МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ**

**до виконання лабораторних робіт**

з дисципліни «Професійний практикум»

для здобувачів освітнього ступеня «бакалавр»

зі спеціальності 126 Інформаційні системи та технології

(освітньої програми «Web-технології, Web-дизайн»)

усіх форм навчання

Черкаси

2020

|  |  |
| --- | --- |
| УДК 004(07) М 54 | *Затверджено вченою радою ФІТІС,**протокол № 4 від 14.02.2020 р.,**згідно з рішенням кафедри* *інформаційних технологій проектування,**протокол № 8 від 10.01.2020 р.* |

|  |  |
| --- | --- |
| Упорядник: | Тарасенко Я. В., к.т.н. |
| Рецензент: | Рудницький В. М., д.т.н., професор |

|  |  |
| --- | --- |
| М 54 | **Методичні** рекомендації до виконання лабораторних робіт з дисципліни «Професійний практикум» для здобувачів освітнього ступеня «бакалавр» зі спеціальності № 126 Інформаційні системи та технології (освітня програма «Web-технології, Web-дизайн») усіх форм навчання [Електронний ресурс] / [Упоряд.: Я.В. Тарасенко]; М-во освіти і науки України, Черкас. держ. технол. ун-т. – Черкаси : ЧДТУ, 2020. – 26 с. – Назва з титульного екрана. |

Методичні рекомендації містять основні теоретичні відомості, практичні завдання до проведення лабораторних робіт та методичні вказівки для виконання цих завдань з дисципліни «Професійний практикум» з використанням сучасних технологій та програмного забезпечення. Особлива увага приділяється формуванню вмінь поєднувати знання з різних галузей веб-технологій для забезпечення усіх етапів життєвого циклу цілісної веб-орієнтованої системи, навичок проектування і реалізації web-орієнтованих інформаційних систем та забезпечення процесу їх подальшого функціонування, навичок роботи з проектною документацією.

Для здобувачів освітнього ступеня бакалавр спеціальності № 126 «Інформаційні системи та технології» (освітньої програми «Web-технології, Web-дизайн») усіх форм навчання.

УДК 004(07)

Виробничо-практичне

електронне видання

комбінованого використовування

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ**

**до виконання лабораторних робіт**

з дисципліни «Професійний практикум»

для здобувачів освітнього ступеня «бакалавр»

зі спеціальності 126 Інформаційні системи та технології

(освітньої програми «Web-технології, Web-дизайн»)

усіх форм навчання

Упорядник **Тарасенко** Ярослав Володимирович

*В авторській редакції.*

ЗМІСТ

[ВСТУП 4](#_Toc33192854)

[Лабораторна робота № 1.](#_Toc33192855) [Підготовка до проектування web-ресурсу. 5](#_Toc33192856)

[Лабораторна робота № 2.](#_Toc33192857) [Проектування структури web-ресурсу та алгоритмів сценаріїв 8](#_Toc33192858)

[Лабораторна робота № 3.](#_Toc33192859) [Програмування серверної частини web-ресурсу 11](#_Toc33192860)

[Лабораторна робота № 4.](#_Toc33192861) [Програмування користувацької частини web-ресурсу 14](#_Toc33192862)

[Лабораторна робота № 5.](#_Toc33192863) [Налаштування елементів дизайну та наповнення ресурсу контентом 17](#_Toc33192864)

[Лабораторна робота № 6.](#_Toc33192865) [Створення та проведення тест-кейсів, забезпечення індексації ресурсу, його підтримки і просування в мережі 21](#_Toc33192866)

[ЛІТЕРАТУРА 25](#_Toc33192867)

# ВСТУП

Навчальна дисципліна «Професійний практикум» належить до циклу дисциплін професійної підготовки і має міждисциплінарні зв’язки з такими дисциплінами, як «Web-програмування», «Алгоритмізація та програмування», «Об’єктно-орієнтоване програмування», «Якість інформаційних систем та тестування», «Електронна комерція та електронний бізнес», «Web-дизайн», «Просування та підтримка web-орієнтованих інформаційних систем», «Управління web-контентом», «Функціональне програмування».

Програма навчальної дисципліни складається з двох частин:

1. Проектування та розробка інтерактивного web-ресурсу.

2. Супровід, наповнення та дизайн.

Методичні рекомендації налічують опис процесу виконання 6 лабораторних робіт з уточненням мети роботи, коротких теоретичних відомостей та методичних вказівок з виконання роботи, ходу роботи, індивідуальних варіантів завдання та контрольних запитань.

Метою виконання лабораторних робіт є закріплення теоретичних основ проектування та реалізації програмного забезпечення для WWW, отримання навичок проектування та створення складних сайтів, розробки на мові програмування JavaScript клієнтських сценаріїв, розробки алгоритмів та програм на мові програмування PHP, програмування та використання прикладних і спеціалізованих комп’ютерних систем та середовищ з метою їх запровадження у професійній діяльності.

Лабораторні роботи спроектовані таким чином, що по завершенню курсу, здобувачі вищої освіти набувають компетенцій для поєднання різних технологічних рішень і засобів з метою забезпечення проведення усіх етапів процесу проектування та реалізації програмного забезпечення для WWW. Крім того, здобувачі набувають здатності вибору, проектування, розгортання, інтегрування, управління, адміністрування та супроводження інформаційних систем, технологій та інфокомунікацій, сервісів та інфраструктури організації, здатності управляти та користуватися сучасними інформаційно-комунікаційними системами та технологіями (у тому числі такими, що базуються на використанні Інтернет), здатності застосовувати правила оформлення проектних матеріалів інформаційних систем та технологій, знань складу та послідовності виконання проектних робіт з урахуванням вимог відповідних нормативно-правових документів для запровадження у професійній діяльності.

Крім того, матеріали методичних вказівок можуть бути використані при дипломному проектуванні.

При проектуванні лабораторних робіт використані такі новітні методи навчання, як модульне, контекстне (на ранніх етапах репродуктивний метод, на пізніх – евристичний), проблемне навчання. При цьому, проблемне інтегроване з модульним.

# Лабораторна робота № 1

# Підготовка до проектування web-ресурсу.

**Мета роботи** – ознайомитися з базовими принципами та етапами передпроектної підготовки до розробки і введення в експлуатації веб-сайту. Набути навичок оформлення технічної документації до проекту. Освоїти особливості обрання програмних та інструментальні засобів для розробки веб-ресурсу.

**Короткі теоретичні відомості**

Розробка будь-якої інформаційної системи проводиться за суворо визначеними нормами і проектування веб-сайту не виключення. Передпроектна підготовка включає в себе аналітичну діяльність, а саме аналіз конкурентних веб-сайтів, визначення мети, задач та дослідження потенційної цільової аудиторії. Важливо також враховувати вимоги замовника, не забуваючи також і про об’єктивні вимоги, які висувають стандарти якості програмного забезпечення.

По завершенню перед проектної підготовки оформлюється технічне завдання. Технічне завдання (ТЗ) є вихідним документом проектування програмного продукту (веб-сайту) відповідно до якого проводиться програмування, приймання та експлуатація інформаційної системи. Стандартів ТЗ існує досить багато, необхідно обрати актуальний дійсний вітчизняний чи зарубіжний стандарт. Однак, слід пам’ятати про суворо визначену структуру ТЗ, що в загальному випадку спільна для усіх стандартів і складається з: вступу (назва та коротка характеристика сайту), обґрунтування розробки (підстави розробки), її призначення, вимог до програмної системи (функціонал, надійність, програмна сумісність тощо), вимог до документації (правила оформлення звітності), розрахунку економічних показників (економічна ефективність), стадій та етапів розробки (з визначенням строків розробки та виконавців), порядку контролю та здачі проекту (види тестових випробувань тощо). Оформлюється ТЗ для сайтів зазвичай у довільному вигляді з урахуванням вимог стандарту. При цьому вказуються лише ті характеристики, які можливо виміряти, описується структура сайту, наявність фірмового дизайну, вказується діапазон розподільчої здатності моніторів користувачів.

Вартість розробки веб-сайту розраховується з огляду на вартість перед проектної підготовки, вартість хостінгу, розробки контенту, розмір заробітної плати програмістів, дизайнерів, вартість інформаційної та технічної підтримки, супроводу, оптимізації та просування, враховуючи рівень конкуренції серед подібних веб-сайтів та конкурентоспроможність розробленого веб-сайту.

Техніко-економічне обґрунтування проводиться на основі порівняння витрат та очікуваних результатів від впровадження веб-сайту.

Оцінка витрати на впровадження веб-сайту передбачає співвідношення сукупних інвестицій в створення веб-сайту до загальних витрат або чистого доходу.

Потенційна економічна ефективність визначається у вигляді відношення очікуваного результату застосування веб-сайту до витрат на розробку та експлуатацію системи.

При виборі системи управління веб-ресурсом зважають на простоту та зрозумілість інтерфейсу, можливості зручного адміністрування, дешевизну процесу редизайну веб-сайту (впливає на ціну) та ергономічність користувацького інтерфейсу (врахування цільової аудиторії, визначеної на етапі перед проектної підготовки). Хостінг слід обирати з огляду на збереження високої швидкості роботи ресурсу за умов високого навантаження та на підтримку сучасних версій серверної мови програмування PHP та можливість реалізації користувацьких сценаріїв мовою JavaScript.

**Завдання:**

Провести перед проектну підготовку, обрати програмні та інструментальні засоби для розробки веб-ресурсу з урахуванням вимог та оформити технічну документацію до проекту.

Хід роботи:

1. Обрати веб-ресурс згідно індивідуального завдання.
2. Провести перед проектну підготовку та оформити технічне завдання з огляду на вимоги до інформаційної системи, враховуючи правила оформлення технічної документації (примітка 1).
3. Провести розрахунки вартості проекту, техніко-економічне обґрунтування розробки веб-ресурсу та оцінити витрати.
4. Визначити потенційну економічну ефективність веб-ресурсу.
5. Обрати систему управління web-ресурсом, середовище розробки, інструментальні засоби та хостінг (примітка 2).
6. Провести розгортання та налаштування веб-серверу. Розмістити базові початкові файли на хостінгу.
7. Формування звіту та висновків.

***Примітка 1:*** *у звіті необхідно надати оформлення технічного завдання згідно стандарту вимог до оформлення технічної документації (3-5 сторінок основного тексту) з обов’язковою наявністю пунктів: вступ, обґрунтування розробки, її призначення, вимоги до програмної системи, вимоги до документації, розрахунок економічних показників, стадії та етапи розробки, порядок контролю та здачі проекту.*

***Примітка 2:*** *система управління web-ресурсом та хостінг обираються з огляду на подальшу розробку проекту мовою PHP із застосуванням JavaScript сценаріїв.*

Тематика веб-ресурсу відповідно до індивідуального варіанту (номер варіанту визначається за порядковим номером у списку групи):

1. Сайт актуальних новин.
2. Блог кіберспортсмена.
3. Веб-сайт закладу вищої освіти.
4. Інформаційний портал та консультування.
5. Веб-ресурс прогнозу погоди.

**Контрольні запитання**

1. Охарактеризуйте загальні принципи роботи сайтів та сервісів Інтернет.
2. Наведіть етапи та послідовність виконання проектних робіт по створенню складного web-ресурсу з урахуванням інженерії вимог.
3. Опишіть правила оформлення проектних матеріалів інформаційних систем та технологій.
4. Яким чином проводиться техніко-економічне обґрунтування розробки web-орієнтованої інформаційної системи?
5. Що являють собою системи управління web-ресурсом, та яким чином їх вибір впливає на перед проектну підготовку?
6. Опишіть правила обрання інформаційних систем та середовища розробки для забезпечення проектування web-ресурсу.
7. Яким чином відбувається процес розгортання та налаштування веб-серверу?
8. Які методи дозволяють обґрунтувати економічну ефективність потенційного та реалізованого web-ресурсу?

# Лабораторна робота № 2

# Проектування структури web-ресурсу та алгоритмів сценаріїв

**Мета роботи –** набути навичок проектування структури веб-ресурсу. Освоїти процес розробки та представлення алгоритмів PHP і JavaScript сценаріїв. Навчитися враховувати подальшу інтеграцію складових дизайну.

**Короткі теоретичні відомості**

Структура сайту прямо впливає на його наповнення та навігацію, а від цього залежить можливість його подальшого просування та підтримки. Карта сайту є його структурою та складається зі списку всіх сторінок веб-ресурсу, які повинні бути проіндексовані пошуковими системами. Вірне проектування структури допомагає роботам пошукових систем зрозуміти, які сторінки потребують індексації, розподілити пріоритетність сторінок, виявити оновлені проіндексовані сторінки та частоту оновлення, проіндексувати всі потрібні сторінки. При проектуванні структури сайту необхідно спочатку визначити очікувану модель взаємодії користувача із веб-ресурсом для забезпечення інтуїтивної роботи користувача із сайтом. Не слід забувати про одноманітність елементів структури при уникненні дублюючих сторінок чи розділів, про забезпечення зручності доступу до усіх розділів та формування карти сайту. Після цього формується логічна та фізична структура. Логічна структура складається з відповідних тематичних розділів, в яких міститься відповідна інформація, а зв'язок між розділами реалізується у вигляді гіперпосилань. Під фізичною структурою мають на увазі фізичне розміщення файлів у директоріях веб-сайту. Логічна і фізична структура часто не збігаються, однак рекомендується зберігати відповідність порядку логічної структури веб-сайту його фізичній структурі.

Планування системи інтеграції дизайну та програмної складової будується на основі системи шаблонів. Місця інтеграції в HTML файлі заключаються у фігурні дужки ({}). Рядок, що в них заключений інтерпретується як ім'я змінної. Тому слід визначити відповідні місця документа для розміщення цих ключових рядків.

Розробка алгоритмів PHP та JavaScript сценаріїв має на увазі, визначення призначення та кількості серверних та клієнтських сценаріїв. При розробці алгоритмів слід керуватися класичними правилами побудови алгоритмів, а саме покрокове визначення необхідних дій для досягнення поставленої мети сценарію. За умов існування варіантів реалізації алгоритму, необхідно обрати найбільш раціональний з точки зору поставленої задачі та графічно представити у вигляді блок-схеми. При цьому, існуючі класичні алгоритми відповідних сценаріїв (пошук по сайту, відправка та обробка форми, автоматична розсилка повідомлень тощо) вже відомі, тому слід окрім них, зосередити увагу на сценаріях, що реалізовані для вирішення нетривіальних задач, пов’язаних з конкретним індивідуальним завданням.

Одним із способів графічного представлення алгоритмів на ряду із блок-схемою є представлення у вигляді графу (абстрактного представлення множини об’єктів та їх зв’язків). Слід пам’ятати основні особливості потокового графу: він є відображенням керуючої структури програми; вершини графа – це лінійні ділянки програми; дуги відображають потік управління (передача керування між операторами); існують операторні (виходить одна дуга) та предикатні (виходять дві дуги) вузли; предикатні вузли символізують прості умови.

Налаштування бази даних проводиться шляхом визначення змінних серверу, імені користувача, імені бази даних та паролю у конфігураційних файлах (configuration.php для Joomla, wp-config.php для WordPress, settings.php для Drupal).

**Завдання:**

Спроектувати фізичну та логічну структуру веб-ресурсу. Визначити базовий функціонал серверної та клієнтської частини та розробити алгоритми сценаріїв

Хід роботи:

1. Обрати веб-ресурс згідно індивідуального завдання.
2. Розробити логічну та фізичну структури web-ресурсу (Примітка 1).
3. Визначити місця подальшої інтеграції дизайнерських рішень.
4. Розробити алгоритми PHP сценаріїв з огляду на призначення та функціонал проектованого веб-ресурсу відповідно до індивідуального завдання (Примітка 2).
5. Розробити алгоритми JavaScript сценаріїв з огляду на призначення та функціонал проектованого веб-ресурсу відповідно до індивідуального завдання (Примітка 2).
6. Навести принцип роботи базових сценаріїв у вигляді графів.
7. Провести налаштування бази даних згідно з обраною системою управління веб-ресурсом (лаб. 1).
8. Формування звіту та висновків.

***Примітка 1:*** *у звіті необхідно надати схематичне зображення логічної структури у вигляді дерева переходів.*

***Примітка 2:*** *алгоритми сценаріїв зобразити у вигляді блок-схем та надати у звіті.*

Тематика веб-ресурсу відповідно до індивідуального варіанту (номер варіанту визначається за порядковим номером у списку групи):

1. Сайт актуальних новин.
2. Блог кіберспортсмена.
3. Веб-сайт закладу вищої освіти.
4. Інформаційний портал та консультування.
5. Веб-ресурс прогнозу погоди.

**Контрольні запитання**

1. Опишіть процес проектування структури web-ресурсу.
2. Яким чином на етапі проектування враховується можливість інтеграції складових частин дизайну та які засоби мови PHP дозволяють це реалізувати?
3. Яким чином структура веб-ресурсу впливає на подальше його просування в пошукових системах?
4. Порівняйте алгоритми серверних та користувацьких сценаріїв.
5. Які базові відомі алгоритми Ви використали у своїх сценаріях?
6. Яким чином будується потоковий граф сценарію?
7. Наведіть особливості роботи з базою даних при використанні різних популярних систем управління web-ресурсом.

# Лабораторна робота № 3

# Програмування серверної частини web-ресурсу

**Мета роботи –** набути навичок програмування серверної частини веб-ресурсу. Освоїти реалізацію серверної логіки та процес розробки скриптів за спроектованими алгоритмами. Навчитися забезпечувати захист та оптимізацію PHP-коду.

**Короткі теоретичні відомості**

Для ефективного зберігання та зміни шаблону існує можливість його зберігання в базі даних. Таким чином, необхідно організувати правильний зв'язок між атрибутом в одній таблиці та властивостями в іншій. Однак, слід пам’ятати про імовірні вразливості, зумовлені зберіганням змінних та контенту в незміненому стані, що може спричинити впровадження шкідливого коду. Для ліквідації вразливості слід використовувати фільтр escape при зберіганні шаблону в базі даних.

Поняття динамічного сайту передбачає необхідність формування сайту з динамічних web-сторінок, а це означає, що вони можуть реагувати на дії користувача та певним чином відповідати на них. Web-сервер формує динамічні сторінки з декількох файлів, які, як вже було згадано, можуть зберігатися в базі даних. Ці файли називають шаблонами.

Серверна логіка передбачає виконання певних функціональних задач на стороні серверу. При визначенні саме серверної логіки слід чітко розмежувати клієнтську та серверну логіку в конкретному випадку реалізації програмного продукту. При проектуванні серверної логік слід в першу чергу зважати на безпеку, транзакційність, відмовостійкість, обробку виключень та помилок та швидкість, крім того необхідно взяти до увагу можливість подальшої інтеграції складових дизайну веб-сайту.

Ядром сайту зазвичай виступає CMS система. Однак, завдання лабораторної передбачає окрім забезпечення правильного функціонування готових скриптів також реалізацію власних. На прикладі Wordpress, це можна зробити, підключивши скрипт в functions.php активної теми сайту. Для доступу до функцій Wordpress, необхідно підключити (require\_once '../../../wp-load.php';), а вже для виведення слід зважати на налаштування теми.

Оптимізація коду передбачає зменшення часу виконання PHP-скриптів. Це можливо зробити, провівши певні модифікації коду, наприклад, print замінити на echo, регулярні вирази на strncasecmp, strpbrk і stripos, preg\_replace на str\_replace. Крім того слід вказувати повні шляхи в конструкціях include/require, замість конкатенації передавати в echo декілька параметрів. Способів та підходів оптимізації досить багато і їх використання залежить від конкретної ситуації.

Для захисту PHP коду від ін’єкцій, слід перш за все розміщати користувацький запит в одинарні лапки. Однак цього підходу не досить і слід екранувати ці лапки. В цьому може допомогти функція mysql\_real\_escape\_string(). Для захисту від XSS слід дотримуватись певних правил: не дозволяється включати параметри $\_GET, $\_POST, $\_COOKIE безпеосередньо в HTML, не дозволяється завантаження довільних файлів на сервер. Потенційно небезпечними вважаються функції eval(), preg\_replace(), require\_once(), include\_once(), include(), require(), create\_function(). Однак деякі функції необхідні для роботи з шаблонами. Для уникнення можливості запуску сторонніх файлів, слід змінити налаштування allow\_url\_fopen на Off. Власне в коді слід заблокувати можливість переходу в інші директорії $myPage=str\_replace("/","",$myPage).

**Завдання:**

Реалізувати серверну частину програмного забезпечення веб-ресурсу відповідно до спроектованих алгоритмів та індивідуального завдання. Забезпечити можливість подальшого впровадження дизайну та оптимізувати роботу скриптів.

Хід роботи:

1. Спланувати та організувати ефективну роботу з базою даних (примітка 1).
2. Реалізувати динамічний сайт за допомогою шаблонів PHP.
3. Реалізувати серверну логіку веб-сайту з урахуванням подальшої інтеграції зовнішніх елементів дизайну.
4. Забезпечити роботу ядра сайту та реалізувати базовий функціонал веб-сайту (примітка 2).
5. До базового функціоналу розробити та впровадити скрипт відповідно до індивідуального варіанту завдання.
6. Провести дії з оптимізації роботи PHP скриптів.
7. Забезпечити захист php-скриптів від аналізу та модифікації при можливих атаках за допомогою php-ін’єкції.
8. Формування звіту та висновків.

***Примітка 1:*** *окрім загальних принципів зберігання та доступу до даних, необхідних для функціонування скриптів також організувати зберігання елементів шаблону веб-сайту.*

***Примітка 2:*** *під базовим функціоналом слід розуміти основні скрипти обробки даних, алгоритми яких було спроектовано в ході виконання лаб. 2.*

Види необхідних PHP скриптів відповідно до індивідуального варіанту (номер варіанту визначається за порядковим номером у списку групи):

1. Реалізувати скрипт файлового менеджера для завантаження новин.
2. Реалізувати скрипт голосування.
3. Реалізувати скрипт замовлення та бронювання лекційного курсу.
4. Реалізувати систему безпеки User Levels для розподілення доступу.
5. Реалізувати систему розсилки електронних листів підписникам.

**Контрольні запитання**

1. Опишіть основні особливості, призначення та застосування мови програмування PHP.
2. В чому полягають технічні особливості розробки та реалізації алгоритмів і програмного забезпечення на мові PHP для WWW.
3. Що являє собою динамічний веб-сайт, реалізований за допомогою мови програмування PHP?
4. Опишіть особливості роботи ядра сайту.
5. Яким чином забезпечується шаблонізація засобами PHP та яким чином цей підхід впливає на подальшу розробку та підтримку web-ресурсу?
6. Яким чином оптимізується php код?
7. Які дії проводяться для захисту php-скриптів від аналізу та модифікації?

# Лабораторна робота № 4

# Програмування користувацької частини web-ресурсу

**Мета роботи –** набути навичок програмування користувацької частини веб-ресурсу. Освоїти реалізацію користувацької логіки та процес розробки скриптів за спроектованими алгоритмами. Навчитися забезпечувати захист та оптимізацію JavaScript-коду.

**Короткі теоретичні відомості**

Інтерактивний сайт передбачає відповідне реагування на дії користувача, що забезпечується використання мови програмування JavaScript. Існує декілька способів вбудовування довільного JavaScript-коду. На прикладі WordPress, це може бути функція wp\_print\_scripts (чи admin\_print\_scripts для адміністратору), wp\_localize\_script() для додавання вбудованих скриптів, проте, найкращим варіантом вважається wp\_add\_inline\_script(), який надає повний контроль над виконанням JavaScript -скриптів.

Як зазначалось в попередній лабораторній, серверна логіка передбачає виконання певних функціональних задач на стороні серверу, в той час як користувацька відповідає за взаємодію з користувачем. При цьому, слід зважати на розмежовану клієнтську та серверну логіку в попередній лабораторній. При проектуванні користувацької логіки слід в першу чергу зважати на функціональність, адаптивність та ергономічність сторінки (врахування UX-дизайну).

JavaScript дозволяє проводити оновлення відповідного блоку на сторінці. Існує 2 способи: оновлення з певним інтервалом (не рекомендується у зв’язку з високим навантаженням на сервер) та оновлення відповідно до користувацької події (клік мишкою, натискання клавіші, наведення мишки тощо). При відповідній події проводиться заміна одного блоку на інший і, таким чином, оновлюється контент згідно з користувацькими діями.

Керування мультимедійними засобами за допомогою JavaScript передбачає, в першу чергу, використання спеціальних тегів HTML5 (<video> <source src="video.mp4" type="video/mp4"> </video> <audio> <source src="audio.mp3" type="audio/mp3"> </audio>). Після цього JavaScript отримує відповідний об’єкт (var video = document.querySelector('video');). Врешті, можливо отримати доступ до керування властивостями, методами та подіями об’єкту.

Оптимізація JavaScript-сценаріїв відбувається на основі технології мініфікації коду та об’єднання скриптів. Зменшення коду можливо проводити як вручну, так і за допомогою інструментально-програмних засобів. Прикладом такої системи може служити JavaScript Compressor. В процесі мініфікації видаляються пробіли, коментарі, зайві рядки. Деякі засоби взаємодіють з областю видимості змінних, що дозволяє оптимізувати внутрішню логіку скриптів. Об’єднання відбувається на основі злиття фрагментів коду за принципом одноманітності дій, наприклад, якщо вони реалізовані у вигляді зовнішніх бібліотек, не забуваючи при цьому про гарантування незалежної працездатності цих фрагментів. Все це впливає на пошукову оптимізацію, оскільки забезпечує швидке завантаження сторінок та покращує ранжирування веб-сайту пошуковими системами.

Не існує безпечного способу вбудовування JavaScript-коду в HTML сторінку, оскільки він спричинює вразливість сторінки, однак, можливо зробити JavaScript безпечним. Проблема впровадження символів у рядки вирішується заміною "<!--" на "<\!--", "<script" на "<\script", а "</script" на "<\/script". Однак, екранувати символи / та ! слід вже після серіалізації. Екранувати слід також окремі рядки. Крім того, не слід

**Завдання:**

Реалізувати користувацьку частину програмного забезпечення веб-ресурсу відповідно до спроектованих алгоритмів та індивідуального завдання. Оптимізувати роботу скриптів та організувати безпечне вбудовування.

Хід роботи:

1. Реалізувати інтерактивність сайту, використовуючи мову програмування JavaScript.
2. Реалізувати користувацьку логіку, враховуючи подальше впровадження UX-дизайну.
3. Реалізувати базові клієнтські сценарії мовою програмування JavaScript (Примітка 1).
4. До базових сценаріїв додати функцію автоматизованого оновлення контенту відповідно до варіанту індивідуального завдання (Примітка 2).
5. Реалізувати керування мультимедійними засобами за допомогою JavaScript.
6. Оптимізувати JavaScript-код сценаріїв.
7. Організувати безпечне вбудовування JavaScript в HTML.
8. Формування звіту та висновків.

***Примітка 1:*** *під базовими користувацькими сценаріями слід розуміти основні JavaScript-сценарії, алгоритми яких було спроектовано в ході виконання лаб. 2.*

***Примітка 2:*** *функцію автоматизованого оновлення контенту додати у вигляді вбудованого (inline) скрипту в обрану CMS .*

Функція JavaScript для оновлення блоку на сторінці відповідно до індивідуального варіанту (номер варіанту визначається за порядковим номером у списку групи):

1. Після завантаження власної новини демонструвати варіант оновленого списку новин з доданою.
2. По кліку мишкою на кнопці «сподобалось» оновлювати інформацію з урахуванням вподобань.
3. При натисканні на підтвердження обраної спеціальності додавати опційну інформацію.
4. При обранні консультанта завантажувати інформацію про нього та його функції.
5. При виборі прогнозу на тиждень оновлювати дані по стану пожежної безпеки.

**Контрольні запитання**

1. Опишіть особливості застосування мови програмування JavaScript для створення інтерактивних web-ресурсів.
2. Наведіть технологічний процес розробки клієнтських сценаріїв на мові програмування JavaScript.
3. Що таке користувацька логіка?
4. Які засоби JavaScript дозволяють впроваджувати автоматизоване оновлення контенту та яким чином це відбувається?
5. Опишіть процес керування мультимедійними засобами за допомогою JavaScript.
6. Яким чином зорганізується безпечне вбудовування JavaScript коду?
7. Що має на увазі оптимізація JavaScript коду та яким чином вона впливає на подальше просування?
8. Яким чином додаються inline скрипти в CMS.

# Лабораторна робота № 5

# Налаштування елементів дизайну та наповнення ресурсу контентом

**Мета** – навчитися інтегрувати складові дизайну із серверною та користувацькою логікою. Набути навичок наповнення веб-сайту контентом та забезпечення його релевантності. Освоїти методи формування семантичного ядра та підготовки веб-сайту до просування в пошукових системах.

**Короткі теоретичні відомості**

Дизайн веб-сайту розробляється відповідно до тематики та цільової аудиторії, враховуючи ергономічні властивості. Елементи дизайну інтегруються у відповідні відведені місця шаблону сторінки. Ергономіка у веб-дизайні має на меті створення максимально зручних сторінок, що враховують усі потреби цільової аудиторії. При цьому змінюються відповідним чином такі елементи: час завантаження сторінок; ширина, довжина сторінки і її структура; інтерфейс засобів пошуку; засоби навігації; процедура реєстрації; розмір і розташування логотипу, контактної інформації, анімаційного, музичного, графічного супроводження; розмір, колір, гарнітури шрифтів; колір і тон фону. Наступні характеристики цільової аудиторії слід враховувати: вік, рівень освіти, приналежність до різних соціальних чи професійних груп і ряд інших фізичних та психологічних характеристик.

Досягнення функціонального комфорту передбачає оптимальний функціональний стан відвідувача сайту та наявність сприятливих для нього умов та процесів.

Сканування контенту має на меті показати, яким чином відбувається ознайомлення із контентом веб-сайту. При скануванні F-патерн спочатку користувачі ознайомлюються з верхньою частиною по горизонталі, після цього сканування іде вниз по лівому краю екрану до знаходження цікавою інформації, знайшовши яку, користувач знову читає по горизонталі, а решту контенту продивляється вертикально вниз по лівому краю. Таким чином, слід розміщати найважливіші елементи та блоки у відповідних зонах екрану, які користувач сканує в першу чергу.

Наповнювати сайт слід лише якісним контентом, тобто унікальним та регулярно оновлюваним. При цьому, наповнюючи сайт контентом, як текстовим, так і графічним, слід звертати увагу на уникнення дублювання контенту, зручність навігації, оптимізацію під геолокальний пошук, наявність посилань та оптимізацію мета-тегів.

Семантичне ядро – це сукупність релевантних ключових слів та словосполучень. Сервіс serpstat дозволяє підібрати такі ключі згідно заданої тематики. Додавши певну кількість найбільш вживаних релевантних ключів можливо розширити семантичне ядро. Слід пам’ятати, що щільність ключових слів не повинна перевищувати 1-2 слова на 5-7 рядків тексту.

При цьому, необхідно дотримуватись природності входження ключового слова. Існує ряд різновидів входжень. Точне входження має на увазі використання ключового слова чи фрази дослівно як в технічному завданні. Пряме входження передбачає дослівне використання ключової фрази з можливістю додавання знаку пунктуації між словами. При розбавленому входженні у ключову фразу додаються нові слова. Морфологічне входження передбачає зміну словоформи. Зворотне входження має характеристики прямого зі зміною місць слів. Додаткова класифікація входжень нараховує такі види, як входження з помилками (які навмисно вносяться у фразу у відповідності з уподобаннями користувачів допускати відповідні помилки у запитах), абревіатури та синонімічні (близькі до початкового ключа). При цьому для органічності необхідно враховувати контекст та уникати помилок при відмінюванні, помилок роду, числа тощо.

Загалом виділяють 5 основних принципів композиції веб-сайту: принцип Z-макета (інформація знаходиться на траєкторії руху очей, що нагадує літеру Z), точка фокусування (розміщується важливий об’єкт, який привертає увагу), принцип Гештальта (всі елементи повинні сприйматися цілісно), центральний візуальний образ (всі деталі сторінки вибудовуються навколо зображення із сильним емоційним зарядом), сітка (врівноваження всіх елементів).

Релевантність текстового контенту передбачає відповідність сторінки ключовим запитам користувача. Крім того, підвищенню релевантності сприяє відповідність вимогам: сторінка обов'язково містити текст, заголовок містить ключову фразу з прямим входженням, описання МЕТА Тегів: Tile, Kewords, Description, Alt, розташування головної ключової фрази близько до початку сторінки, виділення заголовків через теги Н1 і Н2 та дотримання ліміту довжини 70 знаків, а також наявність карти сайту.

Релевантність графічного контенту забезпечується при seo-оптимізації зображень, а саме розмір (чим більше зображення тим краще, проте не слід забувати про загальну структуру та композицію сторінки), розміщення (близько до тексту, або текст повинен обтікати зображення), формат та підпис (підписи під зображення, що максимально точно його називають) зображення впливає на його ранжирування пошуковими системами. Для цього слід виконувати обробку зображень засобами мови PHP, зокрема на зміну розміру (resize), обрізку зображень (crop), їх поворот (imagerotate()), дзеркальне відображення (imageflip()), накладання фону, водяного знаку (watermark) та фільтрів.

Забезпечення регулярності оновлення контенту передбачає створення контент-плану та обов’язкове слідування йому. Контент план є списком статей різнобічної тематики з вказівками крайніх термінів публікації відповідної статті у відповідному розділі веб-сайту та виконавця роботи. Зазвичай контент-план будується на 1-3 місяці (в роботі необхідно побудувати контент-план на 2 місяці), проте зустрічається і на рік.

Коефіцієнт конкордації вказує на необхідність оптимізації досліджуваного ключового слова. Нехай m – кількість пошукових систем, а n – досліджуваних сайтів. Тоді слід побудувати матрицю розмірності m x n. Значення кожного елементу матриці – позиції сайтів за досліджуваним ключовим запитом. Кожному значенню приписується умовна числова мітка (ранг), що означає порядковий номер елемента в стовпці n. Якщо два об’єкти мають однакове значення, їм присвоюється ранг, що дорівнює середньому арифметичному їх значенню. Після цього проводиться обчислення коефіцієнта за формулою (1):

   , (1)
де

   . (2)
 Після дослідження необхідної кількості веб-сайтів проводиться дослідження розробленого сайту за заданим ключовим словом. Якщо статистичний зв’язок коефіцієнту конкордації не зберігається, а значення знижується, це означає необхідність оптимізації сайту по цьому ключовому слову.

**Завдання:**

Розробити ергономічний дизайн та інтегрувати його з виконуваними скриптами. Наповнити веб-сайт релевантним текстовим і графічним контентом, враховуючи особливості оновлення та розробити семантичне ядро для подальшого просування.

Хід роботи:

1. Розробити дизайн сторінки та інтегрувати його (Примітка 1).
2. Забезпечити ергономічність дизайну та функціональний комфорт ергодизайнерських рішень.
3. Підготувати контент головної сторінки до сканування за траекторією F-патерн.
4. Організувати композицію другорядних сторінок за одним з 5 найпоширеніших принципів, що найбільше відповідає тематиці та цільовій аудиторії.
5. Наповнити веб-сайт текстовим (Примітка 2) та графічним контентом.
6. Розробити семантичне ядро.
7. Впровадити ключові слова розробленого ядра у текст та додати ключові слова відповідно до індивідуального варіанту, враховуючи природність входжень, необхідну кількість повторення у необхідних місцях.
8. Забезпечити релевантність текстового контенту.
9. Забезпечити релевантність графічного контенту за допомогою засобів PHP.
10. Розробити контент-план із зазначенням списку статей різнобічної тематики, що потребують оновлення та вказівками крайніх термінів публікації відповідної статті у відповідному розділі веб-сайту.
11. Провести аналіз необхідності оптимізації одного обраного ключового слова, використовуючи обчислення коефіцієнту конкордації.
12. Формування звіту та висновків.

***Примітка 1:*** *інтеграція забезпечується в спеціально визначених у лаб. 3 місцях.*

***Примітка 2:*** *текстовий контент повинен відповідати вимогам якості, а отже унікальність має бути не нижче 85%, рівень спаму не повинен перевищувати 50%, водності 65%, нудотності 10%.*

Ключові слова, які потрібно обов’язково включати до семантичного ядра (номер варіанту визначається за порядковим номером у списку групи):

1. Цікаві новини, корисна інформація.
2. Комп’ютерні ігри змагання.
3. Освітня програма, компетентності.
4. Інформування, актуальні дані.
5. Опади, вологість повітря.

**Контрольні запитання**

1. Охарактеризуйте способи забезпечення ергономічності засобами PHP та JavaScript.
2. Яким чином забезпечується функціональний комфорт ергодизайнерських рішень при проектуванні web-ресурсу?
3. Опишіть процес створення та оновлення текстового контенту з урахуванням можливості подальшого просування web-ресурсу.
4. Охарактеризуйте вплив графічного контенту на процес просування та підтримки web-ресурсу.
5. Яким чином забезпечується релевантність контенту web-ресурсу?
6. Охарактеризуйте математичні моделі та методи визначення релевантності ключових запитів.
7. Наведіть процес формування семантичного ядра та види входжень ключових слів.
8. Яким чином здійснюється обчислення коефіцієнту конкордації?

# Лабораторна робота № 6

# Створення та проведення тест-кейсів, забезпечення індексації ресурсу, його підтримки і просування в мережі

**Мета** – навчитися застосовувати різні види входжень ключових слів. Набути навичок забезпечення морфологічної якості тексту шляхом дотримання природності входження ключових слів. Освоїти процес забезпечення синтаксичної та пунктуаційної якості.

**Короткі теоретичні відомості**

Виявити рівень якості розробленої системи перед початком процесу просування допоможе різнопланове тестування. Таким чином, тестування установки версії передбачає виявлення загальну робото спроможність веб-сайту, тестування білої скриньки перевіряє внутрішнє функціонування і логіку роботи коду, чорної скриньки здійснює перевірку елементів, доступ до вихідного коду яких відсутній (готові реалізовані скрипти CMS), тестування користувацького інтерфейсу передбачає перевірку правильної взаємодії усіх компонентів інтерфейсу між собою та з користувачем, навантажувальне тестування перевіряє роботу системи при стандартних навантаженнях, а стресове при пікових, тестування безпеки має на меті перевірку цілісності та захищеності сайту від атак. Крім того, тестування сірої скриньки має на увазі об’єднання підходів білої та чорної скриньки, стохастичне – передбачає наявність незалежних тестів, в яких дані генеруються випадковим чином, юзабіліті тестування направлене на встановлення рівня зручності користування сайтом, а тестування ефективності перевіряє обсяг коду, який використовується для виконання однієї операції. Обмежень на інструментальні засоби, використання автоматизованого тестування в роботі не висувається.

Виявлення дефектів веб-сайту та його тестування супроводжується звітом, який описує дефект та сприяє його усуненню. Звіт складається з метою надати інформацію про проблему та сприяти її усуненню. У звіті вказуються наступні пункти: виявлено (початковий стан), призначено (призначено відповідального за усунення), виправлений (усунено дефект), перевірений (підтвердження усунення), закрито (з дефектом не планується подальших дій), відкрито заново (дефект відтворюється), рекомендований до відхилення, відхилений (замість стану закрито), відкладений (виправлення дефекту не є раціональним на часі). В стан «Закрито» звіт може переводитись з резолюціями: не є дефектом, дублікат, не вдалося відтворити, не буде виправлено, неможливо виправити.

Підготовка веб-сайту до індексації передбачає дії схожі з правилами забезпечення релевантності сторінок, а саме слід перевірити чи сторінка містить текст, чи заголовок містить ключову фразу з прямим входженням, чи вірне описання МЕТА Тегів: Tile, Kewords, Description, Alt та чи головна ключова фраза розташовується фрази близько до початку сторінки.

SEO поділяється на 2 види: внутрішню та зовнішню оптимізацію. До внутрішньої відносять: розміщення нового та редагування старого контенту, коригування файлу robots.txt, редагування мета-тегів description і title, редагування програмного коду, внутрішня перелінковка, робота з ключовими словами, налаштування юзабіліті та дизайну. До зовнішньої відносять: усунення недоліків, що можуть вплинути на просування, моніторинг позицій сайту, реєстрація сайту в каталогах та пошукових системах, робота із зовнішніми посиланнями.

Одним з етапів внутрішньої оптимізації є налаштування файлів robots.txt та sitemap.xml. Robots.txt містить рекомендації пошуковим системам щодо дозволу індексації певних сторінок чи каталогів на сайті. Директива User-agent містить назву робота пошукової системи, директива Disallow відповідає за дозвіл. Забороняти індексацію слід для тих сторінок, що мають не унікальний контент (ліцензійні умови, політика конфіденційності тощо) чи знаходяться в процесі розробки. Файл sitemap.xml містить усі посилання на наявні сторінки, які оформлюється кожне окремим тегом <url>. В межах цього тегу прописуються теги <loc>, що містить власне посилання на сторінку, <lastmod>, що відповідає за дату останньої модифікації, <changefreq> – частота оновлення та <priority> – приорітетність індексації. Не слід забувати, що у файлі використовується лише кодування UTF-8, його максимальний об’єм 10 Мб, а максимальна кількість посилань 50 тис.

Для налаштування CMS з метою Seo-просування необхідно дотримуватись наступних правил: редагування мета тегів, правка заголовків H1, забезпечення 301 редіректу, налаштування дзеркал сайту, уникнення генерації дублюючих сторінок, пришвидшення CMS, вірне налаштування robots.txt, автоматизована генерація sitemap.xml, спрощення підключення SSL-сертифікату, керування URL, підтримка мультирегіональності, коректне опрацювання помилок, можливість налаштування alt і title, а також зручність редагування контенту.

Адміністрування веб-сайту на хостінгу може здійснюватись за допомогою різних засобів. До базових засобів відносяться панель керування хостінгом, FTP-сервер та веб-диск. Іншим способом, спільним для усіх CMS, як Joomla, Wordpress, Datalife, Drupal, Bitrix є використання власної панелі керування для управління контентом. І останнім способом є власне редагування бази даних, використовуючи PHPMYADMIN.

Додати веб-сайт у пошукові системи можна за одним принципом для усіх систем. Кожна пошукова система надає відповідну форму, що дозволяє це зробити. Наприклад, щоб додати сайт в пошукову систему Google, слід перейти за посиланням <http://www.google.com/addurl/?continue=/addurl> та слідувати інструкціям, для Bing <http://www.bing.com/toolbox/submit-site-url> , для Meta.ua <http://web.meta.ua/?action=add> .

На швидкість підняття веб-сайту в пошуковій системі насправді впливає багато факторів, це і якість внутрішньої оптимізації, і оновлення алгоритмів ранжирування, і вік домену, а також рівень конкуренції та бюджет. В роботі не стоїть задача виведення сайту в топ за короткий проміжок часу. Необхідно лише покращити позиції розташування сайту.

Інформаційна підтримка сайту включає в себе додавання нових матеріалів на сайт, коригування існуючих та видалення неактуальних матеріалів, роботу з датами та постійну пошукову оптимізацію сторінок. До технічної підтримки відносять контроль працездатності хостінгу та серверу, захист від спаму та копіювання контенту, розширення функціональних можливостей сайту та проведення резервного копіювання даних.

**Завдання:**

Провести всебічне тестування розробленого веб-сайту. Провести дії щодо підготовки сайту до індексації та опублікувати його в мережі. Забезпечити просування сайту та можливість його подальшого технічного та інформаційного супроводу.

Хід роботи:

1. Розробити та провести тестування: установки версії, методом білої та чорної скриньки, користувацького інтерфейсу, навантажувальне, безпеки, а також відповідно до варіанту індивідуального завдання.
2. +Сформувати звіт на основі результатів тестування та виправити виявлені дефекти.
3. +Підготувати веб-сайт до індексації.
4. +Завершити внутрішню та провести зовнішню оптимізацію.
5. Розмістити розроблений веб-сайт на обраному в лаб.1 хостінгу.
6. +Провести підготовку веб-сайту до просування (Примітка 1).
7. +Провести налаштування CMS для Seo-просування.
8. +Організувати процес адміністрування веб-сайтом на хостінгу (Примітка 2).
9. +Додати веб-сайт до пошукових систем та забезпечити індексацію.
10. +Підняти веб-сайт у пошуковій видачі.
11. +Надати план дій щодо подальшої інформаційної та технічної підтримки веб-сайту.
12. Формування звіту та висновків.

***Примітка 1:*** *створити, відповідним чином наповнити та розмістити в кореневій папці на хостінгу файли robots.txt sitemap.xml.*

***Примітка 2:*** *У звіті надати демонстрацію керування з адміністративної панелі системи управління контентом чи адміністративної панелі власне конкретного хостінга, де розміщено сайт.*

Вид додаткового тестування відповідно до індивідуального варіанту (номер варіанту визначається за порядковим номером у списку групи):

1. Тестування методом сірої скриньки.
2. Стресове тестування.
3. Стохастичне тестування.
4. Юзабіліті-тестування.
5. Тестування ефективності.

**Контрольні запитання**

1. Які існують вимоги до якості складних інтерактивних web-ресурсів?
2. Перелічіть основні види тестування web-ресурсу та опишіть яким чином кожен вид використовується і з якою метою.
3. Яким чином створюється звіт в процесі та по завершенню тестування web-ресурсу?
4. Що являє собою процес індексації web-ресурсу та яким чином можливо підготувати сайт до індексації? Детально опишіть призначення та наповнення кожного додаткового файлу та особливості структури веб-сайту?
5. Дайте визначення зовнішній та внутрішній оптимізації та опишіть процес реалізації кожного з цих видів оптимізації.
6. Яким чином забезпечується процес подальшого супроводу web-ресурсу?
7. Опишіть методику адміністрування створеним web-ресурсом та підходи забезпечення цього процесу засобами PHP.
8. Як забезпечити можливість простої та зручної підтримки web-ресурсу?

# ЛІТЕРАТУРА

1. Дронов В.А. Lavarel. Быстрая разработка современных динамических Web-сайтов на PHP, MySQL, HTML и CSS. СПб. : БХВ-Петербург, 2017. 768 с.
2. Холл М., Браун Л. Программирование для Web. Библиотека профессионала. М. : Издательский дом «Вильямс», 2002. 1264 с.
3. Севостьянов И.О., Поисковая оптимизация. Практическое руководство по продвижению сайта в Интернете. СПб. : Питер, 2010. 240 с.
4. Кузнецов М.В., Симдянов И.В. PHP. Практика создания Web-сайтов. СПб. : БХВ-Петербург, 2008. 1264 с.
5. Easttom C. Advanced JavaScript. 3rd Edition. Plano : Worldware Publishing, Inc., 2008. 591 p.
6. Rahman M. PHP 7 Data Structures and Algorithms. Birmingham : Packt Publishing, 2017. 340 p.
7. Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL и JavaScript. Часть 2. СПб. : Питер, 2011. 496 с.
8. Shaw M. Copywriting: Successful Writing for Design, Advertising and Marketing London : Laurence King Publishing, 2012. 240 p.
9. Порошин С.М., Можаєв О.О., Можаєв М.О. Методологія проведення pen-тестування веб-додатків. *Системи обробки інформації*. 2016. Випуск 3 (140). С. 33-35.
10. Ткаченко В.П., Огірко І.В., Огірко О.І. Математична модель оцінювання захисту web-сайтів. *Полиграфические, мультимедийные и WEB-технологии (PMW-2016): тези доповідей 1-ї Міжнародної науково-технічної конференції (м. Харків, 16–20 травня, 2016 р.)*. Т. 1. С. 98–101.
11. Кубашева Е.С., Гаврилов А.Г. Методика оценки качества веб-приложений. Программные системы и вычислительные методы. 2013. №1(2). С. 28-34.
12. Голобородько В.М., Опалєв М.Л. Формування дизайн-ергономічних вимог до користувацького інтерфейсу web-сайтів. *Вісник ХДАДМ*. 2011. № 3. С. 16-19.
13. Designing the User Interface: Strategies for Effective Human-Computer Interaction. 6th Edition / B. Shneiderman, C. Plaisant, M. Cohen [at al.]. Boston : Pearson, 2017. 580 p.
14. Назайкин А.Н. Копирайтинг:XXI век. М. : КДУ, 2017. 316 с.
15. Одинокая М.А. Об эффективном использовании современных технологий создания интерактивных веб-документов. *Интерактивная наука*. 2017. № 13. С. 55-56.
16. Бойко Н.І. Моделювання web-орієнтованих систем та напрямки розвитку web-ресурсів. *Вісник Національного університету «Львівська політехніка» (Серія інформаційні системи та мережі)*. 2012. Випуск 743, № 1. СС. 16-25.
17. Antani V. Mastering JavaScript. Birmingham : Packt Publishing, 2016. 250 p.
18. Пугачевська К.Й. Діагностика інтернет-торгівлі як інноваційної технології *Науковий вісник УжНУ Серія: Економіка*. 2013. Випуск 1 (38) . С. 236–239.
19. Ергодизайн. Основи методології і практики / В.М. Голобородько та ін.; під заг. ред. В.О. Свірка. Київ : ТОВ «Школа», 2009. 270 с.
20. Силаков Д.В. Автоматизация тестирования Web-приложений, основанных на скриптовых языках. *Труды Института системного программирования РАН*. 2008. Том 14, № 2. С 159-178.
21. Коцюба И.Ю., Чунаев А.В., Шиков А.Н. Методы оценки и измерения характеристик информационных систем. Учебное пособие. СПб: Университет ИТМО, 2015. 264 с.
22. Шматко О.В., Мироненко М.І. Інформаційна технологія відслідковування помилок програмного забезпечення. *Збірник наукових праць Харківського національного університету Повітряних Сил*. 2018. № 2(56). С. 120-125.
23. Ганеев Р.М. Проектирование интерактивных WEB-приложений. М. : Горячая линия-Телеком, 2001. 272 с.
24. Пелещишин А.М. Методи та алгоритми моделювання Web-систем. *Інформаційні системи та мережі: Вісник Нац. ун-ту “Львівська політехніка”*. 2000. № 406. С. 199–211.
25. Алєксєєва К., Берко А., Висоцька В. Особливості процесу управління Web-ресурсом комерційного контенту на основі нечіткої логіки. *Вісник Національного університету "Львівська політехніка". Серія: Комп’ютерні науки та інформаційні технології : збірник наукових праць*. 2015. № 826. С. 201–211.
26. Зандстра М. PHP. Объекты, шаблоны и методики программирования. 2-е издание. М. : Вильямс, 2010. 480 с.