

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

Бакалаврська кваліфікаційна робота

на тему: Створення інформаційної системи для підприємства «Проектування
веб-сайту для оренди автомобілів та яхт»

Виконав студент 4 курсу групи K-42
За напрямом 6.05.01.01,
Романенко Денис Олександрович

Керівник к.т.н., доцент
Трегубова Ірина Анатоліївна
Рецензент к.ф.-м.н., доцент
Вітавецька Лариса Анатоліївна

Одеса 2019

ЗМІСТ

Вступ.....	5
1 Теоретична частина.....	7
1.1 Обґрунтування потреби у веб-сайті.....	7
1.2 Поняття про веб-сайт	8
1.3 Проектування веб-сайту.....	10
1.4 Етапи розробки веб-сайту.....	15
2 Аналіз задач, шляхів їхнього вирішення, конкурентів.....	20
2.1 Аналіз задач	20
2.2 Аналіз конкурентів.....	20
2.3 Аналіз шляхів вирішення задачі.....	21
2.4 Аналіз доступного ПЗ.....	22
2.4.1 Аналіз варіантів ОС	22
2.4.2 Аналіз варіантів ПЗ для веб-серверів.....	23
2.4.3 Вибір мови програмування серверної частини сайту	24
2.4.4 Аналіз додаткового ПО.....	29
2.5 Аналіз готових рішень	29
2.5.1 Аналіз фронт-енд фреймворків	29
2.5.2 Аналіз CMS.....	30
3 Проектна частина	42
3.1 Загальна структура сайту	42
3.2 Дизайн веб-сайту.....	43
3.3 Створення HTML-шаблону та стилі-CSS	45
3.4 Створення javascript скриптів для динамічної зміни контенту та підключення плагінів.....	48
3.5 Встановлення WordPress	50
3.6 Створення теми	50
3.7 Створення валідації форми	55
Висновки	56
Перелік джерел посилань	57

ВСТУП

Актуальність теми. З появою веб-технологій комп'ютер починають використовувати абсолютно нові верстви населення світу. Можна виділити дві найбільш характерні групи, що знаходяться на різних соціальних полюсах, які були стрімко залучені в нову технологію, можливо, навіть і не за власним бажанням. З одного боку, це були представники елітарних груп суспільства – керівники великих організацій, президенти банків, топ-менеджери, впливові державні чиновники тощо. З іншого боку, це були представники найрізноманітніших верств населення: домогосподарки, пенсіонери, діти та інші.

З розвитком технологій гіпертекстової розмітки в мережі Інтернет почало з'являтися все більше сайтів, тематика яких була абсолютно різною – від сайтів великих компаній, що сповіщають про успіхи компанії та її провали, до сайтів маленьких фірм, що пропонують відвідати свої офіси в межах одного міста.

Також в наш час усе більше розповсюджується купівля будь-яких товарів в Інтернет-магазинах. Шопінг – заняття корисне, іноді захоплююче, що приносить як позитивні емоції, так і розчарування. Уже минають ті часи, коли для того, щоб стати власником бажаної речі, потрібно було на своїх «двух» здійснювати тривалі прогулянки офлайновими торговими місцями. Тепер у наше життя активно входять Інтернет-магазини, ми все частіше здійснюємо покупки шляхом кількох натисків миші, залишаючись при цьому на своєму робочому місці, вдома або взагалі у будь-якому місці, де наявний Інтернет.

Аналіз останніх досліджень і публікацій. Теоретичні й методологічні дослідження даного питання знайшли відображення в працях Консорціуму Всесвітньої павутини (W3C), працях Тімоті Джон Бернерса-Лі, Расмуса Лердорфа, Брендана Ейха та інших. Разом з тим, недивлячись на наявність великої кількості наукових праць, присвячених даній тематиці, в цілому, вона пот-

ребує постійного вивчення, оскільки галузь розробки веб-технологій розвивається стрімкими темпами.

Метою роботи є розробка комерційного веб-сайту для приватного підприємця, що спеціалізується на оренді автомобілів та яхт.

Згідно із сформульованою метою дипломної роботи були поставлені та розв'язанні такі завдання:

- проаналізувати існуючі сайти з подібними послугами;
- розглянути сучасні технології, що можуть допомогти або покращити розробку веб-сайту;
- проаналізувати переваги та недоліки використання CMS;
- обґрунтувати переваги сучасних технологій.

Об'єктом дослідження є процес впровадження новітніх ІТ-технологій у сфері розробки веб-сайтів.

Предметом дослідження є теоретичні, методологічні та прикладні аспекти у створенні комерційних веб-сайтів з використанням нових технологій та стандартів W3C (Консорціум Всесвітньої павутини).

Методами дослідження: аналітичний, статистичний, метод порівняння, систематизації даних та програмний.

Практична значущість отриманих результатів полягає у тому, що отримані висновки сприятимуть автоматизації процесів відправлення заявки на приватному підприємстві та зручнішому доступу до інформації.

Загальна характеристика кваліфікаційної роботи:

- повний обсяг сторінок пояснювальною записки – 59
- кількість рисунків – 21
- кількість таблиць – 0
- кількість посилань – 27

1 ТЕОРЕТИЧНА ЧАСТИНА

1.1 Обґрунтування потреби у веб-сайті

У сучасних реаліях дуже важко уявити життя людини без комп'ютера, оскільки вони стали невід'ємною частиною нашого життя. Вони беруть на себе основну частину рутинної роботи, а також вони кардинально змінили уявлення людини про навчання, роботу і навіть відпочинок. Такі зміни, на нашу думку, зумовлені доступністю користувачу інформації, високою швидкістю її опрацювання та видачі.

У зв'язку з цим, у наш час майже кожна організація має власний веб-сайт, адже це дозволяє розширити поле рекламної компанії та привернути таким чином нових клієнтів або нагадати про своє існування вже існуючим.

Створення та розробка сайтів включає в себе:

- затвердження первинного технічного завдання для подальшої розробки сайту;
- визначення структурної схеми сайту, тобто навігації, розташування розділів та контенту;
- веб-дизайн, який передбачає собою графічне створення макету сайту, стилів та елементів навігації;
- розробка програмного коду, модулів, бази даних та інших елементів сайту, що необхідні для реалізації проекту;
- тестування та розміщення сайту в мережі Інтернет [1]¹⁾.

¹⁾ [1] Бабаев А., Евдокимов Н., Бодє М. Создание сайтов. Санкт-Петербург, 2014. 410 с.

1.2 Поняття про веб-сайт

Інформація, яка доступна користувачам мережі Інтернет, розташована на комп'ютерах (веб-серверах), на яких встановлено спеціальне програмне забезпечення. Значна частина даної інформації організована у вигляді веб-сайтів, кожен з яких має своє ім'я (адресу) в мережі Інтернет. Веб-сервер є обов'язковим для функціонування будь-якого веб-сайту, а для тих сайтів, що є складнішими, ніж набір статичних сторінок, на яких можна виконувати лише керування контентом, використовується система управління веб-контентом.

Насамперед, веб-сайт (англ. *website*) – це сукупність веб-сторінок, що доступні у мережі Інтернет та об'єднані як за змістом, так і навігаційно, використовуючи єдине доменне ім'я. [5]¹⁾ Для перегляду веб-сайтів на комп'ютері користувача використовуються спеціальні програми, які називаються браузерами. Залежно від того, яке ім'я (адресу) сайту ми задамо в рядку «Адреса», браузер завантажуватиме в своє вікно відповідну інформацію.

Кожна сторінка веб-сайту також має свою Інтернет адресу, яка складається з адреси сайту й імені файлу, відповідного даній сторінці. Таким чином, веб-сайт – це інформаційний ресурс, що складається із зв'язаних між собою гіпертекстових документів (веб-сторінок), розміщений на веб-сервері й такий, що має індивідуальну адресу. Подивитися веб-сайт може будь-яка людина, що має комп'ютер, підключений до мережі Інтернет.

Фізично, сайт може розміщуватися як на одному, так і на декількох серверах. Веб-сторінка є текстовим файлом з розширенням *.htm чи *.html, який містить текстову інформацію й спеціальні команди – HTML-теги, що визначають, в якому вигляді ця інформація відображатиметься у вікні браузера. Уся графічна, аудіо- і відео-інформація безпосередньо у веб-сторінку не

¹⁾ [5] Вікіпедія. Веб-сайт. URL: [https://uk.wikipedia.org/wiki/ %D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82) (дата звернення 30.03.2019).

входить і є окремими файлами з розширеннями *.gif, *.jpg (графіка), *.mid, *.mp3 (звук), *.avi (відео), *.swf (Flash-аплікація). У HTML-кодi сторінки містяться тільки вказівники на такі файли.

Окрім файлів з розширенням *.htm та *.html, веб-сторінки можуть бути створенні за допомогою різноманітних мов програмування. Так, наприклад, в даному дипломному проєкті веб-сайт буде створено на основі середовища програмування Linux Mint. Для роботи веб-сайту, розробленому в цьому середовищі програмування, сервер повинен бути обладнаний додатковим програмним забезпеченням, який дозволяє розуміти та компілювати код, перетворюючи його у відповідний HTML-код, який вже в свою чергу сприймається браузером. Для цього нами був використаний сервер Apache. Також у ході роботи були використані: PHP, MySQL, PHPMyAdmin (GUI для управління базою даних), Photoshop, HTML5/CSS3, JavaScript+jQuery.

Кожна сторінка веб-сайту також має свою Інтернет адресу (URL), яка складається з доменного імені, імені файлу та при потребі з GET параметрів. Таким чином, веб-сайт – це інформаційний ресурс, що складається із зв'язаних між собою гіпертекстових документів (веб-сторінок), розміщений на веб-сервері й такий, що має індивідуальну адресу.

Переглянути веб-сайт можна практично з будь-якого комп'ютера, підключений до мережі Інтернет. Сучасні мобільні телефони також обладнані браузером та можливістю завантаження та встановлення браузерів інших розробників, що дозволяє мати можливість доступу до мережі Інтернет. Щоправда, перегляд сайтів на відносно невеликих дисплеях телефонів є дещо незручним і саме тому для більшості веб-сайтів додатково створюють WAP версії, які спеціально оптимізовані під зручне користування ними з мобільних телефонів.

1.3 Проектування веб-сайту

Проектування сайту – це один з перших етапів розробки сайту. На цьому етапі розробляється архітектура майбутнього сайту з урахуванням специфіки бізнесу та цілей веб-проекту.

Проектування сайту представляє собою ключовий етап створення сайту, в процесі якого визначаються цілі та завдання створення ресурсу, способи досягнення цілей і, як сайт буде виглядати й функціонувати у готовому варіанті [14]¹⁾.

Перевагами проектування є:

- висока гарантія досягнення результатів;
- економія часу й коштів;
- ефективний поділ роботи тощо.

Також слід зазначити, що виправити помилку на етапі проектування набагато дешевше та легше, ніж на етапі дизайну, верстки або програмування.

Етапи проектування сайту є наступними:

- визначення цілей розробки;
- сегментація цільової аудиторії;
- загальний опис проекту й необхідних результатів;
- визначення вимог до проекту (графіка, ергономіка, система управління, експлуатація тощо);
- розробка структури сайту (порядок розташування матеріалів сайту відносно один одного);
- розробка дизайну, опис контенту й функціональності сайту [4]²⁾.

¹⁾ [14] Панфилов К. По ту сторону веб-страницы. – М.: ДМК Пресс, 2008. – 440 с.

²⁾ [4] Википедия. Веб разработка. URL: <https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0> (дата звернення: 30.03.2019).

Проектування сайтів має свій результат у вигляді технічного завдання, яке є невід'ємною частиною договору на розробку сайту.

Проектування – це, можливо, ключовий етап створення веб-сайту, що відповідає нам на наступні питання:

- які наші цілі – навіщо ми робимо сайт?
- як ми реалізуємо поставлені цілі?
- як сайт буде виглядати й працювати?

Тільки чітко сформулювавши завдання, визначивши цільову аудиторію сайту та її потреби, змодельовавши взаємодію сайту з його користувачами, ми можемо бути впевнені у тому, що ми отримаємо те, що потрібно.

Виправити помилку на етапі проектування досить просто: міняємо кілька шматків тексту і схем. Зробити це на етапі розробки дизайну або верстки буде вже дорожче. Якщо ви бачите баг на етапі програмування, його виправлення може коштувати делілька тисяч (а, можливо, і більше) гривень і може зайняти місяці, а то й роки.

Проектне завдання – це цілком самодостатній документ. Отримавши його, клієнт може зробити сайт своїми силами або найняти іншу команду, яка, на його думку, краще впорається безпосередньо з розробкою [так само].

Не слід нехтувати проектуванням навіть для невеликих сайтів, адже навіть односторінковій візитці це буде виключно корисно. Виняток, мабуть, становлять недорогі сайти з бюджетом 2-6 тисяч гривень, де проектування стає нерентабельним [так само].

На етапі проектування мета й завдання формуються більш точно й детально, а також визначаються завдання сайту, виконання яких буде працювати на досягнення кожної мети.

Дослідження контексту необхідно проводити для отримання інформації, яку ми будемо називати контекстом сайту. Під контекстом ми розуміємо різні обставини, що оточують сайт і здатні вплинути на його роботу. До таких обставин відносяться:

- цільова аудиторія та її потреби;
- характеристика й тенденції області;
- конкуренти та їхня діяльність;
- досвід інших проектів;
- законодавчі чи інші обмеження;
- інші фактори впливу, в залежності від тематики проекту[16]¹⁾.

Контекст проекту допомагає нам зрозуміти цільову аудиторію, а також те, яким потрібно зробити сайт: як його позиціонувати, яка інформація на ньому повинна бути і, якою мовою він повинен говорити з цільовою аудиторією (це називають комунікативною стратегією), як він буде відрізнятися від конкурентів (природно, в кращу сторону).

Крім того, дослідження занурює команду проекту в тему, дозволяє десь навіть на підсвідомому рівні бачити / приймати правильні рішення.

Найкраще, звичайно, дослідження допомагає зрозуміти аудиторію: як із нею спілкуватися: які слова використовувати в «розмові» з нею, як назвати свій продукт або послугу (можливо, потрібно використовувати професійний сленг, або навпаки, спрощену термінологію), звертатися до неї на «ви» або на «ти».

Які її потреби: що пріоритетно, що необов'язково, але бажане, що не можна робити ні в якому разі.

В ідеалі, дані про контекст ми повинні отримати від клієнта, але, якщо таких даних у клієнта немає, то дослідження контексту слід провести самим.

Для дослідження, на нашу думку, краще використовувати два методи, які найкраще зарекомендували себе на великій кількості проектів:

¹⁾ [16] Тидвелл Дж. Разработка пользовательских интерфейсов., СПб.: Питер, 2011. 480 с.

- дослідження доступних джерел – література, Інтернет-ресурси і так далі;
- інтерв'ю з ключовими дійовими особами – користувачами, експертами.
- Підстави для того, аби проводити дослідження контексту є наступні:
- без даних про контекст шанси зробити добре користувачеві прагнуть до нуля;
- навіть самотійне дослідження дає корисну інформацію, нехай і в меншому обсязі, ніж професійне;
- навіть, якщо дослідження проводиться настільки невміло (що майже неможливо), що не можна отримати відповіді на вищезазначені питання, воно все одно дає змогу зануритись у контекст проекту та хоча б підсвідомо відчувати щось корисне.

Після того, як цілі поставлені, дані про контекст отримані, слід додати всю наявну інформацію в концепцію. Нагадаю, під концепцією ми розуміємо основні ідеї та можливості, закладені в проект:

- що і для кого ми робимо – загальна ідея й цільова аудиторія;
- як сайт буде працювати і яку інформацію містити;
- як сайт буде заробляти (якщо це проект з прямою монетизацією);
- які будуть відмінні риси сайту (від конкурентів), як він буде позиціонуватися;
- як сайт буде розвиватися після запуску[12]¹⁾.

Концепція задає напрямок проектування й допомагає, аналогічно баченню, ще раз зістикувати точки зору на проект – компанії-розробника та клієнта.

¹⁾ [12] Купер А., Рейманн Р. М., Кронин Д. Алан Купер об интерфейсе. Основы проектирования взаимодействия., СПб.: Символ-Плюс, 2017. 688 с.

Моделювання сайту – це створення моделі сайту, яка описує функціональні можливості й інформаційну структуру.

У функціональній частині моделі ми описуємо можливості, які сайт надає своїм користувачам: наприклад, викладати, групувати й коментувати фотографії (соціальна мережа) або замовляти й оплачувати товар (Інтернет-магазин). Важливо розуміти, що можливості є інструментами вирішення завдань. Якщо придумана можливість не вирішує жодного з поставлених завдань, це може означати, що вона зайва. У проектному завданні можливості описуються на досить високому рівні абстракції – не так детально, як це робиться в «Технічному завданні на програмування», оскільки тут це просто не потрібно [так само].

Інформаційна структура – це схема, що показує, з яких розділів складається сайт, які завдання вони вирішують і як користувач буде переміщатися сайтом (схема навігації) [так само].

Далі опрацьовується схему розділу – це більш глибоко і детально пророблена схема (в порівнянні з інформаційною структурою сайту), що показує навігацію по розділу, зв'язку та переходи між підрозділами. Схема розділу в ідеалі включає наступні елементи [так само]:

- завдання – які з раніше поставлені завдання вирішує розділ. Наприклад, розділ «Фотографії» в соціальній мережі вирішує завдання обміну інформацією між друзями й завдання подальшого спілкування;
- повідомлення – це в буквальному сенсі повідомлення, які розділ або його частина передає відвідувачеві. Наприклад, повідомлення вітання «Привіт, ти на порталі XXX! Ми раді бачити тебе тут» бо «Ей, це наш найкращий товар – спробуй, замов зараз і переконайся в цьому сам!». Повідомлення бувають різних типів, що найчастіше зустрічаються: рекламні, заклики до дії, повідомлення та іміджеві повідомлення;

- функціональні елементи – елементи інтерфейсу, що дають можливість відвідувачеві виконати якусь операцію. Наприклад, функціональним елементом є форма для введення повідомлення, що дозволяє відправити повідомлення, або кнопка в інтерфейсі, що зберігає зроблені зміни;
- варіанти поведінки відвідувача – припущення про те, що відвідувач сайту може або повинен зробити після вивчення інтерфейсу або окремих його частин.

1.4 Етапи розробки веб-сайту

Послідовність створення сайту й чітке опрацювання етапів – запорука успіху всього проекту. Можливо, користувачеві може здатися, що розробка сайту – нікчемна справа. Ну що тут складного – придумав дизайн, зверстав, наповнив його парою-трійкою текстів і готово. Насправді ж кожен етап створення сайту – це кропітка робота, за яку відповідальний кожен розробник з команди веб-студії.

Створення веб-сайту включає в себе не тільки розробку дизайну й програмування, а й детальний аналіз проекту, співпрацю з замовником і пошук рішень для досягнення поставлених цілей проекту.

Основними етапами створення веб-сайту є:

- постановка цілей і завдань сайту;
- створення, опрацювання технічного завдання (ТЗ) на розробку сайту;
- прототипування;
- створення макета дизайну сайту;
- верстка;
- програмування;
- наповнення контентом;
- тестування;

- здача готового проекту клієнтові.

Далеко не кожен клієнт бачить відмінності у видах веб-сайту, здатний самостійно визначити його цілі й завдання. Він звертається до професіоналів із проханням створити проект. Команда-розробник, у свою чергу, проводить підготовчий етап – аналізує, для чого клієнтові сайт, які завдання він повинен вирішувати: виконувати функцію представництва компанії в Інтернеті або ж продавати товар і послуги, стати якісною рекламою компанії або бути стартапом для молодшої фірми. Все це обговорюється із замовником. Він повинен сам розуміти навіщо йому сайт, в чому він допоможе його компанії і з чим впоратися не зможе.

Після того як поставлені завдання, визначається цільова аудиторія. Важливо зрозуміти для кого створюється ресурс, що замовник може запропонувати і як відреагує аудиторія на нього. Визначення цільової аудиторії «диктує» свої правила, на основі отриманих даних формується структура сайту, контент, дизайн, функціонал тощо.

Далі проводиться маркетинговий аналіз, до якого включено аналіз конкурентів, сезонності товару або послуги, діяльності та ніші, конкурентоспроможності пропозиції. Визначаються зачіпки й фішки, які допоможуть привернути увагу відвідувачів. По завершенню розробляється прототип сайту й демонструється клієнту.

Наступний етап розробки сайту – формування технічного завдання. Технічне завдання – це документ, який складається разом з клієнтом. Він включає детальні характеристики ресурсу:

- побажання по дизайну (кольорове оформлення, застосування фірмового стилю або його розробка, співвідношення графічних елементів до текстових);
- семантичне ядро;
- структуру сайту, кількість сторінок, категорій, блоків;

- функціонал сайту (тільки стандартний або розробка додаткових модулів, їхній докладний опис та цілі);
- застосовувані технології;
- технічні вимоги до ресурсу.

Технічне завдання – це основа, на яку покладається кожен фахівець, який бере участь у розробці. Тому в складанні ТЗ участь замовника є необхідною. Воно обумовлюється й редагується до тих пір, поки клієнт дасть згоду та підпише документ. Тільки після цього проект переходить у стадію створення [так само].

На етапі формування й обговорення ТЗ також розробляється структура сайту, його навігація, кількість категорій і підкатегорій, їхня послідовність розміщення тощо. Вся інформація також включається в документ, який підписується замовником.

Всі етапи створення сайту важливі для якісного функціонування ресурсу, але розробка дизайну – один з головних. Адже дизайн – це те, що бачить відвідувач в першу чергу, оцінює його й приймає рішення: залишитися на сторінці або закрити вкладку браузера.

Дизайнер опирається на технічне завдання, малює кнопки, банери та інші графічні елементи. Іншими словами, той прототип, який був створений на першому етапі розробки сайту, отримує естетичний зовнішній вигляд, створюється в кольорах, обраних замовником. Якщо компанія має корпоративний стиль, то дизайн розробляється згідно нього. Або ж може спочатку розробити фірмовий стиль, а після – дизайн на його основі.

Важливо відзначити, що дизайнер малює дизайн не кожної сторінки, а шаблони кількох основних, використовуючи тенденції веб-дизайну. Готовий макет передається клієнту і очікує свого схвалення. Якщо необхідні доопрацювання, дизайнер виконує їх і знову показує макет замовнику. Дизайн допрацьовується доти, поки він не буде затверджений.

Наступний етап – злиття дизайну з системою управління сайту (верстка).

Верстальник за допомогою мови HTML переводить готовий дизайн в робочий проект. Ресурс отримує життя, стає динамічним, всі кнопки працюють. Ресурс правильно відображається у всіх існуючих Інтернет-браузерах. На цьому ж етапі створюються стилі CSS.

Далі ресурс підключається до системи управління. З її допомогою створюються сторінки, категорії та підкатегорії проекту, визначається порядок роботи вкладки меню, розставляються гіперпосилання. Ресурс отримує остаточну структуру, яка буде представлена відвідувачам. Іноді сайти можуть використовуватися без CMS (наприклад, у Лендінзі), тому сайт просто верстається HTML і CSS мовами.

На цьому ж етапі верстаються категорії та підкатегорії, контент, який буде розміщений на сторінці. Велика увага приділяється верстці картки товару для Інтернет-магазину, яка повинна відповідати технічним і призначеним для користувача вимогам.

На завершальному етапі верстки розробляється адаптивна версія сайту. Вона не обов'язкова, але рекомендована клієнтам, які бажають охопити мобільний трафік.

На етапах створення простого сайту (наприклад, односторінкового) в програмуванні немає необхідності. Програміст підключається до роботи, якщо розробляється великий і складний проект. Завданням фахівця є розробка складних функцій: калькулятора, системи оплати за допомогою банківських карт, конвертер валюти, тобто всіх тих функцій, які не входять в стандартний пакет функціоналу.

Передостанній етап створення веб-сайту – наповнення сторінок графічним та інформаційним контентом. Розміщується відео, фото, тексти та інша інформація, яку зможе побачити або прочитати відвідувач.

На основі семантичного ядра пишуться SEO-статті (якщо передбачається просування ресурсу SEO-методом), контент-менеджер розміщує графічні елементи в логічній структурі. Сторінки проходять внутрішню оптимізацію.

Найважливішу роль в процесі створення сайту грає його тестування. Виконує цю роботу тестувальник, який перевіряє функціональність ресурсу згідно ряду критеріїв і виявляє помилки, які повинні бути усунені.

Також необхідним є вибір домену та хостингу, але він може проводитися в будь-який момент процесу створення ресурсу. По завершенню роботи готовий проект передається замовнику разом з логінами й паролями до системи управління. Після здачі проекту клієнт може замовити послугу просування та підтримки проекту, сплативши їхню вартість.

Рідко після того, як паролі та доступ до сайту видані замовнику, співпраця з розробником закінчується. Найчастіше клієнти замовляють додаткові послуги, наприклад:

- оновлення каталогу, матеріалів, викладка товарів та їхній опис;
- навчання роботі з сайтом;
- підтримка сайту;
- просування ресурсу;
- написання статей, наповнення категорій SEO-текстами тощо.

Важливо відзначити, що все поетапне створення сайту контролюється замовником. По закінченню кожного етапу готовий результат демонструється клієнту й узгоджується з ним. Це допомагає замовнику контролювати роботу розробника, а розробнику розуміти, чи задоволений клієнт результатами його роботи.

2 АНАЛІЗ ЗАДАЧ, ШЛЯХІВ ЇХНЬОГО ВИРІШЕННЯ, КОНКУРЕНТІВ

2.1 Аналіз задач

Нами була отримана комерційна задача з розробки веб-сайту від приватного підприємства, що надає послуги з оренди лімузинів, яхт та інших послуг для гостей Одеси. На сайті повинна бути надана повна інформація з описом діяльності підприємства та список послуг. Важливим фактом є те, що більша частина клієнтів є гостями Одеси, тому сайт має бути англійською мовою. Наступна проблема полягає в тому, що під час подорожей більшість людей для Інтернет серфінгу використовує мобільні пристрої. Оскільки в наш час їхній вибір є досить великим, то й сайт має працювати на великому спектрі різних девайсів. Важливим недоліком є те, що процес створення заявки на оренду не автоматизований. Для створення заявки потрібно завжди бути на зв'язку та власноруч записувати дані клієнтів й те, що вони замовляють. Цей процес можна автоматизувати за допомогою додавання на сайт можливості залишати свої дані, формувати власний запит на послуги та формувати з них звіт для підприємства.

2.2 Аналіз конкурентів

Наступним кроком є пошук вже готових рішень для схожих підприємств. Для цього використовується пошукова система «Google» й було сформовано список запитів, за якими можливо знайти подібні сайти. Перше, що було виявлено, – це досить застарілий дизайн сайтів конкурентів й те, що більшість не оптимізовані під мобільні пристрої. Для аналізу оптимізації сайтів нами використовувався сервіс «PageSpeedInsight», який дозволяє виявити основні помилки при розробці сайтів та надає поради щодо їхнього вирішення. У більшості конкурентів було виявлено наступні проблеми:

- використовуються застарілі формати зображень;
- час завантаження першого контенту через завантаження великої кількості зображень одночасно;
- застарілий дизайн;
- неприємна користувачеві кольорова гамма;
- відсутність можливості автоматично сформулювати запит;
- більшість сайтів є звичайними інформаційними ресурсами без можливості замовити послуги на самому сайті.

2.3 Аналіз шляхів вирішення задачі

Згідно виявлених проблем, можна визначити такі шляхи вирішення задачі:

- використання зображення у форматах JPEG 2000, JPEGXR і WebP, тому що в них використовується більш ефективне стиснення, тому вони завантажуються швидше й споживають менше трафіку, ніж зображення PNG і JPEG;
- використання принципу lazy loading для прихованих зображень після того, як всі важливі ресурси будуть завантажені. Таким чином, це зменшить час до початку взаємодії;
- знаходження основних трендів та напрямів веб-дизайну та обрати ті, що є найбільш близькими до тематики сайту. Це дозволить вирішити проблему застарілого дизайну. У ході дослідження ми обрали наступні:
- «Більше простору». Тренд на великий простір з'явився досить нещодавно – у 2016-2017 роках. Однак він надійно осів у веб-дизайні завдяки своїй здатності концентрувати увагу на головному. Великий простір візуально збільшує екран, не дозволяє користувачеві втратити концентрацію.

- «Кричущі кольори». Використання яскравих кольорів в дизайні сайту, якщо вони підкреслюють загальну ідею. Кольори передають емоції, а емоції – найголовніше для користувача. Побачивши соковиті й кричущі кольори, які правильно підібрані один до одного, цільова аудиторія належним чином оцінить таку сміливість, залишившись на сайті й продовживши користування ним.
- «Повноекранне відео». Хоча інфоконтент дозволяє максимально розкрити компанію, її конкурентні переваги та послуги, повноекранні відео можуть впоратися з цим не гірше. По-перше, фонові відео роблять дизайн сайту незвичайним, сучасним, а також це збільшує залученість відвідувача й призводить до більшої ефективності проекту. По-друге, відео можуть стати гарною історією про компанію. При цьому користувачеві не потрібно гортати сторінку вниз, щоб знайти потрібну йому інформацію.
- «Напівпрозорі кнопки». Такі кнопки так само, як і непрозорі, відмінно виконують свою функцію лідогенерації, але при цьому виглядають дуже оригінально й не перевантажують сайт.
- додавання на сайт кошику, завдяки чому користувачі зможуть обрати й додати в нього бажані послуги;
- додавання форми, в якій користувачі зможуть залишити свої контактні дані.

Останні два пункти спрямовані на вирішення проблеми автоматизації на сайті.

2.4 Аналіз доступного ПЗ

2.4.1 Аналіз варіантів ОС

Хостинг, на якому буде знаходитися сайт, працює на ОС «Debian», тож для розробки раціонально вибрати найбільш наближену ОС. «Debian» –

операційна система, що складається з вільного ПЗ з відкритим вихідним кодом. У даний час Debian GNU / Linux – один з найпопулярніших і важливих систем GNU / Linux, в первинній формі зробив значний вплив на розвиток цього типу ОС в цілому. Debian може використовуватися в якості операційної системи як для серверів, так і для робочих станцій. Debian став основою цілого ряду дистрибутивів. Найвідоміші з них (в алфавітному порядку) – antiX, Knoppix, Linux Mint, Maemo, SteamOS, TAILS, Ubuntu. Найпростіший спосіб вибрати ОС – подивитися на їхній рейтинг. У рейтингу серед похідних від «Debian» можна виділити дві ОС: «Mint» та «Ubuntu», які займають 3 та 5 місце в рейтингу.

Оскільки обидві ОС є похідними від «Debian», вони є досить схожими й одним з основних відмінностей є інтерфейс, з яким вони поставляються. У «Mint» це «Cinamon» а у «Ubuntu» – «Unity». «Unity» – досить ресурсоємний інтерфейс, в той час як «Cinamon» не використовує багато ресурсів і є досить наближеним до інтерфейсу «Windows» який є звичним для більшості користувачів ПК.

2.4.2 Аналіз варіантів ПЗ для веб-серверів

Серед найбільш популярних веб-серверів можна віділити Apache та nGinx.

Apache HTTP Server був розроблений Робертом Маккула в 1995 році, а з 1999 року розробляється під управлінням Apache Software Foundation – фонду розвитку програмного забезпечення Apache. Оскільки це HTTP сервер перший і найпопулярніший проект фонду, його зазвичай називають просто Apache.

Веб-сервер Apache був найпопулярнішим веб-сервером в Інтернеті з 1996 року. Завдяки його популярності у Apache сильна документація та інтеграція зі стороннім софтом.

Адміністратори часто вибирають Apache через його гнучкості, потужності й широку поширеність. Він може бути розширений за допомогою системи динамічно завантажуваних модулів і виконувати програми на великій кількості різних мов програмування без використання зовнішнього програмного забезпечення.

У 2002 році Ігор Сисоєв почав роботу над Nginx для того, щоб вирішити проблему C10K – вимога до ПЗ працювати з 10 тисячами одночасних з'єднань. Перший публічний реліз був випущений в 2004 році, поставлена мета була досягнута завдяки асинхронної event-driven архітектурі.

Nginx почав набирати популярність з моменту релізу завдяки своїй легкій вазі (light-weight resource utilization) й можливості легко масштабуватися на мінімальній залозі. Nginx чудовий при віддачі статичного контенту і спроектований так, щоб передавати динамічні запити іншому ПЗ, призначеному для їх обробки.

Адміністратори часто вибирають Nginx через його ефективне споживання ресурсів і чуйності під навантаженням, а також через можливість використовувати його і як веб-сервер, і як проксі.

Однак тому, що на хостингу використовуються веб-сервер Apache, раціонально використовувати його.

2.4.3 Вибір мови програмування серверної частини сайту

Нижче на рис. 2.1 [24]¹⁾ показано найбільш широко використовувані мови програмування на сервері сьогодні.

¹⁾[24] Percentages of websites using various server-side programming language. URL: <https://i2.wp.com/www.pixelcrayons.com/blog/wp-content/uploads/2018/01/Usage-Statistics-and-Market-Share-of-Server-side-Programming-Languages-for-Websites-January-2018.png?w=438&ssl=1> (дата звернення: 30.03.2019).

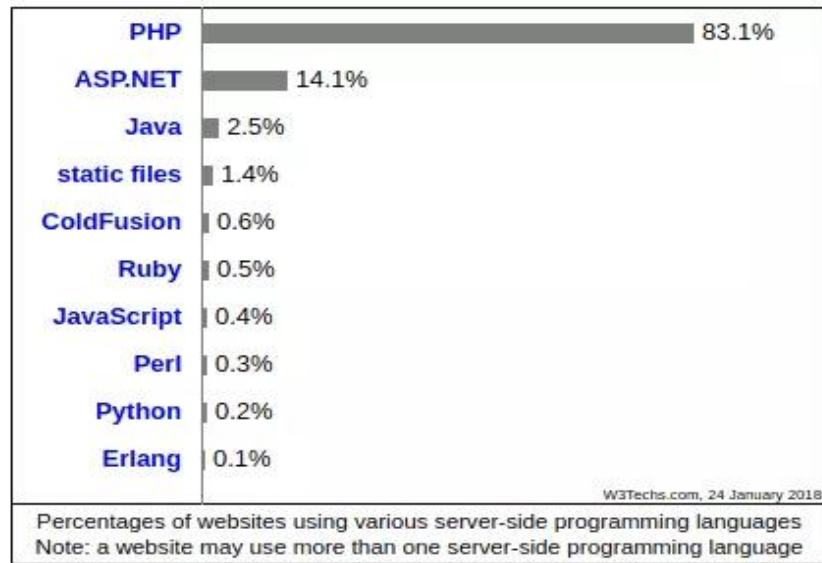


Рисунок 2.1 – Найбільш широко використовувані мови програмування на сервері

Серед можливих і популярних мов програмування для серверної частини веб-сайтів виділяються 3 це PHP, Ruby та Python. Статистика використання та частка ринку будь-якої мови програмування дає вам уявлення про популярність цієї мови серед мас.

Юзабіліті мови програмування є одним з найбільш важливих факторів, які ми повинні розглянути, перш ніж вибрати його для нашого проекту розвитку сайту. Майже всі мови програмування формують інтерфейс між комп'ютером і користувачем. Мова з високою зручністю у використанні допомагає розробити більш потужне, масштабоване та високоефективне програмне забезпечення. За цим критерієм виділяється Python завдяки своєму спрощеному синтаксису та низькому порогу входу.

Без сумніву, PHP є однією з найпопулярніших мов програмування в світі сьогодні. Це – одна з найстаріших мов – набрала багато лояльних кодерів і клієнтську базу. Інші дві мови – це. Веб-розробка Ruby і Python є відносно новими, отже, менш популярні серед мас.

Нижче на рис. 2.2[26]¹⁾ показано зображення індексу програмування ТЮВЕ, яке вимірює рівень популярності різних мов, створених та підтримуваних компанією ТЮВЕ, що знаходиться в Нідерландах.

Порівняння продуктивності будь-якої мови програмування має вирішальне значення.

Високоєфективна мова допомагає створювати масштабовані, безпечні та швидкі програми. Ці два параметри допомагають зрозуміти та порівняти продуктивність цих трьох мов.

В інформатиці час виконання - це час, протягом якого виконується програма, на відміну від інших етапів життєвого циклу програми, таких як час компіляції, час встановлення зв'язків і час завантаження. Коли програма повинна бути виконана, завантажувач спочатку виконує необхідну настройку пам'яті і динамічно пов'язує програму з бібліотеками, в якій вона потребує, а потім виконання починається з точки входу в програму. У деяких випадках мова або реалізація виконуватимуть ці завдання за рахунок середовища виконання мови, але це нетипово для найбільш поширених мов на основних користувацьких операційних системах.

¹⁾ [26] PHP vs Python vs Ruby. Popularity. URL: <https://i0.wp.com/www.pixelcrayons.com/blog/wp-content/uploads/2017/02/php-vs-python-vs-ruby-popularity.png?ssl=1> (дата звернення: 30.03.2019).

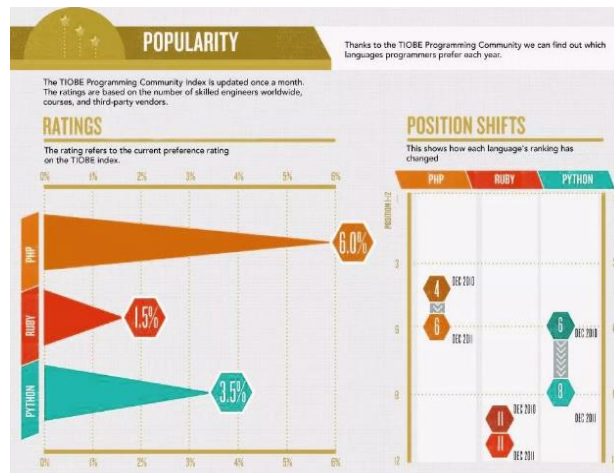


Рисунок 2.2 – Зображення індексу програмування ТІОВЕ

На зображеному нижче рис. 2.3 [27]¹⁾ показано середній час виконання і рядки кодів усіх цих трьох мов.

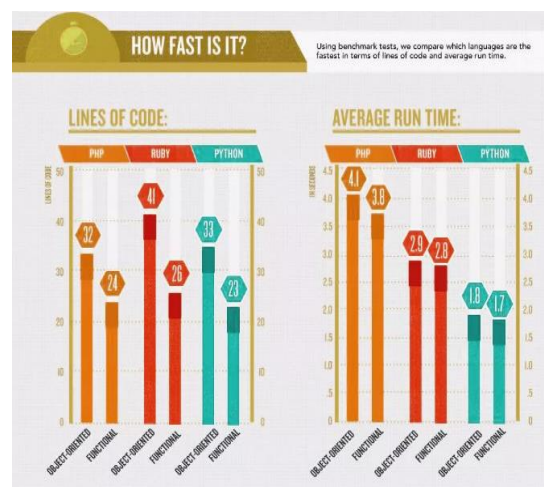


Рисунок 2.3 – Порівняння продуктивності мов програмування PHP, Ruby та Python

¹⁾ PHP vs Python vs Ruby. Usability. URL: <https://i0.wp.com/www.pixelcrayons.com/blog/wp-content/uploads/2017/02/php-vs-python-vs-ruby-usability-performance.png?ssl=1> (дата звернення: 30.03.2019).

Обговорюваність програмування допомагає багатьма способами. Якщо будь-яка мова популярна або обговорюється найбільше, то вона матиме велику спільноту, яке допоможе вирішити ваш запит за мінімальний час. Крім того, найбільш обговорювана мова також має хорошу підтримку. Рис. 2.4 [25]¹⁾ показує статистику, пов'язану з найпопулярнішою мовою програмування.

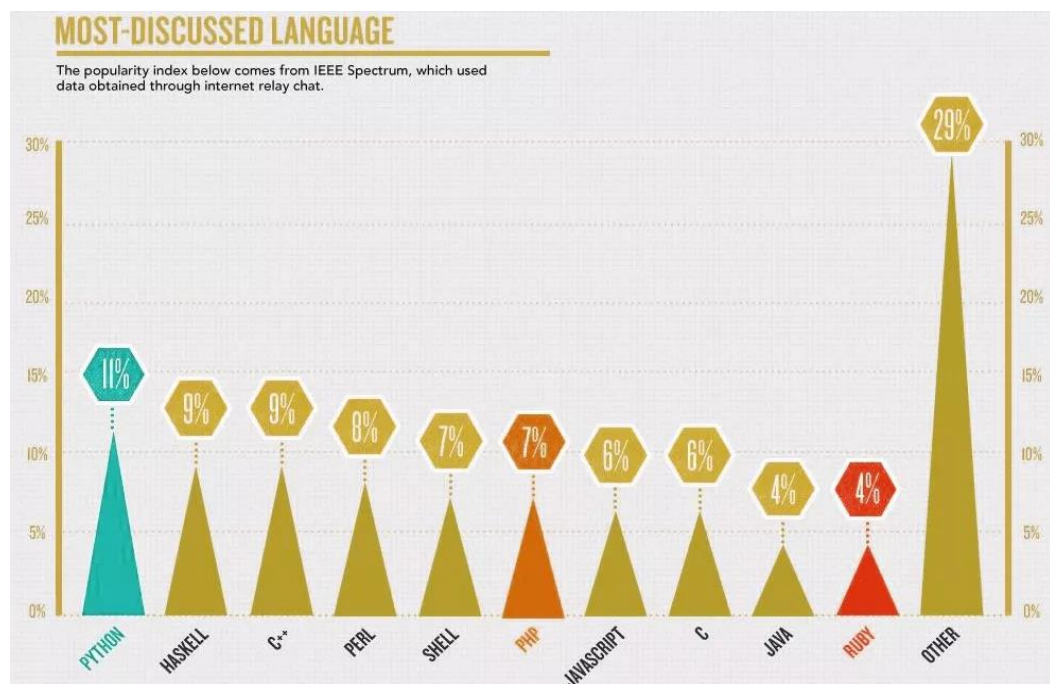


Рисунок 2.4 – Найобговорюваніші мови програмування.

Оскільки РНР є найбільш широко використовуваною мовою та досить обговорюваною то має і більше готових рішень тож раціонально використовувати РНР.

2.4.4 Аналіз додаткового ПО

¹⁾ PHP vs Python vs Ruby. Most discussed language. URL: <https://i0.wp.com/www.pixelcrayons.com/blog/wp-content/uploads/2017/02/php-vs-python-vs-ruby-most-discussed-language.png?ssl=1> (дата звернення: 30.03.2019).

Під додатковим ПО розуміється використання IDE або текстового редактора. Оскільки на сьогодні є досить розвинені текстові редактори які задовольняють основні потреби при розробці простих сайтів то використання ресурсоемних IDE не раціонально. Серед текстових редакторів можливо виділити Atom, SublimeText та Notepad++.

Atom є чудовим інструментом, тому що дозволяє налаштувати свій редактор, присутня досить детальна документація, підсвічування синтаксису, велика кількість плагінів та інтеграція з GitHub. Найбільшим недоліком є проблеми з продуктивністю.

SublimeText текстовий редактор який має підсвічування синтаксису, велику кількість плагінів та приємний дизайн. Недоліком є те що він є умовно безкоштовний.

Notepad++ є досить швидким текстовим редактором однак він є застарілим та не має такої бази плагінів та доповнень як його конкуренти.

2.5 Аналіз готових рішень

2.5.1 Аналіз фронт-енд фреймворків

Серед фронт-енд фреймворків виділяються Bootstrap, Skeleton, Semantic UI та Slides. Bootstrap один з найвідоміших CSS-фреймворків на сьогоднішній день. Має в своєму складі шаблони для відтворення кнопок, сайдбарі, навігаційних панелей, форм та інших елементів сайту. Включає себе JavaScript-розширення.

Основні інструменти bootstrap – шаблони, @media, форми, навігація, АЛЕРТ, типографіка і звичайно ж, сітки. Bootstrap сумісний з усіма основними сучасними браузерами, але в старих версіях браузерів можуть бути проблеми. Підтримує адаптивність. Використовує мови Less і Sass.

Skeleton заснований на JavaScript і CSS, використовується для швидкого і безпомилкового створення адаптивних сайтів які коректно зображаються як на великих моніторах, так і на екранах маленьких гаджетів. Великий плюс Skeleton – його можна пристосувати під будь-який дизайн. Сітка розмітки містить 12 колонок і має базову ширину 960 пікселів. Сумісний з усіма основними браузерами.

Semantic UI має підтримку Firefox, Chrome, Safari, Інтернет Explorer, Android 4, Blackberry10. Містить велику кількість компонентів: іконки, зображення, написи, меню та інші стандартні компоненти. Підтримує останні версії HTML і CSS і має хорошу добірку скриптових модулів і API. Використовує Less. За застосовності змагається з bootstrap. Є варіант російською мовою, але неповний, а організатори проекту пропонують допомогти в перекладі

Slides дуже потужний інструмент для розробки клієнтської частини сайти, має набір готових шаблонів, готових рішень для сучасного веб-програмування на кшталт форм, діалогових вікон, слайдерів, детальну документацію, набір анімацій, є кросс-браузерним. Він в достатній мірі охоплює всі потреби при розробці проекту і єдиним мінусом є те що він платний. Оскільки в компанії мені оплатили його, то він є чудовим вибором.

2.5.2 Аналіз CMS

Так як вибраною мовою програмування став PHP, він за свою історію обзавівся досить великою кількістю готових рішень від фреймворків на кшталт laravel і Symphony до CMS.

Оскільки більшість задач поставлених перед нами є типовими то вигідніше використовувати CMS. Перелік CMS написаних на PHP є досить великим, виберемо основні з них це Wordpress, Opencart, 1C-bitrix, Joomla та Drupal.

WordPress – система керування вмістом сайту з відкритим вихідним кодом; написана на PHP; сервер бази даних – MySQL; випущена під ліцензією GNU GPL версії 2. Сфера застосування – від блогів до досить складних новинних ресурсів і Інтернет-магазинів. Вбудована система «тем» і «плагінів» разом з вдалою архітектурою дозволяє конструювати проекти широкої функціональної складності. Сильні сторони:

- безкоштовність;
- швидка та проста установка;
- гнучка система плагінів та шаблонів;
- готова адміністративна панель.

OpenCart – e-commerce platform, орієнтована на створення Інтернет-магазинів. Є вільним програмним забезпеченням, поширюваним за ліцензією GNU General Public License v3. Системою «OpenCart» підтримуються доповнення.

E-commerce platform «OpenCart» була написана в 1998 році Крістофером Манном для Walnut Creek CDRом. Перший публічний реліз відбувся 11 травня 1999 року. Розроблений на мові Perl, спочатку проект розвивався слабо і остаточно був покинутий в 2000 році, коли Манн заявив, що він більше не може розвивати OpenCart, так як у нього є інші зобов'язання.

Друге життя система знайшла завдяки британському розробнику Деніелу Керр, який використовував напрацювання Манна для створення свого власного движка на PHP. Перший реліз оновленого OpenCart відбувся 10 лютого 2009 року – Керр виклав свою систему на Google Code під індексом 1.1.1.

У вересні 2014 року OpenCart став найпопулярнішим рішенням для Інтернет-комерції в Китаї, а за станом на серпень 2015 року на OpenCart працювало 6,42% всіх Інтернет-магазинів світу. За цим показником OpenCart став третім в світі, відставши тільки від WooCommerce і Magento і випередивши OSCommerce, ZenCart і Shopify.

У жовтні 2014 року побачила версія 2.0, основними відмінностями якої від версії 1.5 стали:

- HTML 5;
- адаптивний дизайн на Twitter Bootstrap;
- іконки і шрифти Font Awesome;
- установник модулів;
- вбудований модифікатор ocmод замість популярного стороннього модуля vqmod в попередніх версіях.

На офіційному сайті вказується, що після установки програмного забезпечення необхідно тільки додати продукти і, при необхідності, замінити шаблон оформлення сайту; корзина товарів окремої настройки не вимагає і працює відразу. В адміністративній панелі є можливість управляти замовленнями і доступно управління декількома платіжними сервісами.

Як переваги програмного забезпечення офіційний сайт відзначає наступні пункти:

фвідкритість вихідного коду;

- документованість ПО;
- необмеженість категорій, продуктів і їх виробників;
- необмеженість інформаційних сторінок;
- підтримка багатомовності та перекладу інтерфейсу;
- можливість встановлювати власні теми;
- вбудовані модулі:
- відгуки клієнтів;
- система рейтингу продуктів;
- система зміни розміру зображень;
- система відображення супутніх продуктів;
- система знижок і купонів;
- система вибору способу доставки.
- можливість вказати кілька податкових ставок;

- можливість вказати вагу продукту без і з упаковкою і динамічно
- що розраховується вартість;
- пошукова оптимізація;
- необмежена модульна система, для створення декількох
- магазинів на одній платформі;
- інструменти резервного копіювання та відновлення;
- звіт про помилки;
- реєстрація помилок.

Також, в OpenCart є ряд недоліків:

- складно додавати нові функції для різних версій. Процес установки і видалення модулів організований не зовсім зручно. При установці завжди потрібно стежити за сумісністю модуля з версією движка;
- проблеми з оновленнями. Після зміни версії, виходять з ладу деякі плагіни, що негативно позначається на роботі всього сайту;
- періодично візуальний редактор самостійно створює зайвий код, який ускладнює роботу ІТ-фахівцям, ускладнює і погіршує індексацію сайту пошуковими системами;
- незручно влаштована функція завантаження картинок. Немає можливості завантажувати картинки пакетом по кілька штук, сортування зображень відбувається тільки в довільному порядку, пошук по зображеннях досить проблематичний, що ускладнює завдання заміни фото до картки товару, якщо на сайті їх багато;
- відсутня можливість виставити автоматичне прописування ЧПУ. У даній CMS ЧПУ прописуються вручну кожен раз після редагування вже доданих товарів або «заливання» нових позицій;
- виникають проблеми при зміні шаблону. Оскільки OpenCart має досить складну структуру коду, зміна шаблону може викликати збої в роботі окремих модулів і всього сайту в цілому;

- сторінка товару закривається після натискання клавіші збереження і якщо потрібно внести ще якісь правки доводиться знову шукати потрібний товар і відкривати його;
- занадто багато зайвих полів при оформленні замовлення. Причому, дані поля розбиті на спойлери. І змінити це, не коригуючи саму систему на технічному рівні, просто не можна.

Розробник може доповнити стандартний код такими рішеннями, які в повній мірі нівелюють вплив всіх негативних факторів. Таким чином почали з'являтися збірки на основі OpenCart. Основна мета таких збірок – спрощення взаємодії з CMS на стороні адміністратора і клієнта. Для цього розробники покращують функціонал вже готового движка, додають різні модулі, працюють над дизайном.

1С-Бітрікс – CMS (система управління контентом веб-проекту) від російської компанії «1С-Бітрікс».

10 лютого 2003 вийшла «Бітрікс: Управління сайтом 3.0» – перша комерційно успішна версія CMS від «Бітрікс».

«Більшість використовувало» самопісний «софт, а ми запропонували універсальне рішення з управління контентом на сайті (CMS)» засновник компанії Сергій Рижиков в інтерв'ю Forbes Russia №03 (168) 2018

За допомогою CMS будь-який користувач міг створити Інтернет-магазин, корпоративний сайт або новинний портал.

У 2006 році кількість партнерів «Бітрікс» склало 1000. У цей час на «Бітрікс: Управління сайтом» вже працювали:

- «Ельдорадо»;
- «Ростелеком»;
- Independent Media;
- «Аргументи і факти»;

У квітні 2007 року «1С» і «Бітрікс» створили спільне підприємство «1С-Бітрікс».

«1С-Бітрікс: Управління сайтом» дозволяє створювати:

- інтернет магазини;
- корпоративні сайти;
- інформаційні портали;
- сторінки спільнот, форуми;
- рекламні Лендінги.

«1С-Бітрікс: Управління сайтом» також включає набір інструментів для просування веб-проектів:

- E-mail маркетинг;
- інструменти комунікації з клієнтами;
- веб-форми;
- опитування;
- веб-аналітика;
- 1С-Бітрікс BigData;
- А / В тестування;
- SEO модуль;
- підписки, розсилки;
- соціальні сервіси;
- композитний сайт.

16 квітня 2014 в «1С-Бітрікс: Управління сайтом» з'явилася запатентована технологія «Композитний сайт», яка, за словами розробників, «скорочує час відповіді сайту в сотні разів». Маркетплейс – платформа готових рішень на базі «1С-Бітрікс: Управління сайтом». Наприклад, модулі для сайтів або готові Інтернет-магазини. Станом на 1 жовтня 2018 року в Маркетплейсе розміщено понад 2000 готових рішень. Всі рішення розроблені партнерами «1С-Бітрікс».

Joomla – система управління вмістом (CMS), написана на мовах PHP і JavaScript, що використовує в якості сховища бази даних СУБД MySQL або

інші стандартні промислові реляційні СУБД. Є вільним програмним забезпеченням, поширюваним під ліцензією GNU GPL.

Система управління вмістом Joomla! є відгалуженням широко відомої CMS Mambo. Команда незалежних розробників відокремила від проекту Mambo через незгоду в економічній політиці. 16 вересня 2005 року в світ вийшла перша версія Joomla!, що є по суті перейменованої Mambo 4.5.2.3 і включає в себе виправлення знайдених на той момент помилок і вразливостей.

До літа 2008 року по числу щоденних завантажень Joomla! посіла друге місце після WordPress зі значним відривом від інших подібних систем.

Версія 1.0 вважається застарілою, її офіційна підтримка припинена 1 липня 2009 року.

Підтримка версії 1.6 припинено 19 серпня 2011 року.

Підтримка версії 1.7 припинено 24 лютого 2012 року.

Підтримка версії 1.5 припинено 27 вересня 2012 року.

Підтримка версії 2.5 тривала до кінця 2014 року.

Поточна версія системи – 3.9.6, випуск якої відбувся 7 травня 2019 року.

CMS Joomla! включає в себе мінімальний набір інструментів при початковій установці, який доповнюється в міру необхідності. Це знижує заха-
ращення адміністративної панелі непотрібними елементами, а також знижує навантаження на сервер і економить місце на хостингу.

Joomla! дозволяє відображати інтерфейс фронтальної та адміністративної частини будь-якою мовою. Каталог розширень містить безліч мовних пакетів, які встановлюються штатними засобами адміністрування. Доступні пакети російського, українського, білоруського та ще деяких мов пострадянського простору.

Функціональність можна збільшувати за допомогою додаткових розширень (компонентів, модулів і плагінів).

Є модуль безпеки для багаторівневої аутентифікації користувачів та адміністраторів (використовується власний алгоритм аутентифікації і «ведення» сесій).

Система шаблонів дозволяє легко змінювати зовнішній вигляд сайту: розташування модулів, шрифти і інше. Можна надати користувачам вибирати одне з декількох відображень. У мережі існує величезний вибір готових шаблонів, як платних, так і безкоштовних. Також існує програмне забезпечення для самостійного створення оригінальних шаблонів.

Передбачені настроюються схеми розташування модулів, включаючи лівий, правий, центральний і будь-яке інше довільне положення блоку. При бажанні вміст модуля можна включити в вміст матеріалу. Наприклад, вираз `{loadposition mod_fpslideshow}`, введене (разом з фігурними дужками) в довільне місце в статті, виведе вміст модуля, якому задана позиція виведення як «`mod_fpslideshow`».

До переваг системи можна віднести те, що всі компоненти, модулі, плагіни і шаблони можна написати самому, розмістити їх в структурованому каталозі розширень або відредагувати існуюче розширення на свій розсуд.

Відбувається регулярний вихід оновлень. Існує публічний «баг-трекер» (система відслідковування помилок). (Див. Список офіційних трекерів.) Існують також трекери міграції зі старих версій Joomla!, трекер побажань розширення функціональності і так далі, де користувачі Joomla! можуть залишати зауваження з приводу роботи CMS, які згодом вивчаються її розробниками, при необхідності включають в чергове оновлення Joomla! виправлення, вирішальні ті чи інші проблеми.

Починаючи з версії 1.6 вбудована багатомовність.

Починаючи з версії 2.5 розширена підтримка баз даних. Реалізована підтримка Microsoft SQL Server, а з версії 3.0 – PostgreSQL. Надалі планується додати підтримку Oracle, SQLite.

Структурно система являє собою ядро, до якого приєднуються компоненти. Компоненти можуть складатися по функціональному призначенню з модуля, плагіна і / або шаблону:

- модулі – елементи, які відображаються на сторінці користувачів у вигляді окремих блоків;
- плагіни – елементи, які відображаються на адміністративній панелі інструментів і надають додаткову функціональність;
- шаблони – елемент, який відповідає за уніфіковане оформлення всього сайту.

В ядро Joomla! входить:

- блок (функціональність визначається HTML-кодом всередині нього) і його різновиди з розширеними можливостями управління з адміністративної панелі: банери (модулі для розміщення реклами), контакти (блок для розміщення контактної інформації та форми зворотного зв'язку), стрічка новин (RSS підписка);
- матеріал (стаття) з мітками (ключові слова) і категоріями (папками) для навігації користувачів, редактором для зручного оформлення матеріалу.

Не входять в ядро вільні сторонні розширення:

- плагін для резервного копіювання (backup) і відновлення, наприклад, Akeeba Backup.

Створення власного розширення здійснюється за допомогою написання плагіна на мові PHP з використанням для взаємодії API Joomla !

Drupal – система управління вмістом (CMS), яка використовується також як каркас для веб-додатків (CMF), написана на мові PHP і використовує як сховище даних реляційну базу даних (підтримуються MySQL, PostgreSQL та інші). Drupal є вільним програмним забезпеченням, захищеним ліцензією GPL, і розвивається зусиллями ентузіастів зі всього світу.

Почав розробку 2000 року бельгієць Дріс Бейтарт (Dries Buytaert), який і понині є керівником проекту. Назва drupal є спотвореним вимовою голландського слова druppel (крапля) і з'явилося в результаті забавної помилки. У 2000 році кілька студентів Антверпенського університету використовували загальний ADSL-модем для доступу в Інтернет, і для спілкування користувачів цієї маленької локальної мережі Дріс створив внутрішній сайт. Програма, на якій працював сайт, не мала імені, поки Дріс не зібрався реєструвати доменне ім'я для нового сайту, і в результаті помилки замість dorp.org (від голландського dorp – село, цілком доречне ім'я для невеликого співтовариства) був зареєстрований drop.org (англ. drop – крапля). Аудиторія нового сайту з часом змінилася на які цікавляться розвитком нових веб-технологій, і новий сайт поступово перейшов з любительського експерименту в проект з обговорень ідей веб-програмування. У 2001 році Дріс опублікував вихідний код drop.org під ім'ям «Drupal». З тих пір розробка перемістилася на зареєстрований пізніше сайт drupal.org, який набув статусу офіційного сайту проекту.

З часу створення в процес розробки були залучені сотні людей. Офіційні розробники ядра зараз – це десятки людей, але багато людей надсилають свої патчі, що включаються в код системи. Також функціональність движка розширюється додатковими модулями, які створюють сторонні розробники. Наприклад, в релізі 4.6 взяли участь 50 осіб, що надіслали 338 поправок до коду, а в релізі 4.7 взяли участь вже 338 розробників, що надіслали приблизно 1500 поправок. За п'ять років існування проекту (до 2006 року) вийшло 13 релізів. На даний момент (січень 2011) було розроблено понад 7000 додаткових модулів, що розширюють базові функції CMS. За статистикою звернення до оновлень движка, на березень 2009 року в Інтернеті – в межах 120 тис. Активних сайтів на Drupal 6 (без урахування сайтів, що знаходяться за файрволлами, сайтів з відключеним механізмом оновлень). Це також не враховує сайтів, що працюють на ранніх версіях Drupal.

У 2007 році Дріс Бейтарт створив фірму Asquia, спрямовану на комерційну підтримку проектів і сервісів, заснованих на Drupal, натхненний успіхами Canonical і RedHat, що побудували бізнес на підтримку відкритих проектів. У 2008 Бейтарт отримав докторський ступінь в галузі інформатики та приладобудування («Computer Science and Engineering»).

Згодом навколо міжнародного сайту з'явилися національні спільноти, функціонують понад 20 національних сайтів підтримки на різних мовах.

Архітектура Drupal дозволяє застосовувати його для побудови різних типів сайтів – від блогів і новинних сайтів до інформаційних архівів або соціальних мереж. Наявну за замовчуванням функціональність можна збільшувати підключенням додаткових розширень – «модулів» в термінології Drupal. Найбільш важливі функції, що надаються Drupal «з коробки»:

- єдина категоризація всіх видів вмісту (таксономія) – від
- форумних повідомлень до блогів і новинних статей;
- широкий набір властивостей при побудові рубрикаторів: плоскі
- списки, ієрархії, ієрархії із загальними предками, синоніми,
- споріднені категорії;
- вкладеність категорій будь-якої глибини;
- пошук по змісту нашого сайту, в тому числі пошук по таксономії і користувачам;
- розмежування доступу користувачів до матеріалів (рольова модель);
- динамічну побудову меню;
- підтримка XML-форматів:
- висновки документів в RDF / RSS;
- агрегація матеріалів з інших сайтів;
- BlogAPI для публікації матеріалів за допомогою зовнішніх програм;
- авторизація через OpenID;
- символічні осмислені URL (інакше «людино-зрозумілі» – ЧПУ);
- переклади інтерфейсу сайту на різні мови, а також підтримка;

- ведення різномовного контенту;
- можливість створення сайтів з пересічним вмістом (наприклад загальною базою користувачів або загальними налаштуваннями);
- роздільні конфігурації сайту для різних віртуальних хостів (мультисайтінг), в тому числі власні набори модулів і тем оформлення для кожного підсайту;
- повідомлення про виходять оновлення модулів.

Критики Drupal ставлять в докір розробникам слабе використання об'єктних можливостей PHP. Дійсно, API Drupal практично не використовує наявні в PHP можливості ООП. Розробники аргументують це слабкою реалізацією ООП в мові (особливо до версії PHP 5). Об'єктна модель в Drupal присутній, але в дещо нетрадиційному для PHP вигляді. В Drupal 7 проте присутній ряд нововведень, що задіюють об'єктні можливості PHP 5. А у версії Drupal 8 офіційно прийнято використання ООП. Однак, в співтоваристві присутні вельми суперечливі думки з даного приводу.

До недоліків (але одночасно і до достоїнств) Drupal можна віднести відсутність зворотної сумісності API. У кожному новому великому релізі відбуваються великі зміни API, коли поряд з додаванням нових функцій прибираються деякі старі або змінюються параметри виклику існуючих. Це призводить до необхідності розробникам сторонніх модулів адаптувати їх для роботи з новими версіями Drupal. Однак зміни API і процедура адаптації модулів до нових версій описуються в документації для кожного релізу, також завжди пропонується механізм автоматизованого апгрейда ядра системи на нову версію. Плюс даної схеми розробки – немає необхідності тягти з версії у версію програмний шар сумісності зі старими API, що полегшує поточний код системи. Дизайн сайту на Drupal будується на основі змінних тем оформлення. Немає єдиної схеми побудови дизайну. Натомість Drupal дає можливість використовувати різні «рушії тем», що використовують шаблони, зручні для редагування (шаблони XML у рушіях xtemplate і Smarty або шаб-

лони на HTML і вбудований PHP у рушії phptemplate тощо), або створювати теми оформлення безпосередньо звертаються до API Drupal. У комплект поставки Drupal включений рушій тем на основі phptemplate і кілька прикладів тем. Інші пакети тем можна завантажити на сайті проекту.

3 ПРОЕКТНА ЧАСТИНА

3.1 Загальна структура сайту

На головній сторінці відображається коротка інформація про сайт з можливістю залишити свої контактні дані для отримання зворотного зв'язку.

На сторінці «Limousines» надано повний асортимент доступних для аренди лімузинів, надано короткий опис товару та його ціна з можливістю додати до кошику.

На сторінці «Yachts» надано повний асортимент доступних для аренди яхт, надано короткий опис товару, його ціна з можливістю додати до кошику

На сторінці «Girls» надана інформація про спектр додаткових послуг та коротка інформація про виконавців

На сторінці «Services» надано повний асортимент послуг які надає власник сайту та контактна інформація для більш детальної інформації

На сторінці «Vip transportation» надано асортимент автомобілів для приватних перевезень з описом ціною та можливістю додати до кошику

На сторінці «Private show» надано список танцювально-розважальних шоу та опис додаткових атрибутів в стилі хореографічної постанови.

На кожній сторінці знаходиться меню для комфортної навігації по сайту.

Кошик знаходиться у сайд-барі для зручного перегляду доданого вмісту кошику.

Також на кожній сторінці присутні посилання у вигляді іконок на соціальні мережі фірми.

3.2 Дизайн веб-сайту

Відповідно до заданої структури був спроектован зовнішній вигляд веб-сайту.

Для проектування зовнішнього вигляду використовувався графічний редактор photoshop.

Відповідно до заданої структури був спроектован зовнішній вигляд веб-сайту рис.3.1, рис3.2. Зовнішній вигляд шаблону сайту було спроектовано згідно новітніх стандартів веб-дизайну, з використанням сучасних трендів та побажань замовника.

Для проектування зовнішнього вигляду використовувався графічний редактор Photoshop. Однак photoshop не підтримується у використаній ОС тому під час розробки верстик використовувався графічний редактор GIMP.

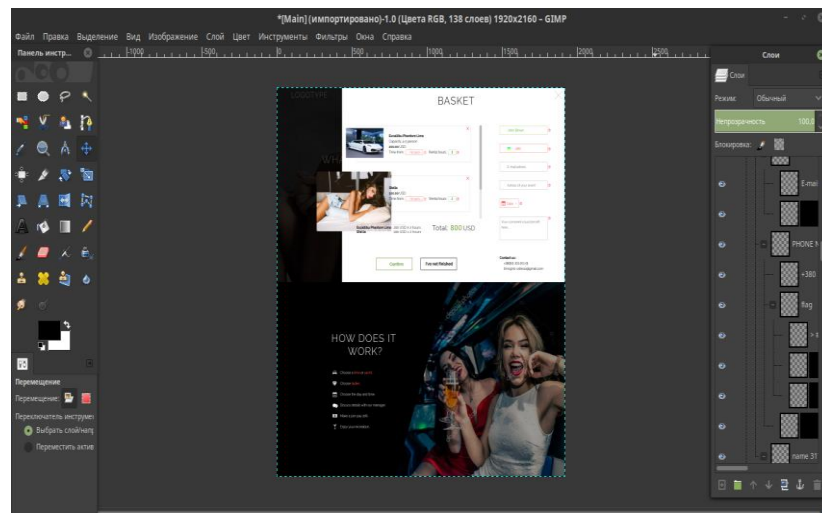


Рисунок 3.1 – Головна сторінка сайту

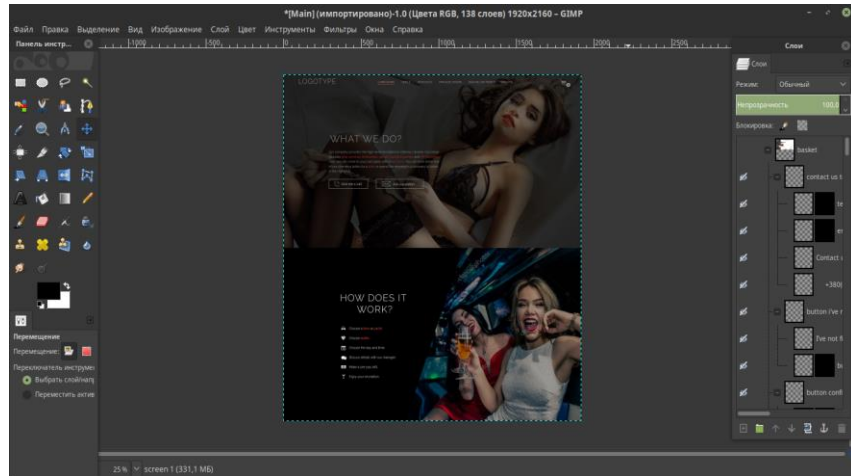


Рисунок 3.2 – Головна сторінка сайту з відкритим кошиком

Також було спроектовано шаблон для сторінок з товарами рис.3.3.

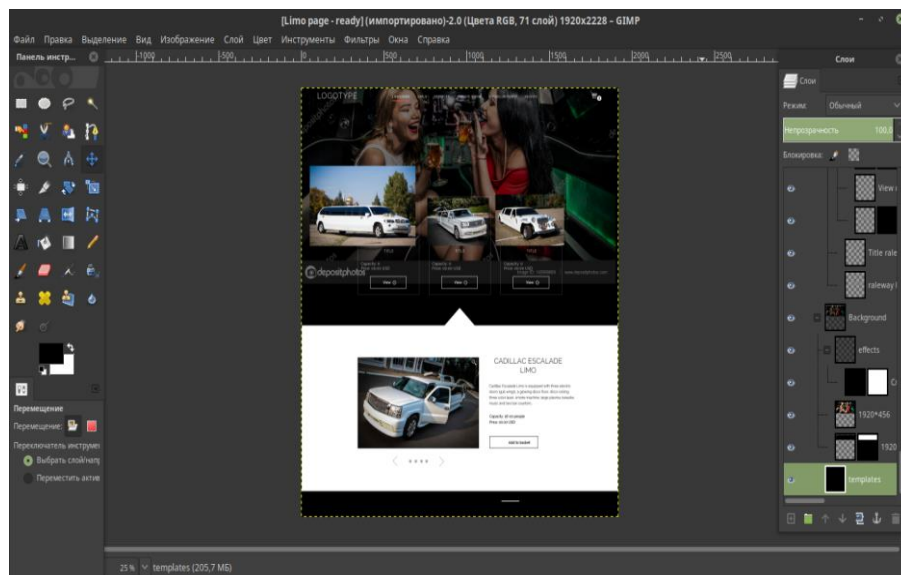


Рисунок 3.3 – Шаблон сторінки з товарами

Також за таким само принципом було створено макети для інших сторінок.

3.3 Створення HTML-шаблону та стилі-CSS

Згідно зі створеними макетами було розроблено HTML-шаблони. HTML-структура була розроблена згідно з стандартами W3. Отже, спочатку була розроблена структура меню, меню являє собою елемент навігації по сайту тож воно повинно знаходитися в середині тегу `<nav>`.

Створено 2 блоки, один для відображення меню на моніторах ноутбуків на персональних комп'ютерів інший для відображення на телефонах. Вдалося це зробити за допомогою медіа запитів CSS. В CSS3 включена підтримка різних технічних параметрів пристроїв, на основі яких потрібно завантажувати ті чи інші стилі. Наприклад, можна визначити смартфон з максимальною роздільною здатністю 640 пікселів і для нього встановити одні стилі властивості, а для інших пристроїв інші. Також можна виявити різні характеристики на кшталт наявності монохромного екрану, орієнтації (портретна або альбомна) і ін. Всі характеристики легко комбінуються, тому допустимо поставити стиль тільки для пристроїв альбомного формату з заданим дозволом екрану.

Можливості медіа-запитів не обмежуються виявленням мобільних пристроїв, з їх допомогою можна створювати адаптивний макет. Такий макет підлаштовується під дозвіл монітора і вікна браузера, змінюючи при необхідності ширину макета, число колонок, розміри зображень і тексту. Медіа-запити обмежують ширину макета і при досягненні цього значення (наприклад за рахунок зменшення вікна або при перегляді на пристрої з вказаним розміром) вже застосовується інший стиль.

Приклад медіа-запитів використаних в проекті:

```
@media screen and (max-width:1580px) {
}
@media (min-width: 768px) {
}
@media(max-width: 565px){
}
```

Потім було створено створена структура головної сторінки. На ній знаходиться 2 секції з інформацією, згідно з цим кожна з них знаходиться в середині тегу <section>. Наступний етап створення діалогових вікон в яких користувачі можуть залишити контактні данні для зворотного зв'язку рис. 3.4. Реалізовані за допомогою методів використаного фреймворку.

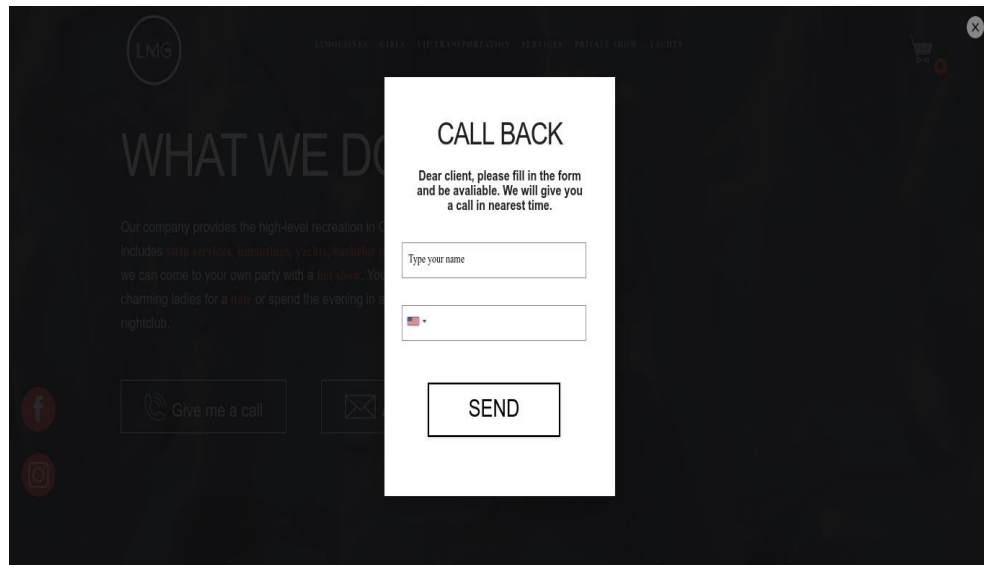


Рисунок 3.4 – Форма зворотного зв'язку

Вспливання діалогових вікон було зроблено за допомогою зміни стилів DOM-об'єктів. Під час кліку на кнопку, до діалогового вікна динамічно додається клас з необхідними стилями.

Далі було створено шаблон сторінки з лімузинами аналогічним чином. Відображення лімузинів було вирішено зробити у вигляді каруселі. Карусель була створена засобами фреймворку slides рис.3.5.

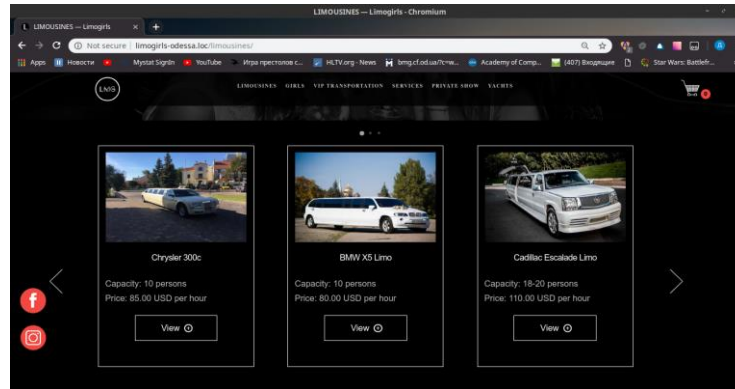


Рисунок 3.5 – Карусель з лімузинами

Також було вирішено не робити для кожного лімузину окрему сторінку для кожного лімузину, а зробити виїжджаючий блок з інформацією про лімузин рис.3.6.

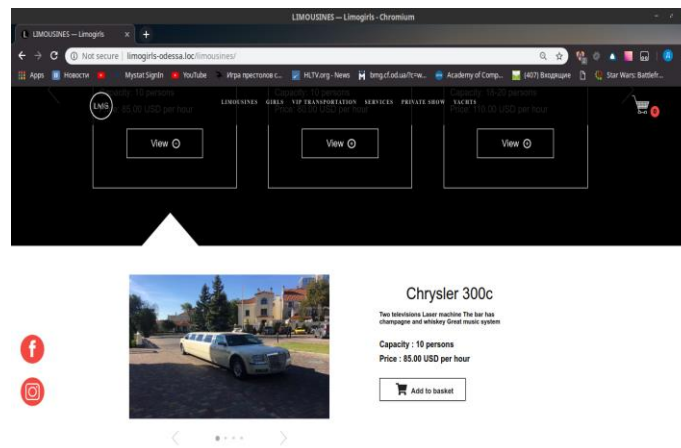


Рисунок 3.6 – Блок з інформацією про лімузин

Наступний крок створення сайдбару з кошиком рис.3.7.

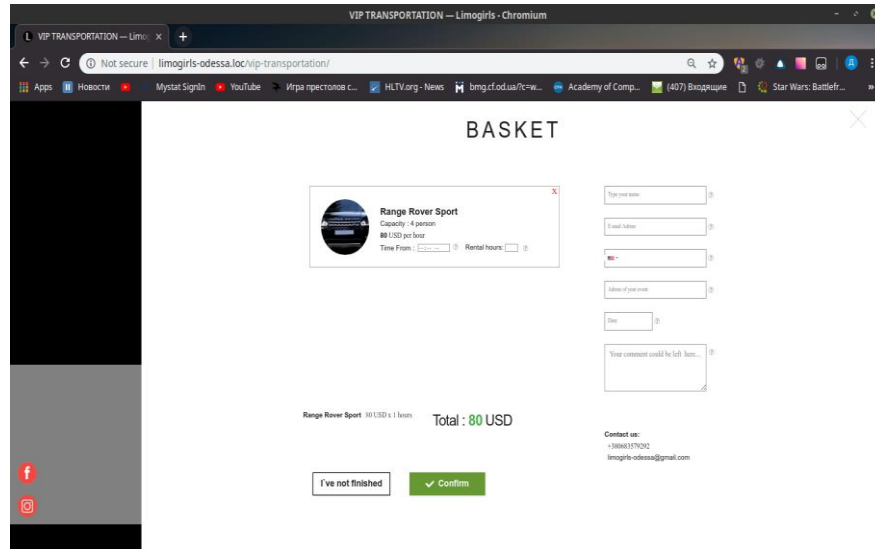


Рисунок 3.7 – Сайдбар з кошиком

Було створено звичайний html блок да за допомогою стилів зпозиційований так щоб він знаходився за межами екрану. При кліку на кнопку викликається функція яка позиціює його в межах екрану. Далі були створені сторінки Services та Prive Show зі статичним html кодом.

3.4 Створення javascript скриптів для динамічної зміни контенту та підключення плагінів

Для зручного заповнення форми з введення номеру телефону було використано плагін intlTelInput.js рис.3.8. intlTelInput.js – Додаток JavaScript для введення та перевірки міжнародних телефонних номерів. Він додає прапорець до будь-якого входу, виявляє країну користувача, відображає відповідний заповнювач і надає методи форматування / перевірки.

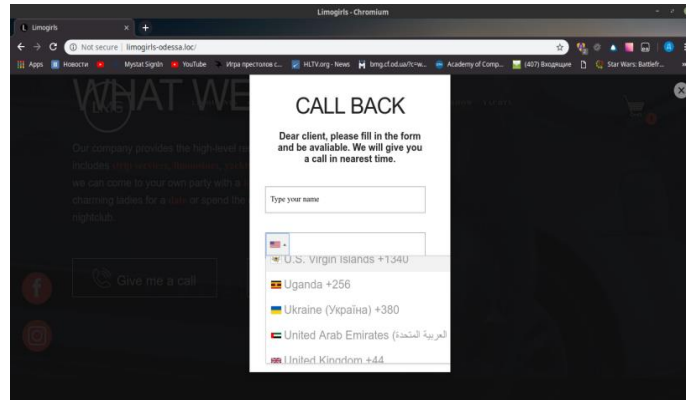


Рисунок 3.8 – Вибір телефонного коду країни

Наступний крок встановлення плагіну для вибору дати. Використовується плагін `jqueryDatetimerpicker` рис. 3.9.

Виклик діалогових вікон було зроблено за допомогою використовуваного фреймворку. До кнопок додавався клас `popupTrigger` та дата-атрибути `data-popup-id` в який записується унікальна назва діалогового вікна. Потім створюються `div` з класом `popup` та дата-атрибутом `data-popup-id` за назвою діалогового вікна.

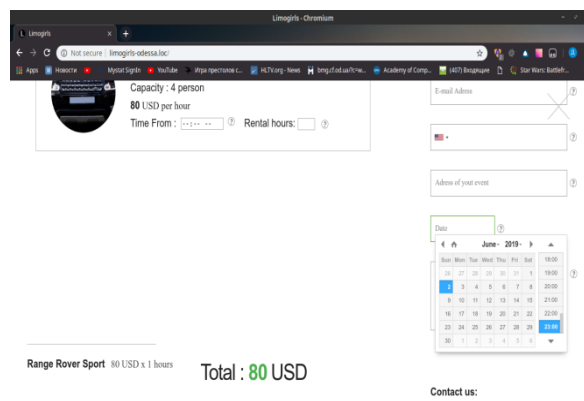


Рисунок 3.9 – Вибір дати

Карусель також була створена засобами фреймворку. Для виїжджаючого блоку була реалізована функція, яка зчитує дата-атрибути з кнопки, на яку клікають, та записує їх у відповідні dom-елементи.

3.5 Встановлення WordPress

Наступним кроком є встановлення CMS WordPress. Встановлення WordPress є досить простим :

- завантажити дистрибутив WordPress з офіційного сайту та розпакувати його;
- створити базу даних;
- перейти на веб-сайт на заповнити необхідні поля слідуючи інструкціям.

Тепер потрібно створити власну тему. Для цього перейдемо до директорії в якій знаходяться теми /wp-content/wp-theme. В ній створемо директорію з назвою теми. В ній необхідно створити файл style.css в якому вказати короткий опис теми, назву та автора теми. Після цього тема з'явиться в адмін панелі WordPress.

3.6 Створення теми

Перший крок у створенні теми створення файлу style.css який містить назву теми після цього тема з'явиться в адмін панелі рис.3.10.



Рисунок 3.10 – Вибір теми в адмін панелі

Далі потрібно створити необхідні файли такі як `index.php`, `header.php`, `footer.php`

У `header.php` виноситься код який буде знаходитися в шапці сайту, у `footer.php` виноситься код який буде знаходитися у футері, обидва ці файли необхідно підключати до кожного файлу шаблону сторінки за допомогою функцій `wp_head` та `wp_footer` рис. 3.11.

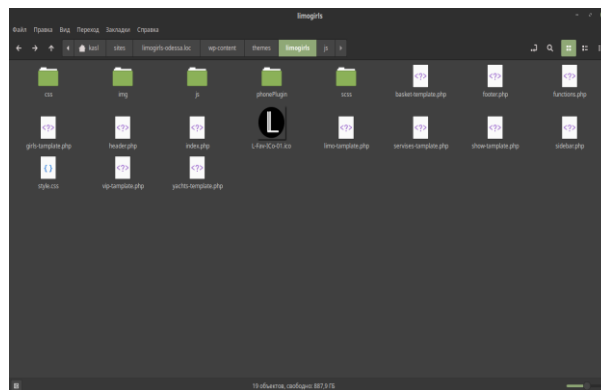


Рисунок 3.11 – Директорія з темою та необхідними файлами

Також потрібно створити файл `functions.php` за допомогою якого буде здійснюватися взаємодія теми та WordPress. Також були створені інші необхідні файли такі як `sidebar.php` та шаблони сторінок.

У `functions.php` потріб додати підтримку меню за допомогою функції `add_theme_support`, а у `header.php` вивести меню за допомогою функції `wp_nav_menu`.

Далі потрібно встановити необхідні плагіни. В цьому проєкті використовувалися плагіни `AdvancedCustomField`, `ContactForm7` та `SimpleBasket`.

Наступний крок створення структури сайту у адмін-панелі WordPress рис.3.12.

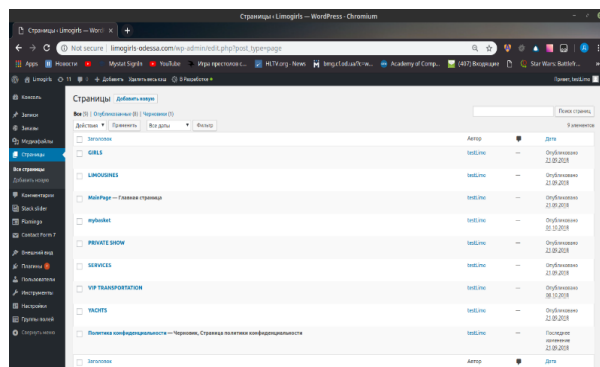


Рисунок 3.12 – Створені сторінки

Далі необхідно створити меню рис.3.13.

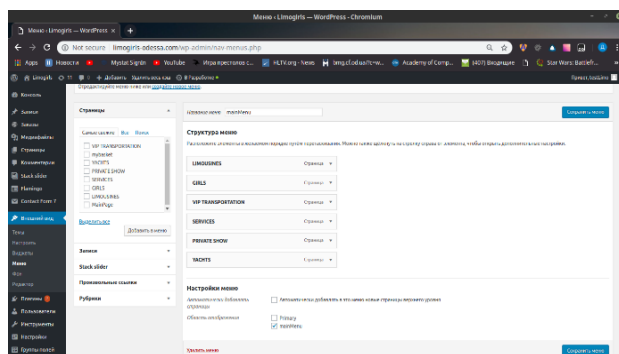


Рисунок 3.13 – Меню сайту

Наступним кроком є створення форм зворотного зв'язку. Зроблено це за допомогою плагіну ContacForm7 рис.3.14. Створенні форми необхідно вивести в файлах теми за допомогою функції `do_shortcode()`.

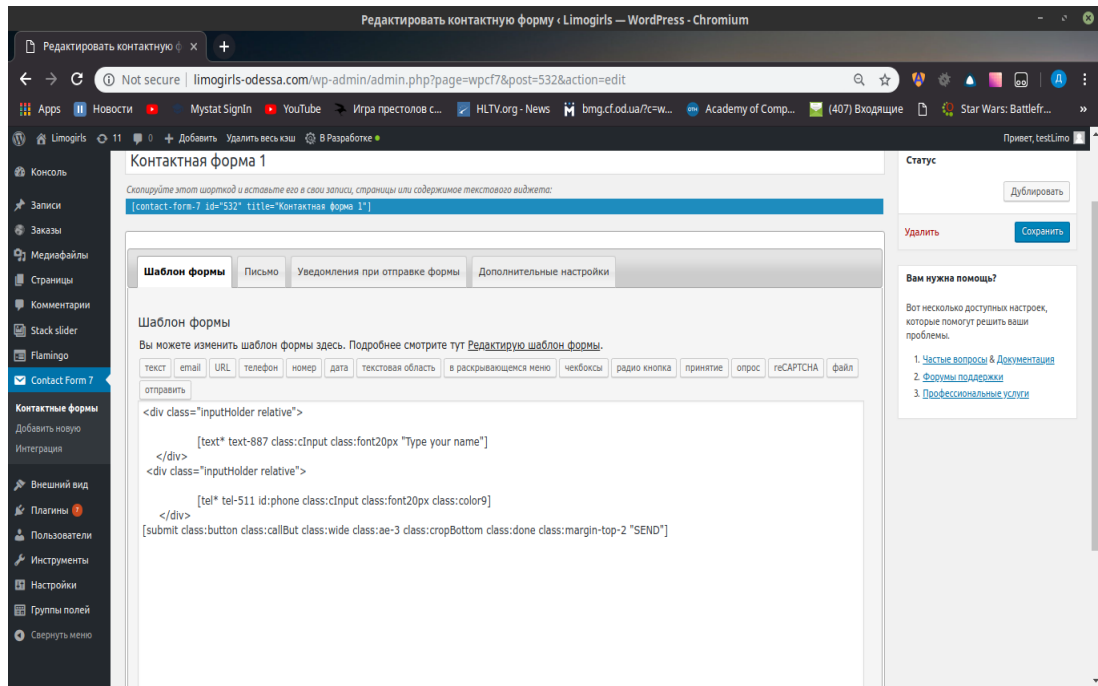


Рисунок 3.14 – Створення форми

Після створення форми необхідно скопіювати створений шорт-код та вивести його в шаблоні за допомогою функції `do_shortcode`.

Потім активувати плагін `simplebasket` скопіювати шорт-код корзини та вставити його до заздалегіть створеної сторінки корзини.

Недоліком плагіну є можливість видаляти товар з корзини та те що для оновлення корзини необхідно оновлювати сторінку що є досить незручним оскільки в моєму випадку корзина знаходиться у сайдбарі.

Тож для плагіну була дописана функція для видалення файлів з корзини та створений `javascript` з використанням технології `ajax` для динамічно оновлення інформації на сторінці. Також було змінено структуру виводу корзини згідно з макетом сайту.

Функція для видалення була дописана до класу SimpleBasketOrder, вона видаляла з існуючої сесії товар по id який передаються їй в якості атрибуту.

Наступний крок активація плагіну AdvencedCustomFields рис. 3.15, створення необхідних полів та додання постів рис.3.16.

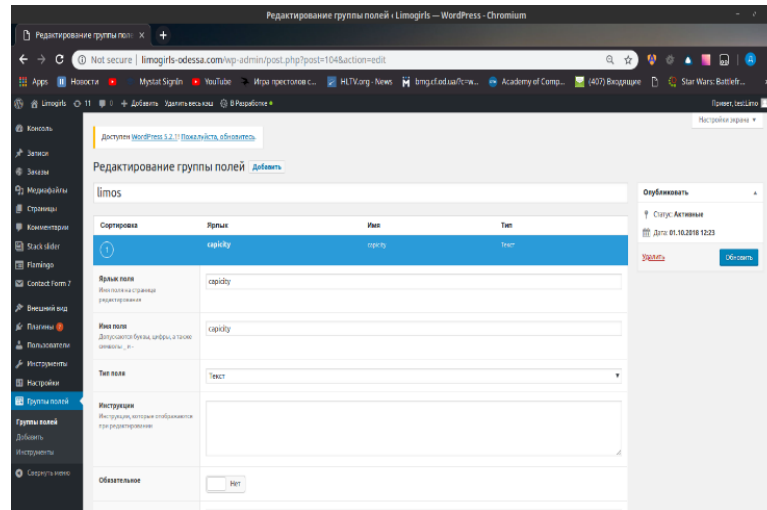


Рисунок 3.15 – Створення полів

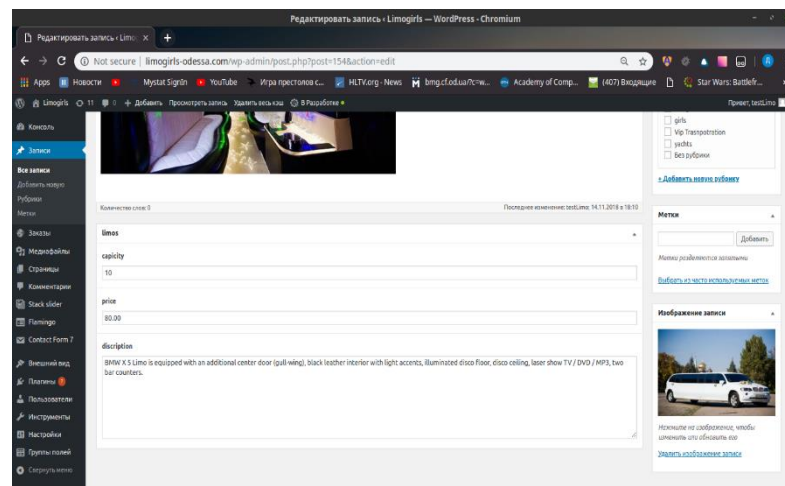


Рисунок 3.16 – Створення записів

Після того як записи створенні їх необхідно вивести до відповідного шаблону зроблено це за допомогою функції `wp_query` яка повертає записи з

бази даних за заданим запитом. Далі в циклі виводимо пости за заданій в запиті категорії інформацію в html видимо за допомогою стандартних функцій WordPress нахшталт `the_title`, `the_content`, інформацію з кастомних полів за допомогою `the_field` в яку передається ім'я необхідного поля.

3.7 Створення валідації форми

Останнім кроком є створення валідацій форми для того щоб користувачі вводили коректні данні рис. 3.17. Перевірки полів використовувалися регулярні вирази.

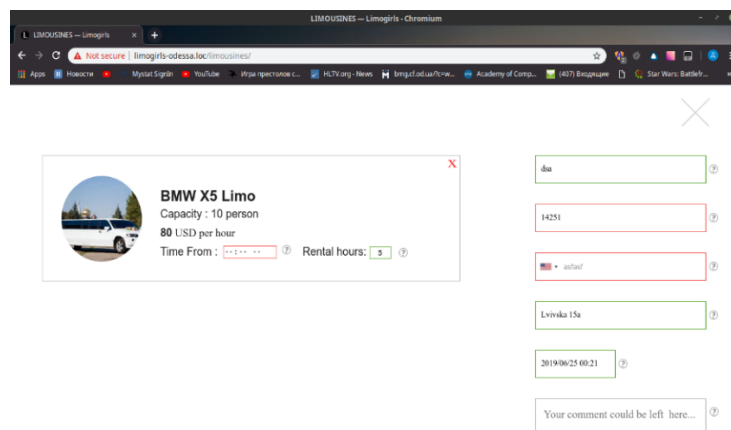


Рисунок 3.17 – Вигляд валідації форми

В полі ім'я не повинно бути цифр, поле з телефоном повинно містити цифри та символи які підходять під маску, поле з імейлом також пивнно підходити під маску:

- ім'я = `/^[^\d]+$ /;`
- імейл = `/^[a-z0-9_\.-]+@[a-z0-9_\.-]+\.[a-z\.-]{2,6}$/;`
- телефон = `/^[\s*]?(\+)?([- _():=+]?\d[- _():=+]?)\{10,14\}(\s*)?$/;`

ВИСНОВКИ

Система управління контентом WordPress, є дуже гнучкою. Практика показує, що цю платформу можна застосовувати для вирішення більшості завдань, пов'язаних зі створенням веб-ресурсів різного призначення, хоча спочатку ця CMS створювалася як система управління блогом.

Гнучкість WordPress і її безкоштовність роблять її такою популярною CMS серед більшості веб-розробників і простих користувачів. Внесок в популярність даної системи вносить також інтуїтивно зрозумілий інтерфейс, локалізації для більшості мов, розширюваність функціоналу за допомогою додаткових плагінів, а також можливість зміни дизайну за допомогою тем.

Створення сайту на базі вільної CMS WordPress процес не складний і не вимагає багато досвіду і часу. Але в той же час, дана CMS при наявності відповідних навичок дозволяє розробнику пристосувати цю систему управління контентом фактично для будь-яких потреб. Модулі і теми ще більш розширюють сферу застосування даної CMS. WordPress – є оптимальним рішенням для більшості проектів в сфері веб-розробки. У даної системи є істотний недолік – це її швидкодія. При збільшенні навантаження на сайт, а також при встановленні додаткових плагінів, швидкість роботи даної CMS істотно падає.

У наступних релізах даної CMS, розробникам WordPress бажано оптимізувати систему з метою збільшення швидкості її роботи. Також з урахуванням того, що дану платформу все частіше використовують не тільки для створення блогів, бажано збільшити її функціональність і відійти в сторону від ролі системи управління блогами.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Бабаев А., Евдокимов Н., Боде М. Создание сайтов. Санкт-Петербург, 2014. 410 с.
2. Вандюк Дж. CMSWordpress. Руководство по разработке системы управления сайтом., М.: Вильямс, 2008. 400 с.
3. Веллинг Л., Томсон Л. Разработка Web-приложений на PHP и MySQL. М.: Вильямс, 2005. 876 с.
4. Википедия. Веб разработка. URL: <https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0> (дата звернення: 30.03.2019).
5. Вікіпедія. Веб-сайт.
URL: <https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82> (дата звернення: 30.03.2019).
6. Даруй К., Бринзаре Б., Черchez-Тоза Ф., Бусика М. Ајах и РНР. Разработка динамических веб-приложений. М.: Символ, 2007. 332 с.
7. Дронов В. А. РНР, MySQL. Разработка интерактивных Web-сайтов. СПб.: БХВ-Петербург, 2005. 448 с.
8. Каслдайн Э., Шарки К. Изучаем JQUERY. Новый стиль программирования на JavaScript., СПб.: Питер, 2011. 366 с.
9. Колисниченко Д. Выбираем лучший бесплатный движок для сайту. CMS Wordpres и Drupal (+ CD-ROM). М.: БХВ-Петербург, 2010. 184 с.
10. Колисниченко Д. Движок для вашего сайту. CMS Wordpres. Slaed, РНР-Nuke., СПб.: БХВ-Петербург, 2008. 368 с.
11. Крейн Д., Паскарелло Э. Ајах в действии., М.: Вильямс, 2006. 638 с.
12. Купер А., Рейманн Р. М., Кронин Д. Алан Купер об интерфейсе. Основы проектирования взаимодействия., СПб.: Символ-Плюс, 2017. 688 с.

- 13.Мейер Э. А. CSS - каскадные таблицы стилей. Подробное руководство., М.: Символ, 2006. 572 с.
- 14.Панфилов К. По ту сторону веб-страницы. М.: ДМК Пресс, 2008. 440 с.
- 15.Сабин-Уильсон Л. WordPress для чайников. М.: Вильямс, 2010. 368 с.
- 16.Тидвелл Дж. Разработка пользовательских интерфейсов., СПб.: Питер, 2011. 480 с.
- 17.Уильямс Б., Дэмстра Д., Стэрн Х.Wordpress для профессионалов. Разработка и дизайн сайтов., СПб.: Питер, 2014. 464 с.
- 18.Bartlett D. WordPress in easy steps. 2nd ed., In Easy Steps Limited, 2015. 192 p.
- 19.Blakeley T. Silver WordPress Theme Design. PACT Publishing, 2008. 224 p.
- 20.Coyier C., Starr J. Digging Into WordPress v3.: WordPress, 2012. 442 p.
- 21.Hussey T. Using WordPress. Que, 2012. 425p.
- 22.Newcomer E. Understanding Web Services: XML, WSDL, SOAP, and UDDI. Addison-Wesley Professional, 2002. 360 p.
- 23.Nixon R. Learning PHP, MySQL, and Javascript (Animal Guide). O'Reilly Media, 2009. 528 p.
- 24.Percentages of websites using various server-side programing language.URL: <https://i2.wp.com/www.pixelcrayons.com/blog/wp-content/uploads/2018/01/Usage-Statistics-and-Market-Share-of-Server-side-Programing-Languages-for-Websites-January-2018.png?w=438&ssl=1> (дата звернения: 30.03.2019).
- 25.PHP vs Python vs Ruby. Most discussed language. URL: <https://i0.wp.com/www.pixelcrayons.com/blog/wp-content/uploads/2017/02/php-vs-python-vs-ruby-most-discussed-language.png?ssl=1> (дата звернения: 30.03.2019).
- 26.PHP vs Python vs Ruby. Popularity. URL: <https://i0.wp.com/www.pixelcrayons.com/blog/wp-content/uploads/2017/02/php-vs-python-vs-ruby-popularity.png?ssl=1> (дата звернения 30.03.2019).

27.PHP vs Python vs Ruby. Usability. URL: <https://i0.wp.com/www.pixelcrayons.com/blog/wp-content/uploads/2017/02/php-vs-python-vs-ruby-usability-performance.png?ssl=1> (дата звернення: 30.03.2019).