МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет радіоелектроніки

Факультет Центр післядипломної освіти
(повна назва)

Кафедра Програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження ефективності розробки web-сайтів без використання фреймворків CSS
(тема)

Виконав:
Студент 2 курсу, групи ІПЗздм-19-1

Шестаков Д. Ю.
(прізвище, ініціали)

Спеціальність 121-Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Керівник проф. каф. ПІ, д.ф.-м.н., проф. Смеляков С. В.
(посада, прізвище)

Допускається до захисту

Зав. кафедри _____ З. В. Дудар
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Центр післядипломної освіти
(повна назва)

Кафедра Програмної інженерії
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 121-Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:
Зав.кафедри _____
(підпис)
« 26 » 03 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Шестакова Дмитра Юрійовича
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження ефективності розробки web-сайтів без використання фреймворків CSS

затверджена наказом університету від 26.03.2021 № 34Стз
2. Термін подання роботи до екзаменаційної комісії 08.05 2021р.
3. Вихідні дані до роботи пояснювальна записка
4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз проблемної галузі і постановка задачі, розробка веб-сайтів без використання фреймворків
5. Перелік графічного матеріалу із зазначенням креслеників, схем, слайдів, ілюстрацій
слайди: титульний лист, актуальність роботи та мета, характеристика поняття

веб-сайту, типи та класифікація веб-сайтів, основні принципи побудови веб-сайту, дослідження інструментів розробки веб-сайтів, загальний аналіз найпопулярніших фреймворків, переваги використання css-фреймворків при розробці веб-сайту, доцільність веб-розробки без використання css-фреймворків, порівняння процесу розробки веб-сайту з використанням css-фреймворків та без них, висновок

6. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
спецчастина	проф. каф. ПІ, д.ф.-м.н., проф. Смеляков С. В.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	25 січня 2021р.	виконано
2	Огляд існуючих методів та інструментів	05 березня 2021р.	виконано
3	Реалізація програмного продукту	01 квітня 2021р.	виконано
4	Підготовка пояснювальної записки	28 квітня 2021р.	виконано
5	Спецчастина	30 квітня 2021р.	виконано
6	Нормоконтроль	05 травня 2021р.	виконано
7	Рецензування	06 травня 2021р.	виконано
8	Підготовка презентації та доповіді	07 травня 2021р.	виконано
9	Попередній захист	08 травня 2021р.	виконано
10	Занесення роботи в електронний архів	09 травня 2021р.	виконано
11	Допуск до захисту у зав. кафедри	10 травня 2021р.	виконано

Дата видачі завдання 25 січня 2021р.

Студент Шестаков Дмитро Юрійович
(підпис)

Керівник роботи _____ проф. каф. ПІ, д.ф.-м.н., проф. Смеляков С. В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 74 с., 30 джер.

CSS-FRAMEWORKS, WEB-САЙТ, CSS-LYBRARY, PHP, HTML, НАУКОВО-ДОСЛІДНИЦЬКА ПРАКТИКА, БАЗА ДАНИХ, WEB-ДИЗАЙН, FRONTEND, MYSQL, ЕФЕКТИВНІСТЬ РОЗРОБКИ.

Об'єкт розробки - Дослідження ефективності розробки web-сайтів без використання фреймворків CSS.

Мета розробки - знизити витрати часу на розробку web-сайтів.

Метод рішення - мова розмітки HTML, каскадні таблиці стилей CSS, мова javascript, Sublime Text, Framework Bootstrap.

В результаті розробки буде створена система залежності розробки web-сайтів від фреймворків, що допоможе зрозуміти актуальність їх використання.

CSS-FRAMEWORKS, WEBSITE, CSS-LYBRARY, PHP, HTML, RESEARCH PRACTICE, DATABASE, WEB DESIGN, FRONTEND, MYSQL, DEVELOPMENT EFFICIENCY.

Object of development - Research of efficiency of development of web-sites without use of CSS frameworks.

The purpose of development is to reduce the time spent on developing web-sites.

Solution method - HTML markup language, cascading style sheets CSS, javascript language, Sublime Text, Framework Bootstrap.

As a result of development the system of dependence of development of web-sites on frameworks that will help to understand urgency of their use will be created.

Я, Шестаков Дмитро Юрійович, студент гр. ІПЗздм-19-1, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження ефективності розробки web-сайтів без використання фреймворків CSS», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіатуі вона може бути опублікована в електронному архіві відкритого доступу ElAr KhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ..... 7
1 Основні поняття та принципи розробки веб-сайту..... 9
1.1 Характеристика поняття веб-сайту..... 9
1.2 Типи та класифікація веб-сайтів..... 12
1.3 Основні принципи побудови веб-сайту..... 19
1.4 Дослідження інструментів розробки веб-сайтів..... 21
1.5 Постановка задачі..... 23
2 Розробка веб-сайту без використання CSS-фреймворків..... 24
2.1 Загальний аналіз найпопулярніших css-фреймворків..... 24
2.2 Переваги використання css-фреймворків при розробці веб-сайту..... 32
2.3 Доцільність веб-розробки без використання css-фреймворків..... 36
2.4 Порівняння процесу розробки веб-сайту з використанням css-фреймворків та без них..... 44
Висновок..... 58
Перелік джерел посилання..... 60
Додаток А Перелік джерел посилання за науковими напрямами керівника та науковців кафедри програмної інженерії..... 63
Додаток Б Звіт результатів перевірки кваліфікаційної роботи на унікальність тексту..... 64
Додаток В Слайди презентації..... 65
Додаток Г Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008:2015..... 73

ВСТУП

Актуальність роботи. Розробка веб-сайтів є перспективним напрямком у розвитку інформаційних технологій. Всесвітня мережа Інтернет охоплює з кожним роком все більше і більше користувачів. Так, в 2013 році було проведено дослідження компанією Genius-Україна, яким було встановлено, що розмір інтернет-аудиторії на кінець травня 2013 року склав більше 16 млн. чоловік. Зростання в порівнянні з травнем 2012 року - 11%. Щомісяця користувачі переглядають 1000-1300 сайтів / веб-сторінок. Це показує, що зі збільшенням інтернет-аудиторії буде збільшуватися кількість корпоративних і інформаційних ресурсів [1].

В останні роки спостерігається бурхливий розвиток Інтернет-технологій. Можна спостерігати значні зміни в мережі Інтернет. Сайти, які раніше були платформою для розташування статичного контенту, тепер стали багатофункціональними, інтерактивними системами для надання різної інформації. Розвиток Інтернету нерозривно пов'язано з проектуванням сайтів.

Масова поява сайтів спровокувало проблему їх якості. Популярність створення веб-ресурсів сприяла розробці різних систем і програм, які спрощують процес написання сайту. До них відносяться фреймворки - структура програмної системи, а також програмне забезпечення, що спрощує розробку і об'єднання різних компонентів великого проекту програмування.

Аналіз досліджень і публікацій. Загальний аналіз показав, що стрімкий розвиток інформаційних технологій дало можливість вибору різних технологій і методів розробки web-ресурсів. Правильність вибору технології на першому етапі проектування гарантує 90% успіху в просуванні і роботі ресурсу в мережі Інтернет. Було виявлено, що фреймворки спрощують процес розробки та підтримки технічно навантажених сайтів. І так як дана тема слабо розкрита, тому доцільно провести дослідження, що розкриває можливості фреймворків.

Мета роботи полягає у дослідження можливостей фреймворків CSS, а також доцільності розробки веб-сайтів без їхнього використання.

1 ОСНОВНІ ПОНЯТТЯ ТА ПРИНЦИПИ РОЗРОБКИ ВЕБ-САЙТУ

1.1 Характеристика поняття веб-сайту

Розробка веб-сайту - це процес створення веб-сайтів. Він охоплює декілька різних аспектів, включаючи макет веб-сторінки, виготовлення вмісту та графічний дизайн. Хоча терміни веб-дизайн та веб-розробка часто використовуються як взаємозамінні, веб-дизайн технічно є підгрупою ширшої категорії веб-розробки.

Веб-сайти створюються з використанням мови розмітки під назвою HTML. Веб-дизайнери створюють веб-сторінки за допомогою тегів HTML, які визначають вміст і метадані кожної сторінки. Макет та зовнішній вигляд елементів на веб-сторінці зазвичай визначаються за допомогою CSS або каскадних таблиць стилів. Тому більшість веб-сайтів включають комбінацію HTML і CSS, яка визначає, як кожна сторінка буде виглядати в браузері.

Поки веб-сайт розробляється, слід враховувати, яку мету можна досягти при розробці веб-сайту. Яка категорія осіб повинна бути потенційними клієнтами товару, який пропонується продати на веб-сайті, яка їх вікова група тощо? Веб-сайт необхідний для задоволення потреб клієнтів, а наступна дія, яку очікують від клієнтів, - рекомендувати веб-сайт своїм друзям. Справа не повинна закінчуватися одним відвідуванням клієнта, має бути достатньо залучення для повторних відвідувань клієнтів[2].

Після прийняття концепції наступним є пошук послуги хостингу. Зараз день отримання хорошого господаря дуже економічний. Існують безкоштовні пропозиції хостингу веб-сайтів, але недоліком є те, що для цих сайтів потрібен банер хоста на веб-сайті, що негативно впливає на клієнтів. Тож вибір завіреного хостингу рекомендується для задовільного обслуговування.

Веб-сайти представлені майже безмежним різноманіттям, включаючи освітні сайти, сайти новин, порносайти, форуми, сайти соціальних медіа, сайти електронної комерції тощо. Сторінки на веб-сайті, як правило, являють собою

поєднання тексту та інших засобів масової інформації. Тим не менш, немає правил, які б диктували форму веб-сайту.

Людина могла створити веб-сайт із нічого, крім чорно-білих фотографій троянд або слова "кішка", пов'язаного з іншою веб-сторінкою словом "миша". Однак багато веб-сайтів дотримуються стандартного шаблону домашньої сторінки, що посилається на інші категорії та вміст на веб-сайті.

Домашня сторінка (або просто “дім”) представляє головну сторінку самого сайту. Часто домашня сторінка є свого роду «хабом», з якого можна отримати доступ до всіх інших сторінок. Внутрішня веб-сторінка, на яку зв’язано кілька інших сторінок у цілісній структурі (наприклад, конкретна категорія тем), натомість називається „батьківською сторінкою“.

Кожна сторінка є єдиним документом HTML, і всі вони пов’язані за допомогою гіперпосилань (або просто “посилання”), які можна зручно поєднати на панелі навігації.

Панель навігації відображається на кожній сторінці, а не лише на домашній сторінці, і дозволяє користувачеві швидко переходити по структурі основного веб-сайту.

Іншим важливим розділом більшості веб-сайтів є нижній колонтитул, який є ще одним періодичним розділом, який знаходиться внизу кожної сторінки. Зазвичай нижній колонтитул містить зовнішні посилання, що вказують на подібні веб-сайти та інші зовнішні ресурси, а також іншу важливу інформацію, таку як відмова від відповідальності, посилання на умови надання послуг, політику конфіденційності та сторінки контактів, а також фізичну адресу компанії, що володіє сайтом[11].

Веб-сайти розміщуються на серверах і вимагають відвідування веб-браузера, такого як Chrome, Firefox або Internet Explorer (на комп’ютері чи мобільному пристрої).

До веб-сайту можна отримати прямий доступ, ввівши його URL-адресу або здійснивши пошук за допомогою такої пошукової системи, як Google або Bing.

Спочатку веб-сайти класифікувались за доменами верхнього рівня. Деякі приклади включають:

- веб-сайти державних установ = .gov;

- веб-сайти навчальних закладів = .edu;

- веб-сайти некомерційних організацій = .org;

- комерційні веб-сайти = .com;

- інформаційні сайти = .info.

Хоча ці розширення доменів верхнього рівня все ще існують, вони мало говорять про фактичний вміст веб-сайту. У сучасному Інтернеті розширення ".com" на сьогоднішній день є найпопулярнішим доменом разом із багатьма іншими розширеннями для окремих країн (.it, .de, .co.uk, .fr тощо)[7].

Перший веб-сайт був створений в 1990 році Тімом Бернерс-Лі, британським фізиком з ЦЕРНу. Через 3 роки, у 1993 році, ЦЕРН оголосив, що кожен може отримати доступ до Інтернету та користуватися ним безкоштовно.

Першим кроком у розробці веб-сайтів є визначення макета для першої сторінки. Корисно, якщо хтось знає HTML, інакше зазвичай зустрічається “Що ти бачиш, те і отримуєш”, тобто редактор (WYSIWYG) служить для цього. Використання HTML може зробити веб-сайт кращим. У мережі є прості підручники. Можна взяти їх допомогу, навчившись і застосувати те саме на веб-сайті, що розробляється.

Настав час провести перегляд сторінки, якщо буде помічена помилка. Подивіться, скільки часу займає сторінка для завантаження та навігації на інші сторінки, це добре? Чи достатньо приваблива тема для очей? Як шрифти? А як щодо орфографії та граматики вмісту, що використовується на сторінці? Невеликі

фактори говорять про людину, для якої створений веб-сайт. Повторне використання та догляд за цими пунктами є обов'язковим, замість того, щоб дозволити мотлоху залишатися, щоб погіршити ваш імідж власника веб-сайту.

1.2 Типи та класифікація веб-сайтів

Класифікація відіграє життєво важливу роль у багатьох завданнях з управління інформацією та пошуку. В Інтернеті класифікація вмісту сторінки має важливе значення для цілеспрямованого сканування, сприяння розробці веб-каталогів, аналізу тематичних посилань на веб-сторінки, контекстної реклами та аналізу актуальної структури Інтернету.

Класифікація веб-сторінок також може допомогти поліпшити якість веб-пошуку. У цьому опитуванні ми вивчаємо простір підходів веб-класифікації, щоб знайти нові напрямки для досліджень, а також зібрати найновіші практики для інформування про майбутні впровадження класифікаторів. Опитування в класифікації веб-сторінок, як правило, не мають детального обговорення використання специфічних для Інтернету функцій. У цьому опитуванні ми ретельно розглядаємо специфічні для Інтернету функції та алгоритми, які були вивчені та виявились корисними для класифікації веб-сторінок. Вклади цього опитування - детальний огляд корисних для класифікації веб-функцій; —Перелік основних додатків для веб-класифікації; та —обговорення майбутніх напрямків досліджень[5].

Класифікація веб-сторінок, також відома як категоризація веб-сторінок, - це процес присвоєння веб-сторінки одній або кільком попередньо визначеним міткам категорій. Класифікація традиційно ставиться як контрольована проблема навчання [4], в якій набір маркованих даних використовується для підготовки класифікатора, який може застосовуватися для позначення майбутніх прикладів. Загальну проблему класифікації веб-сторінок можна розділити на більш конкретні проблеми: класифікація предметів, функціональна класифікація, класифікація настроїв та інші типи класифікації.

Класифікація предметів стосується теми або теми веб-сторінки. Наприклад, судження про те, чи йдеться на сторінці про “мистецтво”, “бізнес” чи “спорт”, є прикладом класифікації предметів. Функціональна класифікація турбується про роль, яку відіграє веб-сторінка. Наприклад, рішення про те, що сторінку буде “персональною домашньою сторінкою”, “сторінкою курсу” чи “сторінкою прийому”, є прикладом функціональної класифікації. Класифікація настроїв фокусується на думці, яка представлена на веб-сторінці, тобто на ставленні автора до певної теми.

Інші типи класифікації включають класифікацію за жанрами [8], класифікацію спаму в пошукових системах [12]. Це опитування зосереджується на предметній та функціональній класифікації. Залежно від кількості класів у задачі класифікацію можна розділити на двійкову класифікацію та багатокласову класифікацію, де двійкова класифікація класифікує екземпляри точно на один із двох класів, а багатокласова класифікація має справу з більш ніж двома класи.

Залежно від кількості класів, які можна присвоїти екземпляру, класифікацію можна розділити на класифікацію за однією міткою та класифікацію з багатьма мітками. У класифікації з однією міткою кожному екземпляру слід призначати одну і тільки одну мітку класу, тоді як у класифікації з багатьма мітками екземпляру може бути призначено більше одного класу. Якщо проблема багатокласова, наприклад, класифікація чотирьох класів, це означає, що задіяні чотири класи, наприклад, Мистецтво, Бізнес, Комп’ютери та Спорт. Це може бути або одна мітка, де екземпляру може бути присвоєна рівно одна мітка класу, або багатомітка, де екземпляр може належати будь-якому одному, двом або всім класам[23].

Залежно від типу присвоєння класу класифікацію можна поділити на тверду та м’яку. У жорсткій класифікації екземпляр може бути або не бути в певному класі без проміжного стану, тоді як при м'якій класифікації екземпляр може бути передбачений в якомусь класі з певною ймовірністю (часто розподіл ймовірності між усіма класами, як на малюнку.

Залежно від організації категорій, класифікацію веб-сторінок також можна розділити на класифіковану класифікацію та ієрархічну класифікацію. У вузькій класифікації категорії вважаються паралельними, тобто одна категорія не замінює іншу, тоді як в ієрархічній класифікації категорії організовані в ієрархічну деревоподібну структуру, в якій кожна категорія може мати ряд підкатегорій.

Досить часто ми отримуємо дзвінки, де нам кажуть: "Мені потрібно створити веб-сайт" - наше перше запитання - це, який саме веб-сайт?

Далі, я маю на меті обговорити типи доступних веб-сайтів та те, для чого вони підходять.

Як правило, веб-сайт підпадає під одну з наступних широких категорій:

- нередагований веб-сайт брошури (його часто називають статичним сайтом);

- веб-сайт для брошур, який можна редагувати (вимагає системи управління вмістом);

- динамічний веб-сайт, який можна редагувати, з більшим залученням користувачів (області входу тощо) та самостійним управлінням через систему управління вмістом;

- сайт електронної комерції - інтегрований із платіжним шлюзом, таким як Paypal, Stripe або банками, такими як Barclays або HSBC;

- прогресивна веб-програма - коли веб-сайт вимагає безлічі спеціальних функцій або процесів, він, як правило, підпадає під категорію веб-додатків[10].

Рішення про те, який тип веб-сайту вам потрібен, залежить від того, що ви насправді хочете, щоб веб-сайт робив. Якщо ви просто хочете веб-сайт, оскільки ви вважаєте, що він повинен бути у вас, простий брошурний веб-сайт - це все, що потрібно. В ідеальному світі жоден веб-сайт ніколи не повинен бути повністю статичним і не змінюватися, але ви повинні бути реалістом, чого можна досягти за допомогою ваших ресурсів.

Хоча очевидно, що у світі існує не лише дванадцять різних типів веб-сайтів, ми підібрали найпоширеніші категорії, щоб дати загальне уявлення.

Найпоширеніші типи веб-сайтів:

а) блог;

б) корпоративний;

в) електронна комерція;

г) портфоліо;

д) брошура;

е) краудфандінг;

ж) новини чи журнал;

з) соціальні мережі;

і) потокове телебачення або відео;

к) виховна;

л) портал;

м) вікі або форум спільноти.

Нижче ви знайдете більше інформації та наш рекомендований конструктор веб-сайтів для проектування кожного типу[23].

а) блог

Ви, мабуть, стикалися з блогами під час перегляду веб-сторінок, але для тих, хто не знайомий, це онлайн-журнали чи інформаційні сторінки, які регулярно оновлюються.

Як правило, блог може керувати окремою особою чи невеликою групою будь-яку тему - будь то поради щодо подорожей, фінансові поради чи відгуки на пампушки. Хоча вони часто пишуться в неформальному або розмовному стилі, професійні блоги стали надзвичайно популярним методом заробітку в Інтернеті.

Хочете створити власний щоденник, але не знаєте з чого почати? Ми рекомендуємо спробувати Wix - це найкращий конструктор веб-сайтів на сьогоднішній день, який дозволяє вам створити власний блог за пару годин, і це безкоштовно. Wix також пропонує деякі чудові інструменти для ведення блогу, такі як аналіз продуктивності, функція коментарів та соціальні закладки.

б) корпоративний

Менш ніж дві третини (64%) малого бізнесу мають веб-сайт. Це напрочуд низький показник, враховуючи, наскільки важлива присутність в Інтернеті для довіри компанії. І на ваше щастя, це означає, що ви можете створити веб-сайт, щоб надати своєму бізнесу конкурентну перевагу.

Ви не можете продавати безпосередньо через корпоративний веб-сайт, але ви можете використовувати ці веб-сайти, щоб надати інформацію про свій бізнес та повідомити потенційних клієнтів або клієнтів, як вони можуть зв’язатися з вами.

Найкраще, це не коштує великих витрат на створення надійного корпоративного веб-сайту. За допомогою 1 & 1 IONOS ви можете безкоштовно створити для вас веб-сайт, який не вийшов з коробки (протягом перших трьох місяців, потім він коштує 9 фунтів на місяць). Ціни правильні станом на січень 2021 року.

в) електронна комерція

Сайт електронної комерції, який інакше називають Інтернет-магазином, дозволяє здійснювати онлайн-платежі за товари чи послуги. Магазини можуть функціонувати як самостійні веб-сайти, так і поєднуватися з блогами чи корпоративними веб-сайтами.

Наприклад, чисто корпоративний веб-сайт без функцій електронної комерції все ще може опосередковано спонукати користувачів щось купувати, але не може приймати будь-які платежі.

Якщо ви хочете побудувати магазин електронної комерції сьогодні, тоді ми рекомендуємо спробувати 14-денну безкоштовну пробну версію Shopify. Це не

тільки найпростіша у використанні платформа, але Shopify - це найповніша та найрізноманітніша платформа електронної комерції на ринку, що дозволяє продавати через кілька каналів, включаючи Facebook, Instagram, eBay та Amazon.

Ви можете спробувати Shopify безкоштовно, ціни починаючи з £ 23 / місяць після безкоштовного пробного періоду.

г) портфоліо

Як і фізичне портфоліо, такі типи веб-сайтів використовуються для відображення та просування прикладів попередньої роботи. Портфоліовий веб-сайт, який в першу чергу використовується тими, хто займається креативною індустрією, може використовуватися як резюме, демонструючи свої навички, щоб справити враження на клієнтів, клієнтів або майбутніх роботодавців.

Чудовий ресурс Squarespace для створення вражаючого портфоліо, яке допоможе вашій роботі виділитися. Ця платформа має найкращі інструменти дизайну будь-якого конструктора веб-сайтів, що дозволяє створити професійний веб-сайт за лічені години.

д) брошура

Веб-сайти брошур схожі на цифрові візитні картки. Ці типи веб-сайтів, що використовуються в основному для малого бізнесу, використовуються для реклами послуг та відображення контактної інформації лише на декількох сторінках.

Наприклад, невелика сантехнічна компанія створить веб-сайт брошури із домашньою сторінкою для відображення контактної інформації, сторінки «про нас» із описом компанії та, можливо, декількох фотографій своєї роботи.

Якщо у вас немає часу чи ресурсів для створення веб-сайту самостійно, ми можемо допомогти вам знайти веб-дизайнерів відповідно до ваших потреб. Просто скажіть нам, що ви хочете від вашого наступного веб-сайту, і ми підберемо вас до перевірених дизайнерів, які запропонують вам необов’язкові ціни для порівняння [13].

е) краудфандинг

Краудфандінг це практика фінансування проекту або підприємства шляхом залучення невеликих сум грошей від безлічі різних людей. Ці типи веб-сайтів стають джерелом для нових стартапів.

У минулому єдиним способом фінансувати нове підприємництво було шукати великі інвестиції лише з кількох людей (думаю, «Dragon's Den»). Але в наші дні ви можете легко створити сайт для краудфандингу - вам просто потрібно створити презентаційне відео для вашого проекту, а потім встановити цільову суму та кінцевий термін.

Користувачі Інтернету, які вірять у те, над чим ви працюєте, покладуть на вашу справу певну суму грошей. Ви також можете запропонувати заохочення в обмін на пожертви, такі як продукти зі знижкою або VIP-досвід[3].

ж) новини чи журнал

Новини та веб-сайти журналів потребують невеликих пояснень. Основна мета веб-сайту з новинами - тримати своїх читачів в курсі поточних подій, тоді як інтернет-журнали більше зосереджуватимуться на розвагах.

Ви початківець журналіст, який хоче створити присутність в Інтернеті? Тоді ви дійсно не можете помилитися з піднесеними шаблонами Wix. Ви навіть можете встановити додаток Сторінка новин на веб-сайт вашого бізнесу, який автоматично подаватиме та оновлюватиме ваш веб-сайт відповідними статтями новин. А ще краще, якщо платні тарифи стартують лише від 3 фунтів на місяць, Wix не буде коштувати вам землі.

з) соціальні мережі

Ми всі знаємо Facebook та Twitter, але сайти соціальних медіа можуть мати багато інших форм. Ці веб-сайти зазвичай створюються для того, щоб дозволити людям ділитися думками, образами чи ідеями або просто спілкуватися з іншими людьми стосовно певної теми. Сайти в соціальних мережах також дедалі частіше стають місцем, де люди можуть читати новини[16].

і) потокове телебачення або відео

Netflix, разом із подібними сайтами, як NowTV, зробили революцію в тому, як світ дивиться телевізор. За останні роки популярність цих веб-сайтів стрімко зросла, і такі веб-сайти, як BBC iPlayer та All 4, представляють більш традиційні приклади цієї конкретної теми.

к) виховна

Навчальні веб-сайти також цілком зрозумілі. Ці веб-сайти призначені для відображення інформації на певні теми, або за допомогою інтерактивних ігор, або привабливих дизайнів, щоб тримати користувача на гачку. Якщо ви хочете створити один із цих веб-сайтів, вам слід подумати про те, щоб найняти веб-розробника-фрілансера для створення цікавих інструментів, ігор або вікторин[8].

л) портал

Портали в основному використовуються для внутрішніх цілей у бізнесі, школах чи установах. Ці веб-сайти часто передбачають процес входу, що дозволяє учням отримувати доступ до веб-сайту школи або надавати працівникам доступ до їх електронних листів, сповіщень та файлів в одному місці.

Що стосується дизайну, веб-портали досить складні, тому ми рекомендуємо найняти експерта з веб-дизайну, який би впорався зі складним процесом веб-розробки.

м) вікі або форум спільноти

Вікі-сайт дозволяє людям співпрацювати в Інтернеті та писати вміст разом. Найпопулярнішим прикладом є сама Вікіпедія, яка дозволяє будь-кому змінювати, додавати та оцінювати зміст кожної статті[14].

1.3 Основні принципи побудови веб-сайту

До «веб» можна віднести сферу інформаційних технологій, спрямованих на розробку і верстку веб-сайтів. Мова тут йде Не тільки в тому вигляді, в якому буде представлений майбутній сайт, але і в створення різного роду призначених для користувача інтерфейсів.

Етапи веб-розробки.

Розуміння процесу розробки будь-якого інтернет-ресурсу дозволяє детально розібратися, що відбувається на кожному етапі.

Основні етапи створення сайтів:

- ідея;
- технічне завдання;
- макет;
- HTML + CSS верстка;
- програмування функціоналу;
- розміщення на хостингу;
- просування[17].

А тепер докладніше про деяких з них.

Перше, з чим необхідно визначитися ще на підготовчій стадії, є узгодження технічного завдання. У компаніях, які займаються розробкою і створенням сайтів, відповідальним за узгодження технічного завдання зазвичай є менеджер проекту. На даному етапі узгоджуються цілі, які буде нести майбутній проект, визначається обсяг сайту, його функціональність, структура і параметри візуального відображення.

Другий етап включає в себе структурування наявною інформацією. Займається цим також менеджер проекту в парі з веб-дизайнером. На даному етапі остаточно визначається форма сайту, його зміст і наповненість то чи іншою інформацією. Добре, коли на цьому етапі структура продумується з технічної точки зору. Положення HTML блоків, присутність на сайті списків і таблиць. Менеджер проекту продумує цілий ряд різних моментів, починаючи від складання логічної структури інтернет-ресурсу і Закінчуючи вибором Найбільш вигідного способу подачі інформації інтернет-користувачам, наприклад, створюючи таблицю можна представить результати змагань, в списку - переваги компанії, в відкриваються блоках - відповіді на питання користувачів[20].

Після узгодження технічного завдання та структурування інформації в графічному редакторі опрацьовується візуальний простір майбутнього сайту. Для навігації та прикраси використовуються вже існуючі або індивідуально Розроблені графічні елементи. Зовнішній вигляд інтернет-сторінки, що представляє собою графічний файл, розробляється дизайнером.

Макет будь-якого існуючого в мережі ресурсу створюється за допомогою HTML (HyperText Markup Language) і CSS (Cascading Style Sheets). За допомогою мови розмітки гіпертексту розробляється сама структура ресурсу і його наявних сторінок. Перенесення тексту, вставка графічних елементів, створення таблиць в HTML, форм, списків здійснюється за допомогою формування коду розмітки. Хід формування макету майбутнього мережевого ресурсу часто ще називають HTML верстkою або верстkою інтернет-ресурсу. Займається створенням коду, як правило, верстальник або програміст. HTML код майбутнього сайту можна переглядати за допомогою браузера, але це мало що дасть середньостатистичному користувачеві, адже код складається в основному з тегів розмітки, наприклад <p> - абзац, <table> - таблиця,
 - перенесення рядка, - блок[19].

На заключному етапі створення сайту здійснюється його розміщення на хостингу і пошукова оптимізація. Останній момент дуже важливий, так як розкрутка будь-якого сайту в мережі дозволяє піднятися йому на перші позиції в пошукових системах. Тут важливо наповнити сайт тематичною інформацією з використанням популярних у користувачів інтернету ключових запитів. Інформація повинна бути унікальною і корисною для користувачів, в іншому випадку інтернет-ресурс може потрапити під АГС-фільтр і випасти з пошукової видачі.

1.4 Дослідження інструментів розробки веб-сайтів

В процесі перетворення Інтернету з набору інформаційних ресурсів в інструмент ведення бізнесу, технології створення сайтів істотно змінилися. На

сьогоднішній день існує величезна кількість різних методів створення сайтів, які різняться в залежності від призначення і типу сайту, умінь і навичок розробника.

Будь-сайт має призначену для користувача і серверну частину. Призначена для користувача (або клієнтська) частина будується на HTML-розмітки, CSS-стилях і JavaScript. HTML потрібен нам для відображення контенту сайту: тексти, заголовки, зображення, таблиці, текстові блоки, нумеровані і ненумеровані списки. CSS - це стильове оформлення контенту: колір і розмір шрифту, позиціонування елементів, відображення меж об'єктів, розміри блоків. JavaScript реалізує динамічне взаємодія з користувачем: перевірка введених даних, відображення діалогових вікон, додавання і приховування HTML-елементів. Серверна частина забезпечує формування HTML-коду, збереження призначених для користувача даних, взаємодія зі сторонніми веб-сервісами[7].

Якщо говорити більш детально, то у HTML-документів є загальні правила запису та загальні поняття, що використовуються при створенні сайтів. Основою мови HTML є тег. Інформація, укладена між відкривається і закривається тегом, називається його контейнером. Атрибути повідомляють браузеру, яким чином повинен відображатися той чи інший елемент сторінки. Атрибути дозволяють зробити більш різноманітним зовнішній вигляд інформації, що додається за допомогою однакових тегів. Тег використовується для службових цілей, введена в ньому інформація не відображається у вікні браузера, однак він містить безліч даних, які вказують браузеру, як слід обробляти сторінки Стилі зазвичай зберігаються в зовнішніх файлах CSS. Зовнішні таблиці стилів дозволяють змінити зовнішній вигляд і розташування всіх сторінок веб - сайту, тільки шляхом редагування одного файлу. Стиль включає в себе всі типи елементів дизайну: шрифт, фон, текст, кольори посилань, поля, і розташування об'єкта на сторінках[11].

JavaScript є мовою сценаріїв. Сценарії JavaScript - це невеликі програми, які виконуються на комп'ютері користувача при завантаженні з сервера разом з веб - сторінками. Мова сценаріїв являє собою мову, який легко програмується, легко

працює з будь-якими браузерами. Вони широко застосовувалися для вирішення таких завдань, як, наприклад, перевірка інформації, введеної користувачем в форму, перед її відправкою на сервер або програмування відповідних реакцій на дії користувача, що роблять веб - сторінки інтерактивними. Програмний код, може бути вставлений в HTML сторінках або зберігатися в окремому файлі. Щоб вставити наявну в HTML сторінку JavaScript, використовуйте тег <script>. JavaScript - об'єктно-орієнтованою мовою програмування, тобто всі елементи на веб - сторінці постають у вигляді об'єктів.

Всі об'єкти має свої властивості, і над ним можна здійснювати певні дії. Це дозволяє розробнику отримувати доступ до будь-якого елементу веб - сторінки дуже швидко і легко[21].

Для того щоб JavaScript бачив вміст HTML-сторінки і стан браузера використовується інструмент - Document Object Model. DOM потрібно для скриптів JavaScript, які бажають спостерігати або змінити веб-сторінку динамічно.

1.5 Постановка задачі

Для того, щоб повністю розкрити дану тему та досягнути поставленої мети, слід вирішити наступні питання:

- проаналізувати, які саме типи фреймворків існують;
- провести дослідження, яке розкриває можливості фреймворків;
- визначити як впливає їх використання на якість сайтів та швидкість їх написання;
- провести аналіз доцільності розробки веб-проектів без використання css-фрейморків.

2 РОЗРОБКА ВЕБ-САЙТУ БЕЗ ВИКОРИСТАННЯ CSS-ФРЕЙМОРКІВ

2.1 Загальний аналіз найпопулярніших css-фреймворків

Як відомо, фреймворк CSS - це бібліотека коду, яка абстрагує загальні веб-дизайни та полегшує розробнику їх розробку у своїх веб-програмах. Простіше кажучи, фреймворк CSS - це колекція таблиць стилів CSS, які готові до використання.

Завдання, які виконують фреймворки:

- пришвидшує індивідуальний розвиток розробника;
- виконує функцію крос-браузера;
- застосовує хороші звички веб-дизайну;
- дає чіткі та симетричні макети;
- вони роблять робочий процес продуктивним, чистим та ремонтопридатним;[25].

Крім того, вони структуровані для використання в загальних ситуаціях, таких як налаштування панелей навігації, і часто посилюються іншими технологіями, такими як SASS та JavaScript. Головною перевагою належного фреймворка CSS є те, що він економить час, оскільки не потрібно починати все з нуля.

Фреймворк CSS Tailwind

Tailwind CSS - це високо настроювана, низькорівнева утиліта, перша платформа CSS, яка надає вам всі будівельні блоки, необхідні для побудови дизайну на замовлення, без будь-яких настирливих самовпевнених стилів, з якими потрібно боротися, щоб їх замінити. На відміну від інших фреймворків CSS (Bootstrap або Materialize CSS), він не постачається із заздалегідь визначеними компонентами. Натомість він працює на нижчому рівні та пропонує вам набір допоміжних класів CSS. Використовуючи ці класи, ви можете швидко створити нестандартний дизайн. CSS Tailwind дозволяє створити свій власний унікальний дизайн.

Причини використання Tailwind:

- немає теми за замовчуванням;
- не нав'язує дизайнерських рішень, за які доведеться боротися, для скасування;
- пропонує першочергову реалізацію нестандартного дизайну з власною ідентичністю;
- поставляється з меню попередньо розроблених віджетів, на яких можна створити сайт[27].

Фреймворк Bootstrap CSS

Bootstrap - це найкраща у світі система CSS за підтримки великої громади. Цей фреймворк побудований у HTML, SASS та javascript. В даний час Bootstrap 4.5.0 є останньою версією, що забезпечує більшу реакцію класів утиліти та нових компонентів. Він орієнтований на чуйну розробку інтерфейсів, що передує мобільним пристроям, що робить його придатним для використання на будь-якому пристрої та зручним для розробників. Bootstrap підтримує всі сучасні браузери. Найкраща перевага bootstrap полягає в тому, що цей фреймворк має чудові компоненти javascript зі спеціальними файлами або CDN.

Причини використання Bootstrap:

- bootstrap пропонує безліч прикладів та заздалегідь встановленого макета для початку;
- за допомогою Bootstrap розробники можуть легко з’єднати різні компоненти та макети, щоб створити новий вражаючий дизайн сторінки;
- багато детальної документації надається з цими макетами, щоб користувачі могли легко їх зрозуміти;
- bootstrap базується на ліцензії MIT, тому його можна безкоштовно використовувати, безкоштовно розповсюджувати, щоб ви могли розвиватися і ви могли робити свій внесок у спільноту[13].

Сторінка Gitub Bootstrap GitHub складається з понад 19 000 комітів та 2000 авторів.

Фреймворк Bootstrap CSS ідеально підходить для:

- новачків, які не знайомі з CSS, оскільки вони зможуть запустити Bootstrap без будь-яких перешкод;
- розробника, який мало знає JavaScript, і все ще може використовувати компоненти Bootstrap, не записуючи рядок у JS;
- бек-енд розробник, який хоче внести деякі зміни в інтерфейс, навіть якщо він або вона не новачок як в HTML, так і в CSS.

Materialize CSS Framework

Materialize CSS - це чуйний інтерфейсний фреймворк, заснований на матеріальному дизайні з колекціями UI-компонентів з мінімальними ефектами, на які користувачі можуть легко залучити. Materialize повністю реагує на планшети та мобільні пристрої. Це легко навчитися, а також надається відмінна документація. Цей фреймворк має велику підтримку з боку громади та позитивні відгуки. Матеріалізація CSS дозволяє налаштувати параметри за допомогою вражаючого набору кольорових колекцій[19].

Шаблони Materialize Admin, засновані на фреймворці Materialize CSS, широко використовуються у всьому світі завдяки своїй чуйності.

Якщо ви шукаєте деякі безкоштовні шаблони адміністратора / шаблони завантаження на основі дизайну матеріалу, тоді ви можете перевірити матеріалізований шаблон адміністратора.

Причини використання матеріалізувати CSS:

- сторінка документації Materialize дуже вичерпна і досить проста для початку;
- GitHub від Materialize містить понад 3800 комітів та 500 учасників;
- сторінка компонентів Materialize включає картки, кнопки, навігацію та багато інших додаткових функцій.

Даний фреймворк доступний кожному і його легко швидко використовувати.

Material Design Lite CSS фреймворк

Material Design Lite - це бібліотека компонентів інтерфейсу, створена за допомогою CSS, HTML та JavaScript. Це дозволяє вам додати зовнішній вигляд Material Design на ваші веб-сайти. Крім того, він не покладається на будь-які фреймворки JavaScript і має на меті оптимізувати для використання на різних пристроях, витончено погіршити роботу у старих браузерах та запропонувати досвід, який буде доступний відразу. Ви можете використовувати компоненти для створення привабливих, послідовних та функціональних веб-сторінок та веб-програм. Сторінки, розроблені за допомогою MDL, зможуть підтримувати всі сучасні принципи веб-дизайну, такі як портативність браузера, витончена деградація та незалежність пристрою[22].

Бібліотека компонентів MDL пропонує нові версії загальних елементів керування інтерфейсом користувача, таких як кнопки, текстові поля та прапорці, що відповідає концепціям Матеріального дизайну. Бібліотека також включає розширені та спеціалізовані функції, такі як картки, макети стовпчиків спінерів, повзунки, типографіка, вкладки тощо. MDL можна безкоштовно завантажувати та використовувати і може використовуватися з будь-якою бібліотекою чи середовищем розробки (наприклад, Web Starter Kit). Це набір інструментів веб-розробника для різних веб-переглядачів.

Причини використання Material Design Lite:

- створений Google, MDL є сучасним, простим у користуванні, має широкий спектр функцій і не має зовнішніх залежностей. Важливою перевагою є те, що MDL можна використовувати з Elm, мовою графічних інтерфейсів користувача;
- MDL забезпечує чудовий зовнішній вигляд, який, можливо, не потребує налаштування;
- завдяки своїм шаблонам для ведення блогу, Material Design Lite дає змогу розпочати ведення блогу за лічені хвилини;

- MDL пропонує багатий набір компонентів, включаючи кнопки дизайну матеріалів, текстові поля, підказки, вертушки та багато іншого.

Ідеально підходить для всіх, хто хоче писати продуктивніші, портативні та головне корисні веб-сторінки, оскільки MDL по суті робить концепцію доступною для плавного та легкого підбору[18].

Bulma CSS Framework

Bulma - це реагуючий сучасний фреймворк CSS. Цей фреймворк є вбудованим HTML, SASS CSS prospector та CSS flexbox. Bulma надає безліч варіантів, щоб пристосувати ваші вимоги за допомогою файлів sass, веб-пакетів та змінних. Bulma створений у чистому CSS. Це означає, що під час використання фреймворку вам знадобиться лише один файл .css і відсутність .js.

Цей фреймворк має кілька надзвичайно вдосконалених функцій, які допомагають зробити ваш веб-сайт більш привабливим і менш кодовим. Ви можете використовувати функції утиліти для створення темних і світлих кольорових візерунків. Він має ті самі сітки, що і bootstrap. Bulma дозволяє додати модульність SASS. Він сумісний з Font Awesome 4 і Font Awesome 5 завдяки елементу .icon.

Причини використання Bulma:

- Bulma пропонує чіткі та прості пресети, які полегшують вибір відповідно до тем, які розробник хоче вивчити;
- Bulma пропонує пристойну кількість веб-компонентів, серед яких можна просто вибрати та використовувати їх для проектування відповідно до вимог та модифікацій;
- сторінка Bulma на GitHub містить понад 1000 комітів та 600 авторів.

Foundation CSS Framework

Foundation - це вдосконалений фронтендний фреймворк CSS, вбудований HTML, CSS, SASS та javascript. Цей фреймворк має компілятор sass з більш швидким способом створення веб-сайту. Foundation має власний CLI, який можна встановити на ваш ПК / ноутбук за допомогою певних команд, і ви можете легко

переглядати. Подібна та сама структура файлів, як bootstrap Bulma та матеріалізують CSS. Це фреймворк CSS, що підходить для мобільних пристроїв, і повністю реагує на компоненти.

Багато форумів доступні для підтримки та допомагають швидко вирішити будь-які проблеми. Здебільшого цей фреймворк використовується для великих веб-додатків, для створення дивовижного веб-сайту з початковим шаблоном та для створення чудового веб-сайту з привабливим інтерфейсом користувача. Це семантично, читається, гнучко і повністю налаштовується [23].

Причини використання тонального крему:

- Foundation - це найдосконаліший фреймворк CSS, який дозволяє користувачам створювати великі веб-програми та багато іншого;
- на сторінці GitHub Фонду показано майже 2000 авторів та 17 000 комітів;
- він модульний і складається переважно з таблиць стилів Сасс;
- він постійно оновлюється для підтримки найновіших функцій, таких як сітки з підтримкою flexbox.

Цей фреймворк найчастіше використовуєтьсяпрофесійними, висококваліфікованими розробниками та дизайнерами, метою яких є створення унікального веб-сайту та хоче налаштувати структуру.

Skeleton фреймворк CSS

Skeleton - це невелика колекція файлів CSS, яка може допомогти вам швидко розробити сайти, які виглядають красиво будь-якого розміру, будь то 17-дюймовий екран ноутбука або iPhone. Це інструмент для швидкого розвитку. Ви можете швидко розпочати роботу з найкращими практиками CSS, добре структурованою сіткою, яка полегшує розгляд мобільних питань, з упорядкованою структурою файлів та супер базовими елементами інтерфейсу, такими як легкі стилі форм, кнопок, вкладок тощо[14].

Ви можете перевірити сторінку Skeleton’s Github.

Причини використання Skelton:

- легкий;
- адаптивна сітка;
- ванільний CSS;
- медіа-запити;
- скелетна фреймворк CSS.

Чудовий фреймворк для початківців, оскільки це найпростіший фреймворк.

Семантична структура інтерфейсу CSS

Семантичний інтерфейс побудований навколо унікальної мети створення спільного словника навколо інтерфейсу. Заснований на принципах природної мови, Semantic надає можливість розробникам та розробникам, роблячи код більш читабельним та легшим для розуміння. Користувачі кажуть, що з семантичним інтерфейсом легко працювати, і це просто має сенс.

Семантичний інтерфейс виділяється функціональністю, яка виходить за рамки CSS і включає спрощену налагодження та можливість визначати елементи, колекції, подання, модулі та поведінку елементів інтерфейсу[21].

Причини використання семантичного інтерфейсу:

- семантичний інтерфейс пропонує дуже добре організовану документацію. Більше того, фреймворк має окремий веб-сайт з керівництвом для початку роботи, налаштування та створення тем;
- усі класи семантичного інтерфейсу - це людські слова, і кодування нагадує написання звичайного тексту. Цей зручний підхід полегшує розуміння та розуміння структури навіть для початківців.

Pure фреймворк CSS

Pure - це набір невеликих адаптивних модулів CSS для всіх ваших потреб. Розмір Pure неймовірно малий - лише 3,8 КБ. Крім того, якщо ви вирішите використовувати лише підмножину доступних модулів, ви заощадите ще більше пропускної здатності. Він побудований на Normalize.css, Pure забезпечує макет та

стиль для власних елементів HTML, а також найпоширеніших компонентів інтерфейсу. Його мінімальні стилі заохочують писати стилі додатків поверх них.

Причини використання чистого CSS:

- дизайн Pure полегшує заміну стилів. Його мінімалістичний вигляд дає дизайнерам основу, на якій вони можуть будувати свій дизайн. Тим не менш, Pure надзвичайно легко налаштувати;
- фреймворк дуже простий. Назви класів легко запам’ятовувати, поширювати та підтримувати.

Ідеально підходить для тих, кому не потрібна повнофункціональна структура, а лише конкретні компоненти для включення у свою роботу.

UI Kit CSS Framework

UI Kit - це легкий, модульний інтерфейс CSS та веб-інтерфейс дизайну інтерфейсу, який пропонує майже всі основні функції інших фреймворків. У наборах інтерфейсу користувача є багато заздалегідь побудованих компонентів, таких як акордеон, попередження, падіння, Iconnav, анімація, відступ тощо, що показує схеми використання, варіанти компонентів та методи.

Інтерфейс користувача допомагає веб-розробникам створювати чисті та сучасні інтерфейси. Він пропонує вражаючі особливості, особливо що стосується дизайну, тут немає конкуренції з набором інтерфейсу[22].

В основному, UIKit - це майбутнє розробки додатків на платформах Apple.

Причини використання набору інтерфейсу користувача:

- легкий та модульний інтерфейс для розробки швидких та потужних веб-інтерфейсів. UI Kit визначає основні компоненти, такі як кнопки, мітки, контролери навігації та подання таблиці Набір інтерфейсу користувача містить заздалегідь побудовані компоненти, такі як Drop, Alert, Accordion, Padding, Iconnav, анімація тощо;
- допомагає розробити гнучкі, потужні та швидкі веб-інтерфейси. Він складається з вичерпної колекції компонентів CSS, HTML та JS;

- набор інтерфейсу на GitHub перелічує понад 4000 комітів;
- він розширюваний, простий у налаштуванні та простий у використанні.

У наведеному вище списку подано детальний огляд найкращих фреймворків. Кожен фреймворк має унікальні функції та різноманітність компонентів, що робить їх кращими для ваших веб-додатків. Ви напевно знайдете цю статтю корисною та заслуговує на увагу, оскільки ми висвітлили всі основні аспекти фреймворку CSS[15].

Ось декілька параметрів, які слід врахувати для правильного середовища CSS:

- перевага дизайну інтерфейсу;
- сіткова система;
- ліцензія;
- підтримка браузера;
- чуйність;
- підтримка громади.

Тим не менш, ти сам знаєш, що для тебе найкраще, тому врешті-решт важливим є твій погляд та вибір. Отже, ми сподіваємось, що цей найкращий список фреймворків CSS вам стане в нагоді та заслуговує на увагу. Після глибокого дослідження та копання ми зібрали ці дивовижні, чуйні та настійно рекомендовані CSS Frameworks, що підходять для будь-якого проекту.

2.2 Переваги використання css-фреймворків при розробці веб-сайту

Фреймворк CSS - це частина програмного забезпечення, що має безліч варіантів для використання у розробці HTML, що потенційно спрощує та спрощує розробку веб-сайту або веб-програми. Фреймворк CSS робить це, містячи заздалегідь визначені бібліотеки коду. Одним з прикладів є мережевий фреймворк, який встановлює розташування в кілька стовпців із заданою шириною пікселів, щоб ви могли зосередитись на створенні вмісту, а не на вишикуванні блоків тексту.

Але чи добре використовувати фреймворк CSS? Зрештою, іноді щось, що економить ваш час, в кінцевому підсумку спричиняє більші проблеми - робить заощаджений час марним. Ну, як і в усьому в житті, існують як плюси, так і мінуси використання фреймворка CSS[5].

Існує безліч фреймворків CSS, які добре роблять різні речі, тому ви можете знайти конкретний, який допоможе швидше досягти мети веб-розробки. Speckyboy виділяє 15 легких та мінімальних фреймворків CSS, а також є деякі фреймворки CSS у списку 20 нових фреймворків для розробників веб- і мобільних додатків.

Але чи варто використовувати ці фреймворки CSS для веб-розробки? Ось плюси і мінуси використання фреймворка CSS:

Плюси використання CSS Framework

1. Прискорює ваш розвиток

Структура CSS викладає основи для вас, щоб ви могли швидше розпочати розробку. Це робить це, надаючи вам код для повторюваних і загальних завдань, таких як скидання, тому вам не потрібно писати щоразу з нуля. І якщо ви працюєте з командою або співпрацюєте з іншими розробниками, у вас усіх буде однаковий спільний код CSS, тож спільні зусилля також пришвидшуються[6].

2. Увімкнює функцію крос-браузера

Пам’ятаєте розчарування, пов’язане з тим, що вам доводиться постійно допрацьовувати код CSS, щоб ваш веб-сайт або веб-програма виглядала однаково у всіх браузерах? Ну, ви можете попрощатися з цією досадою, використовуючи фреймворк CSS, який подбає про це за вас. Він уже закодований, щоб виглядати схожим у всіх браузерах, тому ви можете зосередитись на налаштуванні та створенні вмісту, а не на налаштуванні базового вигляду. Структура CSS також усуне специфічні помилки браузера, що є приємним плюсом.

3. Дає чіткі та симетричні макети

Структура CSS, що базується на сітці, встановлює розташування в кілька стовпців із заздалегідь визначеною шириною пікселів, щоб ви могли зосередитись на створенні вмісту, а не на вишикуванні блоків тексту. Більше не потрібно налаштовувати ширину пікселів стовпців, щоб переконатися, що вони вирівнюються, або задаватися питанням, чи відповідає ширина бічної панелі стандартам для віджетів чи зображень, або будь-яких інших неприємностей, пов'язаних із здогадуванням чи з'ясуванням, яка ширина потрібна для створення ваших стовпців.

4. Впроваджує хороші звички веб-дизайну

Структура CSS забезпечує добрі звички, такі як включення таблиці стилів друку. Він також надає вам набір селекторів, які ви будете використовувати для всіх веб-сайтів та веб-програм, які ви розробляєте разом із фреймворком, що створює послідовність у вашому веб-дизайні. Вам не потрібно вгадувати чи згадувати, що ви робили для того чи іншого сайту - це все послідовно[6].

Мінуси від використання CSS Framework

1. Обмежує свободу

Оскільки фреймворк CSS має стандартний набір сіток, селекторів та іншого коду, він обмежує те, що ви можете спроектувати: розміри макета, ширину сітки, типи кнопок, стилі та інше. Ви в основному застрягли в тому ж фреймворку, якщо дійсно хочете заощадити час та скористатися перевагами використання фреймворка CSS. В іншому випадку ви витратите час на зміну коду або вивчення нового фреймворку кожного разу, коли захочете зробити щось інше - наприклад, якщо у вас є проект, який вимагає унікальних або нетрадиційних параметрів або дизайну.

2. Додає додатковий код

Структура CSS неминуче матиме код, який вам не потрібен. Навряд чи ви будете використовувати всі функції фреймворку. Таким чином, ви застрягли в

додатковому коді, що може чи не може бути проблемою в залежності від того, наскільки легким і витонченим ви хочете зробити свій веб-сайт або веб-програму. Якщо кожен байт має вирішальне значення, вам потрібно буде увійти і видалити невикористаний код у своєму CSS.

3. Змушує використовувати семантику фреймворку

Використовуючи фреймворк CSS, ви змушені вносити семантичні зміни, особливо якщо фреймворк використовує нестандартну схему іменування. Це може дратувати, якщо у вас є улюблена унікальна система для елементів та селекторів CSS, а також використання ідентифікаторів та класів тощо, оскільки рамка CSS змусить вас використовувати її систему. Слід визнати, що для більшості це не є великою проблемою - це як навчання водінню нового автомобіля, тобто. Ви швидко адаптуєтесь, але це реальність використання фреймворка CSS. Проблема, коли це може бути більшою проблемою, полягає в тому, що ви співпрацюєте з іншими розробниками або дизайнерами або створюєте для клієнта, який згодом отримає доступ до коду; вони не уявляють, які це імена, тому їм спочатку потрібно ознайомитись із семантикою фреймворку[11].

4. Потенційна втрата часу

Якщо ви вже знайомі та ефективні з певним способом проектування та розробки, і вас змушують використовувати фреймворк CSS, який вам не знайомий, ви можете потенційно втратити час. Це може бути випадок, коли клієнт хоче використати фреймворк, з яким ви не знайомі, або співпрацювати з дизайнерами, які наполягають на певному фреймворку, або будь-якому іншому. Справа в тому, що фреймворки CSS - це реальність, вони існують там, люди про них знають, а деякі захочуть використовувати фреймворки, які вам не байдужі, і могли б виконати те саме завдання швидше, по-своєму. Так, це трохи натягнуто на мінус, але, як і на попередньому шахраї, це реальність фреймворків CSS, тому він про всяк випадок потрапляє до списку мінусів.

Плюси та мінуси CSS Framework

Зрештою, чи хороші фреймворки CSS? Звичайно, з тієї ж причини автоматична коробка передач хороша в автомобілях. Це означає, що більшість з них використовуватимуть автоматичну коробку передач, оскільки це допомагає їздити на простіший та швидший шлях. І коли ви отримуєте нову машину, ви звикаєте до неї досить швидко. Але є менша частина водіїв, які хочуть або потребують їзди за допомогою механічної коробки передач - або для економії палива, і для задоволення, і для перегонів, і для будь-яких інших причин[18].

Більшість отримають вигоду від веб-розробки та спрощення дизайну, які надають фреймворки CSS. Але буде менша підгрупа, яка дотримуватиметься самостійно виконувати ці завдання, щоб отримати точний контроль та результати - як і ручні водії із перемиканням передач.Тож чи слід вам використовувати фреймворк CSS? Для більшості дизайнерів, так, частіше за все, слід розглянути можливість використання фреймворка CSS. Однак це врешті-решт залежить від того, чи ви автоматична або ручна передача веб-дизайнерів.

2.3 Доцільність веб-розробки без використання css-фреймворків

Індустрія веб-додатків набирає величезних темпів з тих пір, як Google і Facebook запустили свої веб-програми. Якщо ви мрієте запустити власну веб-програму, але застрягли у вирішенні питання, чи потрібна вашій веб-програмі інтерфейсний фреймворк, цей блог допоможе вам чіткіше бачити речі. Щоб дійти до остаточного висновку, ми проведемо вас через усі відповідні частини процесу веб-розробки, які мають значення для цього рішення.

Веб-програма - це програмна програма, яка виконує завдання в Інтернеті за допомогою веб-браузера. На відміну від власних програм, вам не потрібно завантажувати веб-програми для використання на своєму пристрої. Веб-програми становлять основну частину Інтернету, яким ми його знаємо сьогодні. Netflix, Facebook, Gmail - це всі веб-програми, які перетворили світ, в якому ми живемо.

Технічний стек для розробки веб-додатків - це набір програмного забезпечення та технологій, які роблять веб-додаток функціональним. Весь стек

можна розділити на дві частини - передню частину, яку також називають стороною клієнта та задню частину, також звані стороною сервера[19].

Оскільки наша основна увага приділяється розробці інтерфейсу, ми дамо вам короткий підсумок інтерфейсу, а потім заглибимось у розробку інтерфейсу для веб-додатків.

Відповідальний за надання логіки веб-додатку, розробка back-end передбачає кодування для полегшення зв'язку між базою даних та браузером. Таким чином, це визначає, як функціонуватиме програма. Бек-енд розробка включає роботу з мовами програмування, базами даних та серверами.

Розробка інтерфейсу для веб-додатків - що це робить?

Передній кінець веб-програми - це все, що ви бачите та взаємодієте з ним під час використання програми. Він відповідає за створення користувальницьких інтерфейсів та забезпечення безперебійного користувацького досвіду шляхом організації та стилізації компонентів на веб-сторінці.

В основному, інтерфейс повинен представляти інформацію користувачеві бажаним способом. Як це робиться?

Основними технологіями, що беруть участь у розробці веб-додатків, є:

- HTML (мова розмітки HyperText) використовується для організації структури вмісту на веб-сторінці;
- CSS (каскадні таблиці стилів) використовується для додавання стилю та ефектів до вмісту веб-сторінки;
- JavaScript – багатозадачність[2].

Спочатку JavaScript був введений як мова програмування для додавання інтерактивності до елементів веб-сторінки. За останнє десятиліття JavaScript перетворився на екосистему, яка складається з фреймворків, програм, що виконують завдання, та багато іншого.

Можна легко створити інтерфейс для простого веб-додатку, використовуючи лише HTML, CSS та JavaScript. Цей інтерфейсний стек технологій для веб-

розробки достатній для створення веб-програми з базовими динамічними елементами управління, які можуть відповідати на запити користувачів.

Таким чином, навіть без інтерфейсних фреймворків ваш веб-додаток може працювати нормально. Однак його інтерактивність, функціональність та ремонтопридатність будуть дуже обмеженими.

Наприклад, якщо ви хочете побудувати статичну веб-програму, вам не потрібні інтерфейсні фреймворки або навіть JavaScript. Якщо ви хочете, щоб ваш веб-додаток був інтерактивним, вам знадобиться JavaScript, але вам не потрібна структура.

Технічно, використання інтерфейсного середовища для розробки веб-додатків є більшим вибором і менше необхідністю[21].

Що відбувається, коли створюється веб-додаток без інтерфейсного фреймворку? Скажімо, створюється веб-додаток Gamers Battleground для демонстрації профілів найкращих команд PUBG у світі. Домашня сторінка програми буде структурована як таблиця лідерів зі списком найкращих імен команд разом із загальною кількістю очок, які вони набрали на міжнародних турнірах.

Інтерфейс веб-застосункуматиме просту форму «створити профіль», щоб користувачі могли зареєструватися. Список на дошці лідерів можна буде натиснути, що призведе до сторінки профілю команди.Тепер на сторінці профілю буде детальніше про команду, як-от імена гравців, загальна кількість турнірів, які вони зіграли, і кількість перемог.

З часом, коли додаток стає більш популярним, виникне потреба класифікувати команди на основі їх національностей та мати окремі дошки лідерів з точки зору країни. Існуючі команди продовжуватимуть оновлювати свою інформацію, а нові команди продовжуватимуть додавати нову інформацію. Саме до цього моменту розробник не відчуватимете, що веб-програмі потрібен інтерфейс[23].

У міру розширення інтерфейсу веб-програми та збільшення кількості користувачів під час оновлення та обслуговування веб-програми можуть виникнути наступні проблеми:

Необхідність постійно додавати дані та читати дані з DOM.

Потрібно буде створити величезну кількість файлів JS та CSS, оскільки все більше і більше користувачів реєструються та оновлюють свої записи. Більше того, між цими файлами JS та CSS буде багато повторюваних даних.

Ручне написання функцій для оновлення змінних значень в елементах інтерфейсу вимагає багато часу.Внесення навіть невеликих змін вплине на весь код програми, і вам може знадобитися переписати величезну частину коду.Звичайно, можна вирішити ці проблеми за допомогою ванільного JavaScript, інвестуючи час і сили. Але як довго це буде можливо? Якщо програма стане успішнішою, інтерфейс буде продовжувати рости, і стане зрозуміло, що не можливо продовжувати додавати HTML-код безкінечно[25].

На цьому етапі розробники можуть задатися питанням, чи не допустили вони помилку, не створивши легко підтримуваний інтерфейс на попередньому етапі розробки веб-додатків.

Важливо пам’ятати про певні вказівки, перед початком розробки інтерфейс для веб-програми:

- переконатися, що код програми чистий, легкий для розуміння та ремонтопридатний;
- необхідно побудувати інтерфейс таким чином, щоб була можливість повторно використовувати компоненти для створення нових сторінок;
- користувальницький інтерфейс програми повинен мати можливість оновлюватись оновленнями в даних;
- код повинен бути написаний мовою, яка допоможе легко повідомляти деталі іншим розробникам.

У попередніх розділах ми бачили проблеми, які може становити зростаючий фронт-енд, і те, як активний підхід може мінімізувати ризик цих проблем. Найпростіший спосіб забезпечити всі суттєві особливості ефективного інтерфейсу - це використовувати один або декілька фреймворків веб-інтерфейсу.Перш ніж пояснити, як фронтенд-фреймворк корисний при розробці веб-додатків, ось коротке визначення[13].

Фронтальний фреймворк - це заздалегідь написаний комплект коду, який виконує роль будівельних лісів для вашої інтерфейсної структури. Ви можете скористатися цими готовими компонентами та додати власні файли та каталоги, щоб швидше створити бажаний інтерфейс більш зручним для обслуговування способом.Не кожному веб-застосуванню потрібна фронтед-структура, але більшості веб-додатків він потрібен для надійної роботи.

Повернімось до прикладу веб-програми Gamers Battleground, про яку ми вже говорили раніше. Ми вже перерахували кілька викликів, з якими зіткнеться цей додаток, якби він не був побудований із фронтальною структурою.А тепер давайте подивимось на іншу сторону. Наскільки іншим був би сценарій, якби ми використовували популярний фронт-енд фреймворк, такий як Angular або ReactJS, для створення програми? Це може дати вам уявлення про те, чому веб-додаткам потрібен інтерфейс[23].

Ось як фронт-енд фреймворк полегшить вам роботу, якщо ви використовуєте його для створення програми Gamers Battleground:

Компоненти вашого коду будуть багаторазовими, тому вам не доведеться створювати стільки різних файлів HTML та CSS для додавання нових функцій, таких як таблиці лідерів.

Архітектура на основі компонентів полегшить вам здійснення змін у певних елементах, не впливаючи на решту каталогів.

Особливості фреймворку, пов’язані з даними, полегшать клопоти щодо управління DOM. Додаток автоматично відтворюватиме відповідні компоненти автоматично, коли дані змінюються.

Якщо вам потрібен ще один розробник інтерфейсу, який приєднається до процесу розробки, вони вже були б цілком знайомі зі структурою коду. Це пояснюється тим, що більшість фреймворків мають подібні шаблони дизайну.

Навіть якщо у вас застрягла помилка і ви хочете її вирішити, не приєднуючись до когось іншого, ви можете отримати допомогу від величезної спільноти веб-розробників, яка працює саме з цими фреймворками[29].

Переваги використання Front-End Framework для веб-розробки

Ми бачили, як інтерфейсні фреймворки можуть усунути або вирішити проблеми, з якими може зіткнутися зростаюча веб-програма. Однак пункти, які ми згадали, стосувалися веб-прикладу, який ми придумали.

Тож, щоб дати вам більш чітке уявлення, ось більш чітке пояснення того, чому більшості веб-додатків потрібна фронтальна платформа. Ось переваги фронтальних каркасів:

1. Швидший розвиток

Фронтальні фреймворки дають вам заздалегідь написаний базовий код для різних компонентів інтерфейсу. За допомогою цього фундаменту та його багаторазових компонентів ви можете набагато швидше розробити решту фасаду.

2. Чудовий інтерфейс та UX

Фронтальні каркаси надають вам естетичні базові стилі та міцну основу для чуйного дизайну. Це полегшує вам підтримку типографіки та інших візуальних елементів узгодженими у всьому додатку.

3. Надійний та ремонтопридатний кодекс

База кодів популярних фронтед-фреймворків широко використовується і ретельно перевірена. Для більшості інтерфейсних фреймворків код є відкритим, простий в обслуговуванні і має сумісність між браузерами.

4. Простіша співпраця

Фронтальні конструкції відповідають подібним принципам архітектури та дизайну. Розробники інтерфейсу знайомі з цими моделями дизайну. Отже, іншим легко зрозуміти код коду веб-програми та працювати над ним.

5. Підтримка громади

Бути частиною спільноти завжди полегшує життя як у реальному світі, так і в індустрії розробки програмного забезпечення.

Коли ви стикаєтесь із проблемами під час роботи з інтерфейсними фреймворками, ймовірно, хтось інший вже знайшов рішення. На додаток до вирішення проблем, ці спільноти також допомагають вам з детальною документацією та простими у впровадженні темами для багатьох видів веб-додатків[30].

Чи є недоліки використання інтерфейсних фреймворків?Безумовно. Ось чому постійно тривають суперечки щодо того, чи потрібні веб-програми інтерфейсній структурі чи ні.Існує дуже мало інструментів, які є ідеальним рішенням для всіх видів веб-додатків. Фронтальні рамки, безумовно, не є одним з них. Структура дійсно має власний набір обмежень, і ви повинні врахувати ці фактори, перш ніж приймати її.

Нижче наведено деякі недоліки використання інтерфейсного фреймворку для веб-розробки:

1. Технічний борг

Для всіх розробників непросто бути повністю знайомим з базою коду фреймворку. Як результат, це може зайняти дуже багато часу, щоб з’ясувати,

наскільки це буде корисно для вашого додатка. У підсумку ви можете додати непотрібну технічну заборгованість до своєї веб-програми.

2. Оновлення у фреймворку можуть вплинути на ваш веб-додаток

Фронтальні фреймворки постійно розвиваються, тому ви можете очікувати оновлення в кодовій базі час від часу. Іноді ці оновлення можуть зіпсувати функції вашого додатка або вимагати додаткових зусиль для вирішення нових проблем.

3. Вони можуть бути занадто самовпевненими

Багато разів фронт-енд-фреймворки можуть виявлятися занадто жорсткими та сумнівними щодо своїх принципів проектування. Це віднімає гнучкість, необхідну для створення бажаного інтерфейсу користувача для вашого веб-додатку.

4. Крива навчання

Причиною того, що веб-розробники неохоче використовують фреймворки, є необхідність вивчати нову технологію з нуля. Деякі основи інтерфейсу можуть мати дуже круті криві навчання. Тож не варто вкладати стільки часу на програми, які можуть добре функціонувати без інтерфейсних фреймворків[23].

Отже, нарешті, чи потрібний веб-програмі інтерфейсний фреймворк? Тут не може бути правильної відповідді для всіх видів веб-додатків. Створюючи невеликий веб-додаток з обмеженими динамічними функціями, розумно не вдаватися до додаткових технологій. Буде достатньо HTML, CSS та JS. Для веб-додатків середнього розміру, які вимагають складних функцій, але безперебійного інтерфейсу користувача, слід принаймні використовувати базові фреймворки HTML і CSS, такі як Bootstrap або Foundation. Якщо ви хочете створити надійну, швидко реагуючу веб-програму для великої бази користувачів, використання інтерфейсних фреймворків JavaScript прикрасить вашу програму прекрасним інтерфейсом, а також мінімізує час та зусилля на розробку.

2.4 Порівняння процесу розробки веб-сайту з використанням css-фреймворків та без них

CSS Grid або CSS Framework? Що потрібно використовувати для наступного проекту? Це питання, яке часто задають веб-розробники, зокрема нові після введення їх у макет CSS Grid. Макети CSS Grid дозволяють розробникам створювати власні складні макети з абсолютним контролем лише за допомогою власних властивостей CSS, не покладаючись на будь-які фреймворки, які обмежені базовими макетами сітки з 12 стовпців, забитими правилами стилю за замовчуванням, і не пропонують багато місця для налаштування. З іншого боку, побудова макетів сітки за допомогою таких фреймворків, як Bootstrap, здається легким, без необхідності писати правила стилю CSS або медіа-запити, щоб зробити макет адаптивним.

Спочатку, коли було представлено макет CSS Grid, йому не вистачало перехресної браузерної сумісності, що змусило багатьох веб-розробників трохи вагатися щодо його реалізації, але зараз це не так! Що підводить нас до запитання, де ми оцінюємо, чи усунув макет сітки CSS необхідність у таких структурах CSS, як Bootstrap, створювати макети?

Структуру CSS також взаємозамінно називають бібліотеками CSS. Основною метою є запровадження більш стандартизованої практики веб-розробки. Використовуючи CSS Framework або бібліотеку CSS, ви можете швидко відстежувати свої зусилля з веб-розробки, оскільки це дозволяє використовувати заздалегідь визначені веб-елементи. Це означає, що якщо ви хочете додати елемент інтерфейсу для вашого веб-додатку, то все, що вам потрібно зробити, це включити в ваш проект CSS-файл та JS-файл та вказати клас HTML для елемента, який ви хочете стилізувати. Ось як працює фреймворк CSS[17].

Майже кожна фреймворк CSS має сітку, лише деякі з них мають більше сіток, деякі фреймворки CSS навіть пропонують розширені функції JavaScript навколо інтерактивних шаблонів інтерфейсу.Щоб це було зрозуміло, ми можемо

взяти приклад фреймворку Bootstrap, який, на мій погляд, є одним з найвідоміших фреймворків CSS.

Bootstrap - це зовнішній фреймворк CSS, з відкритим кодом та на GitHub, який допомагає у розробці інтерфейсу користувача основних компонентів CSS. Ви можете змінити стиль своїх кнопок, шрифтів та багато іншого, використовуючи Bootstrap.

Давайте візьмемо приклад, скажімо, вам потрібно визначити округлу кнопку для вашого веб-сайту, тепер замість того, щоб писати новий фрагмент коду, ви можете використовувати фреймворки CSS, такі як Bootstrap, щоб надати вам вже визначений код для кнопки, яку ви хочете мати.

Щоб відобразити вищезазначені кнопки на вашому веб-сайті, вам потрібно лише скопіювати наведений нижче HTML-код і вставити його у проект веб-програми, звичайно, вам доведеться вказати текст кнопки відповідно до ваших вимог, але все інше прийнято турбуватись[22].

CSS Frameworks забезпечує велику гнучкість для розробників у їхніх проектах та економить багато часу. Таким чином вони можуть розробити щось більш приємне для очей за допомогою фреймворків, не пишучи все з нуля і не турбуючись про будь-які перехресні проблеми браузера або невідповідності.

Сумісність між браузерами - це головний біль, з якою веб-розробники стикаються щодня. З таким величезним прогресом у розробці браузерів потреба у розробці веб-сторінки, яка забезпечує уніфіковану взаємодію між браузерами, зростає з кожним днем.

Пройшли ті часи, коли всі були в Safari та Internet Explorer. Багато інших браузерів бурхливо захопили Інтернет, такі як Google Chrome, Mozilla Firefox, Opera, і їх люблять більше, ніж Internet Explorer, якщо говорити про частку ринку браузерів. Мільйони людей віддають перевагу різному браузеру, і кожна компанія-браузер фокусується на тому, щоб бути різними. Саме це допомагає

йому подобатися людям. Це посилило головний біль, що виникає через сумісність браузерів.

CSS Frameworks знімає цей тягар за вас. Фреймворк не кодується для одного браузера. Якщо це так, як ви думаєте, як воно стане популярним? Це не буде. Розробник фреймворку вже знає, що ним будуть користуватися мільйони людей, які будуть сидіти в якому браузері, ніхто не знає. Отже, фреймворки нейтральні та вирішують проблеми сумісності між браузерами. Ви можете просто зосередитись на дизайні та креативних частинах веб-сайту, використовуючи фреймворк[7].

Майте на увазі, що незалежно від того, який фреймворк CSS ви використовуєте, і незалежно від того, наскільки сумісним він є з браузером. Завжди рекомендується проводити наскрізне тестування інтеграції веб-програми в різних браузерах. Процес називається перехресним тестуванням браузера, і ви можете виконувати його понад 2000 реальних браузерів у режимі реального часу за допомогою LambdaTest, онлайн-інструмента для перехресного браузера. Ви навіть можете виконати тестування на автоматизацію селену, використовуючи нашу мережеву сітку селену.

Обслуговування коду: CSS Frameworks допомагає полегшити обслуговування коду. Обслуговування коду потрібно після розробки коду і, можливо, веб-сайт також був опублікований. Ми часто вносимо зміни до коду та додаємо або модифікуємо функції певних елементів або дороблюємо веб-сайт. Нам потрібно підтримувати код у таких випадках. Технічне обслуговування - це процес впливу на частини веб-сайту, які повинні бути постійними. Оскільки фреймворки зменшують код і вносять фіксовані позначення в код, завжди легко підтримувати код[27].

Послідовність та одноманітність: Цілком очевидно, що коли у вас є щось, визначене деякими власними правилами та функціями, ви застосовуєте ці речі скрізь по всьому світу. Те саме відбувається, коли ви використовуєте фреймворк в CSS. Фреймворк має заздалегідь визначені функції для використання його

функціональних можливостей. Якщо ви хочете використовувати певну особливість фреймворку, вам доведеться використовувати ті самі функції незалежно від того, де ви сидите на планеті. Це створює послідовність у коді. Код є послідовним і єдиним протягом усього проекту, і один і той же проект можуть робити люди, які сидять в різних місцях на географічній території. Ця послідовність та одноманітність допомагають розробникам зрозуміти код та заощаджують багато часу. Це одна з основних причин того, що розробники не намагаються розробляти власний код або функціонал, а намагаються використовувати існуючі та модифікувати їх відповідно до себе. Всім легше[21].

Швидкі доставки: Цей момент є досить неявним, і ви, мабуть, уже про це здогадалися. Використання фреймворків економить багато часу на створенні проекту, а отже, доставка є дуже швидкою. Економить час завдяки використанню заздалегідь визначених функцій та функціональних можливостей, які вам потрібні. Дозвольте навести вам простий приклад, щоб показати вам те саме. Ви розробник і вам дали проект на розробку. Проект, де ви маєте можливість писати текст у полі (бажано із закругленими кутами) та підказкою, що описує його значення.

Тепер ці дії можна зробити протягом 5 хвилин, якщо ви використовуєте фреймворк CSS під назвою Bootstrap. Але ви хочете використовувати свій власний код. Отже, ви починаєте розвиватися. Залиште в стороні закруглені кутові кнопки. Вам однозначно потрібно буде витратити багато своєї голови та часу, розробляючи функціональність підказки. Це зайво збільшить ваш час до багатьох складок.

Крім того, пам’ятайте, що це цілісний проект, а не одна веб-сторінка, яку ви можете уникнути виправданням. Ваш проект міститиме багато таких функціональних можливостей, які є лише однорядковою роботою, якщо ви використовуєте фреймворк. З іншого боку, ваш клієнт не матиме жодного ефекту і не хвилює, чим ви користуєтесь. Незалежно від того, використовуєте ви фреймворк або ви хочете використовувати власне кодування, підказка - це

підказка, і саме це має значення для клієнта. Отже, краще використовувати фреймворк, щоб заощадити час і швидко доставити[14].

Великі спільноти: CSS Frameworks над CSS Grid отримує величезні переваги від своїх величезних спільнот. Ці спільноти створені через величезну кількість користувачів фреймворку та того факту, що фреймворки працюють вже давно. Зі збільшенням кількості користувачів наростають і проблеми, з якими стикаються люди. Ці люди звертаються до інших людей в Інтернеті, які, можливо, вже вирішили цю проблему або звернулися до інших людей у минулому і просто знають рішення. Цей ланцюжок триває і продовжується. Цей процес збільшив кількість користувачів, які використовують фреймворк.

Цілком очевидно, що якщо я даю вам вибір, вибирати між чимось, з чого є величезна підтримка, і тим, чия посередня підтримка є; ви будете схильні до величезної підтримки. Ця підтримка допоможе вам подолати будь-яку перешкоду, з якою стикаєтесь, і поекспериментувати з чимось своїм. У вас буде величезна кількість людей, які знають і готові підтримати.

Фреймворки не є новими: Фреймворки не є новими. Вони існують з того часу, коли розробники стикалися з цією проблемою повторення деяких складних речей знову і знову. Вони розробили основи, якими можуть користуватися багато людей та будь-яка людина по всьому світу. CSS Grid не така стара. Це трапилось довгий час після фреймворків, коли люди почали занадто часто використовувати сітки, чи слід сказати, що потреба в сітках виникла занадто багато[16].

Вік обох цих понять має значення. Оскільки фреймворки стартували до сіток, кількість проектів, які використовували фреймворки, більше. Навіть для будівництва сіток. Значна частина цих проектів була завершена, але багато проектів тривають ще довго після того, як вони були передані або опубліковані. Це тому, що завжди є поле для модифікації або вдосконалення. Ви можете почати додавати нові функції до своїх веб-сайтів або модифікувати старіші до гарного вигляду. Якщо ваш проект став хітом, ви напевно затримаєтесь із ним. У цих

проектах використовувались фреймворки, і якщо нова людина працює над цим, він звикне до фреймворків. Можливо, він використовує те саме в наступних проектах, над якими буде працювати. Це збільшує використання фреймворків, і все підключається. Чим більше проектів на фреймворках, тим більше людей працюють над ними, тим більше спільнота, тим краще рамки[20].

Документація: Рамки CSS, які вже були розроблені, містять документацію, яка допоможе навчитися ефективно використовувати структуру. Цей процес недооцінений, але дуже важливий, оскільки займає багато часу на розробку документації. Вважайте, що ви не використовуєте фреймворк, а більше зацікавлені в розробці власного. Один каркас зроблено; Вам також потрібна документація, щоб навчити всіх, як працює фреймворк. Якщо ви збираєтеся завантажувати свій фреймворк в Інтернет як з відкритим кодом або для використання іншими людьми, вам потрібна якась платформа, яка розповість їм, як працює ця фреймворк.

Тут корисна документація, яка часто сприймається як само собою зрозуміле, коли ми вивчаємо нову структуру. Як тільки ми знайдемо нову структуру, ми негайно перейдемо до документації, щоб дізнатись про неї. Це базовий крок, але коли справа стосується нашої власної документації, ми розуміємо її важливість. Більше того, цю документацію потрібно оновлювати у міру оновлення вашого фреймворку. Спочатку створення документації може здатися певним болем, але це дуже допомагає та полегшує вашу роботу в довгостроковій перспективі. Отже, загалом наявність документації фактично прославляє структуру та її використання[16].

Недоліки CSS Frameworks

Змушує працювати по-своєму: Framework обмежує спосіб кодування та спосіб роботи. Обговорена структура розроблена для всіх і пам’ятає про кожного розробника. Це обмежує спосіб, яким ви мали працювати або могли працювати. Тепер вам потрібно працювати так, як каркас хоче, щоб ви працювали. Сюди

входить проектна частина. Ви не можете спроектувати щось, про що ви думали своїм розумом. Вам потрібно використовувати фреймворкові класи, які вже створили цю частину. Це може бути кнопка, простий макет або що завгодно, що вам заманеться. Можна стверджувати, що так, ви можете змінити цю річ, закодувавши її поверх фреймворку, але який сенс використовувати фреймворк, коли вам так багато потрібно використовувати поверх нього. Це породжує уніфіковані веб-сайти, які ми будемо обговорювати як наступні шахраї.

Уніфікований веб-сайт: використання фреймворку обмежує спосіб, необхідний для проектування дрібних елементів дизайну. Використання класів фреймворку дасть вам однакову структуру більш-менш кожного елемента. Оскільки кожен великий елемент є поєднанням багатьох дрібних елементів, великі елементи стають однаковою структурою. Це робить два веб-сайти, що використовують однакову структуру як уніформу. Не зовсім, але ви можете легко визначити, подивившись на конструкції[8].

Уніфіковані веб-сайти - це завжди погано для розробника веб-сайтів. До тих пір, поки у вас немає веб-сайту, де відвідуваний вами трафік є обов’язковим, як-от веб-сайт про вступ до коледжів чи урядовий веб-сайт, вам не потрібно мати іншу та привабливу структуру. Перше враження про користувача - це завжди те, як ви структурували свій веб-сайт і як він відображається. Досить часто два веб-сайти на певну тему, такі як Новини чи Цифрові продукти, будуть виглядати однаково, якщо вони використовують одну і ту ж структуру.

Змушує вас вивчити нову структуру: У вищезазначених розділах я згадав, що сьогодні на ринку доступно багато фреймворків, які працюють над CSS. На початку цього допису я перелічив декілька. Зараз очевидно, що кожна структура по-своєму відрізняється. Подібно до різних браузерів, які ми маємо на ринку. Оскільки фреймворків так багато, очевидно, що ви повинні мати свій власний улюблений фреймворк і вже деякий час працюєте над ним[9].

Це створює проблеми, коли ваш клієнт хоче, щоб ви працювали за іншою структурою. Це означає, що він хоче, щоб його проект розроблявся з використанням іншої основи, про яку ви навіть не уявляєте. Тепер вам доведеться вивчити абсолютно новий фреймворк лише для одного проекту, і, можливо, ви більше ніколи не будете використовувати цей фреймворк. Це втрата часу для розробника, але варіанту немає.

Для новачків у веб-розробці, які ще не освоїли CSS, надають перевагу використанню фреймворків CSS. Такі фреймворки, як Bootstrap, надзвичайно прості в освоєнні і можуть створювати цілі макети веб-сторінок за лічені секунди без необхідності писати будь-який CSS-код. Пропонуючи безпрецедентну простоту використання, блискавично швидке розгортання, майже 100% підтримку крос-браузера, стандартизацію кодової бази та легку криву навчання, роблять CSS-рамки привабливим вибором для вас. Якщо ви прагнете створити стандартні макети, сумісні з крос-браузером, швидкими темпами, а не складними найсучаснішими макетами, вам слід вибрати рамки CSS над сіткою CSS[10].

Більше того, фреймворки CSS, такі як Materialize або Bootstrap, є набагато більше, ніж Grid-системи. Насправді, сітки є лише крихітною особливістю фреймворка CSS. Якщо ви хочете створювати різні компоненти інтерфейсу користувача, такі як панель навігації, карусель, повзунки, картки, акордеони, вкладки, спливаючі вікна, модалі, кнопки тощо з блискавичним темпом із мінімальним стилем CSS, тоді вам слід використовувати фреймворки CSS.

Тепер, коли ми добре розуміємо, що таке фреймворк CSS! Пора зупинитися на макеті CSS Grid.

CSS Grid - це техніка макетування, яка використовує сітку на вашій веб-сторінці. Це техніка, яка описана лише в CSS, а не в HTML. Отже, щоб відповідати потребам веб-розробників, сітки CSS забезпечують легкість розробки складного та крученого адаптивного веб-дизайну. Якщо ви веб-розробник або веб-тестувальник, ви вже знаєте актуальність адаптивного веб-сайту. Перш ніж

розпочати свій наступний веб-проект, не забудьте уникнути цих основних помилок при адаптивному веб-дизайні.

Наведене зображення показує різні сітки для різних елементів веб-сторінки, таких як заголовок, бічна панель тощо.

Як ви можете помітити, CSS Grid допомагає у вирівнюванні елементів відповідно до веб-розробника. Вибір CSS Grid проти CSS Frameworks може дозволити вам доставити складну веб-програму з великою кількістю таблиць за дуже короткий час[22].

Хоча CSS Grid ніяк не може зрівнятися з універсальністю CSS Frameworks, іноді розробник повинен думати про два для одного конкретного проекту або певних частин проекту. CSS Grid розвивався краще, ніж до розширення своїх функціональних можливостей. Враховуючи, що вам потрібно створити макет, який приваблює візуально, дуже просто, якщо ви вибрали CSS Grid. Використовуючи макет CSS Grid, він мінімізує зусилля та складність макета. Отже, якщо вам потрібно буде змінити структуру в будь-який час пізніше, інші структури більше не постраждають.

Наприклад, припустимо, у вас є дві сітки поруч і зображення після сіток в одній площині. Тепер, у майбутньому, якщо ви хочете додати ще одну сітку в тій же площині, сітка CSS розділить необхідний простір порівну, не впливаючи на координати зображення.

Сітки дуже пристосовані до відведеного їм простору. Це допомагає елементам ніколи не перекриватись або створювати певні проблеми з макетом.

Переваги розмітки сітки CSS

Створення складних 2D-сіткових систем: використання CSS Grid проти CSS-середовища ідеально підходить у випадках, коли вам потрібно створити складні двовимірні макети, які популярні фреймворки, такі як bootstrap, не можуть запропонувати. Створення складних макетів, таких як - Мозаїчна фотогалерея або

Масонські макети, неможливо створити без використання сітки CSS. Більшість фреймворків CSS пропонують базові 12-стовпчасті адаптивні сітки, які мають обмежене застосування, непридатне для сучасного сучасного дизайну[12].

Зменшений розмір коду: Ще однією головною перевагою використання сітки CSS над структурою CSS є зменшений розмір коду. За допомогою CSS Grid ви можете створити будь-яку форму системи розмітки сітки, яку можна собі уявити, лише покладаючись на власні правила стилю CSS. З іншого боку, для реалізації макетів Grid із використанням фреймворків CSS, весь файл CSS фреймворку / бібліотеки повинен бути пов'язаний з вашим проектом.

Це непотрібне програмне забезпечення, яке додасть до розміру вашого проекту та зменшить швидкість завантаження вашого веб-сайту. Отже, коли користувач відвідує ваш веб-сайт, повний список файлів фреймворку / бібліотеки отримується із сервера разом з іншими ресурсами до завантаження веб-сторінки. Це різко збільшить час завантаження вашого веб-сайту та пошкодить рейтинг SEO на SERP та вищий показник відмов. CSS Grids, на відміну від фреймворку, не вимагає роботи зовнішньої таблиці стилів. Це допомагає швидше завантажувати ваш веб-сайт, і навіть при повільному з'єднанні вам не потрібно турбуватися.

На макет це не впливає: веб-сторінка складається з багатьох елементів. Ці невеликі елементи поєднуються разом, утворюючи макет веб-сторінки. Макет - це те, що ви бачите і як бачите. Наприклад, я розмістив два зображення поруч і поруч із ними рухому колонку новин. Зараз тут є три елементи, і вони потрапляють під макет. Тепер, якщо я хочу змінити розмір першого зображення, можливо стовпець новин переміститься в наступний рядок. Це перемістить елементи, присутні в наступному рядку, і це вплине на весь макет. Це може статися в будь-який час, оскільки зміна розташування елементів є дуже поширеною справою веб-розробки. У цьому допомагає сітка CSS. Коли ми використовуємо CSS Grids, зміни, які ми робимо після розробки, не впливають на макет веб-сайту. Тепер замість зображень у мене CSS Grid і я змінюю розмір сітки, пізніше це взагалі не вплине на панель новин стовпців.

Швидший розвиток: рамки величезні і важкі, але сітки короткі та легкі. Їх легко і швидко вивчити. Фреймворк має багато речей всередині. Повна структура вашого веб-сайту. Ці речі часто пов’язані між собою. Хороший користувач фреймворку використовуватиме все це для розробки чогось унікального, оскільки фреймворки часто звинувачують у своїх монотонних макетах. Це займе час, і цей час змінює часові межі для ваших проектів, якщо такі є. Це не стосується сіток[18].

Сітки дуже легкі і містять дуже мало понять у порівнянні з каркасами. Ви можете легко ознайомитися з поняттями в сітці та розпочати розробку. Немає складнощів у розробці мереж. Вам не потрібно мати різні елементи, пов’язані між собою. Сітки не дають монотонних макетів. Це повністю залежить від розробника. Отже, розвиток сітки відбувається швидше, ніж рамки.

Розвиток: CSS Grids ще не розробляється таким чином, як розробляються основи. Це пов’язано з тим, що сітки є досить новими, і багато проектів було розпочато ще до того, як вони з’явились у картині. Це означає, що якщо ви працюєте над цим проектом (у компанії чи з відкритим кодом), ви неодмінно схиляєтесь до основ. Якщо ви щось досить добре знаєте, ви точно спробуєте реалізувати цю річ.

Як вже зазначалося раніше, CSS Grid та CSS Framework - це два дуже важливі аспекти CSS. Хоча деякі впевнені, що вони будуть використовувати CSS Framework, інші впевнені в CSS Grids. Є також люди, які вирішують залежно від потреби про те, до чого вони рухатимуться. Оскільки ми побачили деякі переваги сіток, давайте подивимося, наскільки корисними є фреймворки CSS.

Крос-браузерна сумісність сітки CSS

Найбільшою перешкодою, з якою стикається CSS Grid на шляху до більш широкого прийняття спільнотою веб-розробників, була погана сумісність та підтримка між браузерами. Кілька років тому до 2017 року CSS Grid

підтримувався лише Google Chrome і Firefox, тоді як Internet Explorer, Microsoft Edge, Opera і навіть Safari не забезпечували підтримку браузера для сітки CSS.

Використання сітки CSS у готовому до виробництва коді означало б або обслуговування масивної користувальницької бази, яка все ще використовує застарілі невлаштовані та поламані веб-сторінки, або марнотратство годин, щоб якимось чином кодувати можливі резервні копії float / flexbox. Ось чому більшість розробників застрягли у використанні макетів Grid на основі Float або Flexbox або покладались на фреймворки CSS, такі як Bootstrap, Materialize або Bulma, щоб створити макети сітки з 12 стовпців. Саме тут фреймворки CSS мали ключову перевагу над CSS Grid і продовжували своє панування[9].

Однак з 2017 року CSS Grid спостерігає значне покращення підтримки браузера. Це повністю знищило ключову перевагу CSS-фреймворків, якими користувалися раніше. Станом на листопад 2019 р. Сітку CSS підтримують 93,85% браузерів (Чи можу я використовувати статистику). Раніше не підтримувані браузери, такі як IE (v10-11), Edge (v16 +), Opera (v44 +) та Safari (v10.1 +), тепер підтримують сітку CSS, включаючи мобільні браузери як для Android, так і для iOS.

Одним з найбільших аргументів проти використання фреймворків CSS, таких як Bootstrap, Materialize або Bulma, є надмірне здуття коду. Тонни непотрібних правил стилістики CSS, які ніколи не будуть використані, обтяжать ваш проект. Якщо ви хочете зберегти свій веб-сайт компактним за розміром для підвищення швидкості завантаження та продуктивності, найкращим вибором буде CSS Grid. Ще однією причиною вибору сітки CSS над фреймворком CSS є те, що з такими фреймворками, як bootstrap, ви застрягнете у примітивній системі сіток із 12 стовпців. Досягти макета стовпців 5 або 7 або 9 було б дуже складно[19].

Цю проблему повністю усуває сітка CSS. Якщо ваш проект вимагає дуже спеціального макету, якого неможливо досягти за допомогою базової 12-колонової сітки, сітка CSS підходить саме для ваших потреб. Третім аргументом

на користь сітки CSS є те, що вона пропонує вам неперевершену гнучкість для перестановки розділів / областей. За допомогою систем Grid Framework CSS ви можете досягти лише основних адаптивних змін у макеті відповідно до розмірів екрану. Але CSS Grid пропонує набагато потужнішу гнучкість для завершення зміни структури макета як у горизонтальному, так і у вертикальному напрямку.

Наприклад, - переміщення вертикального меню бічної панелі з боку вгору, розміщеного горизонтально. Останньою причиною, через яку вам може знадобитися вибрати сітку CSS над фреймворками CSS, є отримання перепочинку від потворної розмітки HTML за допомогою постійно зростаючих імен класів та тегів div, які роблять ваш HTML-код безладним і важким для читання. За допомогою сітки CSS ваша розмітка HTML залишатиметься незайманою.

Майбутнє CSS Grid проти CSS Frameworks. Це питання виникає у свідомості багатьох розробників, які звикли використовувати фреймворки та сітки для їх використання.

CSS Grids продовжують набирати популярність з тих пір, як вони стали сумісними з переглядачами. Також здійснюються основні розробки мереж, які запровадили підмережу CSS. Це дасть вам змогу розробляти веб-сайт на більш детальний рівень, ніж ваш звичайний макет CSS Grid, кращий та легший макет із вдосконаленим дизайном CSS. Однак сумісність перехресних браузерів для CSS Subgrid на сьогоднішній день виглядає не так добре.

Однак, якщо макет сітки CSS був прийнятий різними браузерами, цілком природно вірити, що CSS Subgrid не буде занадто довгим. Суворі користувачі макета CSS Grid сподіваються, що CSS Subgrid найближчим часом стане сумісним із переглядачами[20].

На сьогодні CSS Frameworks приймаються багатьма величезними фірмами. Причиною є його популярність. CSS Frameworks дуже популярні і сьогодні використовуються більшістю розробників через економію часу та енергії. Однак майбутнє фреймворків CSS виглядає дещо похмурим завдяки введенню CSS

Subgrids. Однак на сьогодні веб-розробники працюють над фреймворками, які будуть набагато легшими за вагою та матимуть можливість змінювати макет веб-сайту відповідно до використання.

Суворі користувачі фреймворків CSS вважають, що сітка CSS повинна розчинитися, щоб стати частиною фреймворків CSS. Хоча багато хто погоджується з цим, проблема полягає в тому, що вага фреймворку збільшиться набагато більше, що ми хочемо зменшити. Але майбутнє можна побачити лише в майбутньому, і ми побачимо, чи злиються ці дві дороги в найближчі роки.

ВИСНОВОК

У роботі проведено дослідження найбільш затребуваних і ефективних в розробці інтернет-ресурсів фреймворків, таких як: Bootstrap, Foundation, Skeleton, Grid System, HTML KickStart, Gumby Framework. Проведено факторний аналіз, що дозволив виявити найбільш оптимальний інструмент для створення інформаційного ресурсу. Результати роботи можуть бути використані веб-дизайнерами при проектуванні і розробці сайтів. Рекомендації щодо найбільш ефективного використання CSS-фреймворків, отримані в результаті проведеного дослідження, були використані при створенні інформаційного ресурсу.

В роботі проведено аналітичний огляд літературних джерел, досліджені основні засоби реалізації даної системи, що існують на сьогодні в області вебдизайні. Проаналізовано відомі структури та функціональні можливості таких систем, перспективи їх розвитку. Розглянута проблематика можливих порушень в роботі системи та прийняття відповідних заходів по забезпеченю вирішення цих питань.Ви дізналися, що в it-галузі існує безліч css-фреймворків і бібліотек, і важко визначити, що саме вибрати з них для вирішення конкретного завдання. Внаслідок чого для спрощення цього процесу я зібрався зробити для Вас їх повноцінний аналіз і порівняння на живих прикладах.

В ході дослідження було наочно доведено, чим відрізняється css-фреймворк від css-бібліотеки, і виділено 4 типи рішень:

- прості css-фреймворки;
- Web-components;
- Css-сітки;
- комплексні css-фреймворки.

Було досліджено питання доцільності розробки веб-проектів без використання css-фрейморків, і зроблено висновок, що залежно від типу проекту та рівня підготовки розробника доцільно буде використовувати певний вид css-фрейморків, або ж взагалі обійтися без їхнього використання. Було втсановлено,

що css-фрейморки значно спрощують процес розробки, проте цілком можливою залишається можливість розробки без їхнього використання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Апопій В. В. Інтернет-торгівля: проблеми і перспективи розвитку / Апопій В. В. // Регіональна економіка. - 2003. - № 1. - с. 25.
2. Агалаков С.А. Статистические методы анализа данных [Текст] / С.А. Агалков. – 18 с.
3. Айлебрехт Л. Web-сервер Apache: Пер. с англ. - М.: Новое знание, 2002. – 592 с.
4. Аргерих К., Чой В., Коггсхол Д., Эгервари К., Сколло К. Профессиональное PHP программирование. 2-е издание: Пер. с англ. - М.:Символ-Плюс, 2003. - 1048 с.
5. Бин Д. XML для проектировщиков: Пер. с англ. - М.:Символ-Плюс, 2004. - 256 с.
6. Боуэн Р., Лиска А. Apache. Настольная книга администратора: Пер. с англ. - М.:ДиаСофт ЮП, 2002. - 384 с.
7. Балабанов И. П. Интерактивный бизнес / Балабанов И. П. - СПб.: Питер, 2001. - 128 с
8. Бруннер М. Принципы электронного бизнеса / Бруннер М. - М.: Мир электронной коммерции, 2000. - 453 с.
9. Bielievtsov, I. Ruban, K. Smelyakov, D. Sumtsov Network technology for transmission of visual information // Selected Papers of the XVIII International Scientific and Practical Conference on Information Technologies and Security (ITS 2018). – CEUR Workshop Processing. – Kyiv, Ukraine, November 27, 2018. – Pp. 160-175.
10. Використання освітніх веб-ресурсів [електронний ресурс] // Режим доступу: http://galanet.at.ua/publ/1-1-0-23– Назва з екрану.
11. Leshchynskyi Volodymyr. Principles of explanation in e-commerce system based on sales dynamics. COMPUTER AND INFORMATION SYSTEMS AND TECHNOLOGIES KHARKIV, APRIL 2020, p.76-77.

12. Гончаров А. Ю. Web-дизайн: HTML, JavaScript и CSS. Карманный справочник.. — "КУДИЦ-ПРЕСС", 2007. —320 с.
13. Гарсиа-Молина Г., Ульман Дж., Уидом Дж. Системы баз данных. Полный курс. — М.: Вильямс, 2003. — 1088 с.
14. Гешвинде Э., Шенинг Г. Разработка WEB- приложений на PHP и PostgreSQL: Руководство разработчика и администратора: Пер. с англ. - М.:ДиаСофт ЮП, 2002. - 608 с.
15. Етапи створення веб-ресурсів [електронний ресурс] // Режим доступу: http://edufuture.biz/index.php?title=Етапи створення вебресурсів – Назва з екрану.
16. Интернет-аудитория Украины. Статистика 2012-2013 и прогноз на 2014 год [Электронный ресурс] / Netpeak. – Режим доступа к ресурсу: URL:http://blog.netpeak.ua/internet-auditoriya-ukrainy-statistika2012-2013-i-prognoz-na-2014-god.
17. Мейерт Д.О. Небольшая книга о HTML/CSS фреймворках [Текст] /Д.О. Мейерт // Орейли. – 2015. – 30 с.
18. Ньюкомер Э. Веб-сервисы. Для профессионалов: Пер. с англ. - С.-Петербург: Питер, 2003. - 256 с.
19. Мартин Д., Бирбек М., Лозген Б., Пиннок Д., Ливингстон С. XML для профессионалов: Пер. с англ. - С.-Петербург: Лори, 2001. - 900 с.
20. Уорсли Д., Дрейк Д. PostgreSQL. Для профессионалов: Пер. с англ. - С.-Петербург: Питер, 2002. - 496 с. 72
21. Стоунз Р., Мэттью Н. PostgreSQL. Основы: Пер. с англ. - М.:Символ-Плюс, 2002. - 640 с.
22. Макконнелл С. Совершенный код: Практическое руководство по разработке программного обеспечения: Пер. с англ. - С.-Петербург: Питер, 2005. - 896 с.
23. Цвики Э., Купер С., Чапмен Б. Создание защиты в Интернете (2 издание): Пер. с англ. - М.:Символ-Плюс, 2002. - 928 с.
24. Козье Д. Электронная комерция: Пер. с англ. - Москва: Издательськоторговий дом "Руская редакция". 1999. - 288 с.: ил.

25. Кобелев О.А. Электронная коммерция: учеб. пособие / С.В. Пирогов (ред.). — 3-е изд., перераб. и доп. — М. : Дашков и Ко, 2008. — 683с
26. Скотт Хокинс. Администрирование веб-сервера Apache и руководство по электронной коммерции = Apache Web Server Administration and e-Commerce Handbook. — М.: Вильямс, 2001. — 336 с.
27. Кузнецов С. Д. Основы баз данных. — 2-е изд. — М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. — 484 с.
28. Хабибулин И.Ш. Самоучитель XML: Пер. с англ. - С.-Петербург: BHV-СПб, 2003. - 336 с.
29. Якоб Нильсен Веб-дизайн. — СПб: Символ-Плюс, 2003. — 512 с .
30. Яковлев Алексей Александрович. Раскрутка и продвижение сайтов: основы, секреты, трюки.. — СПб.: БХВ-Петербург, 2007. — 336 с