedIn();
include_once(SP_CTRLPATH."/searchengine.ctrl.php");
$controller = New SearchEngineController();
$controller->view->menu = 'se-manager';
$controller->layout = 'ajax';
$controller->spTextPanel = $controller->getLanguageTexts('panel', $_SESSION['lang_code']);
$controller->set('spTextPanel', $controller->spTextPanel);
$controller->spTextUser = $controller->getLanguageTexts('searchengine',
$_SESSION['lang_code']);
$controller->set('spTextSE', $controller->spTextUser);

if($_SERVER['REQUEST_METHOD'] == 'POST'){

    switch($_POST['sec']){

        case "activateall":
            if (!empty($_POST['ids'])) {
                foreach($_POST['ids'] as $id) {
                    $controller->__changeStatus($id, 1);
                }
            }
            $controller->listSE($_POST);
            break;

        case "inactivateall":
            if (!empty($_POST['ids'])) {
                foreach($_POST['ids'] as $id) {
                    $controller->__changeStatus($id, 0);
                }
            }
            $controller->listSE($_POST);
            break;

        case "deleteall":
            if (!empty($_POST['ids'])) {
                foreach($_POST['ids'] as $id) {
                    $controller->__deleteSearchEngine($id);
                }
            }
            $controller->listSE($_POST);
            break;

        default:
            $controller->listSE($_POST);
            break;
    }

}

}else{
    switch($_GET['sec']){

        case "Activate":

```

```
        $controller->__changeStatus($_GET['seId'], 1);
        $controller->listSE($_GET);
        break;

    case "Inactivate":
        $controller->__changeStatus($_GET['seId'], 0);
        $controller->listSE($_GET);
        break;

    case "delete":
        $controller->__deleteSearchEngine($_GET['seId']);
        $controller->listSE($_GET);
        break;

    default:
        $controller->listSE($_GET);
        break;
    }
}
?>
```