МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

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

Бакалаврська кваліфікаційна робота

на тему: Розробка веб-сайту інтернет магазина з продажу одягу

Виконав студент 2 курсу групи К-25
Спеціальність 122 комп'ютерні науки
Клюшник Ігор Валентинович

Керівник асистент
Бучинська Ірина Вікторівна

Консультант к.геогр.н., доцент
Коваленко Людмила Борисівна

Рецензент к.т.н., доцент
Гнатовська Ганна Арнольдівна

Одеса 2020

ЗМІСТ

СПИСОК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

Акаунт – обліковий запис у комп'ютерній системі − сукупність наданої інформації про користувача, засобів та прав користувача відносно багатокористувацької системи;

Браузер – програмне забезпечення для комп'ютера або іншого електронного пристрою, як правило, під'єднаного до Інтернету, що дає можливість користувачеві взаємодіяти з текстом, малюнками або іншою інформацією на гіпертекстовій веб-сторінці;

Домен – це область (зона) простору ієрархічних імен мережі Інтернет, яка обслуговується набором серверів доменних імен і централізовано адмініструється;

ПС – пошукова система;

AJAX – Asynchronous JavaScript and XML;

BOM – Browser Object Model;

CMS – Content Management System – система керуванням контенту сайта;

Cookie – у комп'ютерній термінології поняття, яке використовується для опису інформації у вигляді текстових або бінарних даних, отриманих від веб-сайту на веб-сервері, яка зберігається у клієнта, тобто браузера, а потім відправлена на той самий сайт, якщо його буде повторно відвідано;

CSS – Cascading Style Sheets – каскадні таблиці стилів;

DNS – Domain Name Service – служба доменних імен;

DOM – Document Object Model;

HTML – HyperText Markup Language – мова розмітки гіпертексту;

HTTP – HyperText Transfer Protocol – протокол передачі гіпертексту;

IP – Internet Protocol address, це ідентифікатор (унікальний числовий номер) мережевого рівня, який використовується для адресації комп'ютерів чи пристроїв у мережах;

JS – JavaScript – скриптова мова програмування;

JSON – JavaScript Object Notation

SQL декларативна мова програмування для взаємодії користувача з базами даних;

URL – Uniform Resource Locator – єдиний вказівник на ресурс;

WEB – World Wide Web всесвітня павутина;

XML – Extensible Markup Language – стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет.

ВСТУП

На сьогоднішній день практично кожна організація має власний веб-сайт. В умовах використання сучасних інформаційних технологій – це необхідний чинник існування, що дозволяє розширити поле рекламної діяльності і привернути тим самим додаткових клієнтів.

Метою кваліфікаційної дипломної роботи є розробка унікального, який не має аналогів, повністю функціонуючого веб-сайту з адаптивною верстkою, що спеціалізується на ознайомленні клієнтів з продукцією та продажу товарів одягу.

Об'єктом дослідження кваліфікаційної роботи є веб-сайт. Сайт "Zycsel Store" розроблений для реалізації інтернет-магазину. Для його перегляду та експлуатації необхідна програма-браузер та з’єднання з мережею Інтернет. Користуватися сайтом або адмініструвати може людина, що має навички звичайного користування мережі Інтернет і досвід роботи з Інтернет-браузером (Opera/Google Chrome/Mozilla Firefox/Internet Explorer).

Створення і розробка сайтів включає:

- затвердження первинного технічного завдання на розробку сайта;
- визначення структурної схеми сайту − розташування розділів, контенту і навігації;
- веб-дизайн – створення графічних елементів макету сайту, стилів і елементів навігації;
- розробка програмного коду, модулів, бази даних і інших елементів сайту необхідних в проекті;
- тестування і розміщення сайту в мережі Інтернет.

Даний програмний продукт дозволить мені продавати свої товари в мережі Internet за допомогою ознайомлення клієнтів з товарами одягу, сортування товарів за посиланнями на номер телефону та соціальну мережу

Instagram, а також дозволить клієнтам дізнатися свій розмір для покращення вибору товару.

Кваліфікаційна робота містить Вступ, 4 глави, висновки, 29 рисунків, 0 таблиць, 12 посилань.

1 ЗАГАЛЬНІ ВІДОМОСТІ ПОШУКОВОЇ СИСТЕМИ

1.1 Призначення розробки

Веб-сайт − це сукупність програмних, інформаційних, а також медійних засобів, логічно пов'язаних між собою. По суті ж веб-сайт − це віддзеркалення успішності фірми, її обличчя. [1][1)]

Веб-сайт виконує такі основні завдання:

- реклама продукції, послуг, ідей (правильно зроблений веб-сайт із легкістю приведе клієнта до висновку про необхідність покупки товару, або послуг, або ідей, що пропагуються на ньому);
- продаж товарів, послуг, інформації, ідей (у сучасної людини немає багато часу для походів по магазинах, тому можливість замовлення товарів і послуг, не відходячи від комп'ютера, значно розширює можливості і клієнта, і продавця);
- безкоштовне надання інформації або послуг (насправді надання інформації або послуг − це засіб залучення відвідувачів до даного ресурсу для здобуття, наприклад, статистичної інформації або ж для показу реклами, якщо це рекламний майданчик);
- підтримка клієнтів;

Найпопулярніші типи сайтів:

- сайт-візитка − найпростіший сайт, який не містить багато сторінок;
- сайт компанії з каталогом продукції За своєю суттю − це теж сайт-візитка;
- інтернет-магазини;
- новинні та пошукові портали;

[1] [1] Використання освітніх веб-ресурсів. URL: http://galanet.at.ua (дата звернення 27.04.2020).

– інформаційні портали;
– веб-системи обліку товарів, бухгалтерії.

Головне призначення програмного продукту – це надання інформації користувачам у мережі інтернет. У наш час інтернет-магазин – це додаткова, а часом і основна можливість розвитку вельми успішного бізнесу. Адже саме ця форма торгівлі доступна цілодобово. У будь-який зручний час потенційний клієнт може зайти на сайт компанії, ознайомитися з каталогами, уважно вивчити спеціальні пропозиції, акції та інші плюси даного ресурсу. І тут же зробити замовлення, вибравши зручний спосіб оплати і доставки. Інтернет магазин робить підприємницьку діяльність більш мобільною, здатною швидко реагувати на потреби ринку, використовувати всілякі ресурси мережевого простору. Інтернет робить пропоновані інтернет-магазином товари доступними величезній кількості людей на всій території країни. Програмний продукт призначений ознайомлювати користувачів з товарами Zycsel Store та надавати можливість за користувачам оформлювати та сплачувати замовлення цих товарів. Також веб-ресурс відповідає за просування товарів на ринку, збільшення обсягів продажу та залучення нових покупців, які активно користуються мережею інтернет.

1.2 Вимоги до програмного виробу

Після етапу виготовлення, програмний продукт має відповідати визначеним вимогам. У першу чергу мають бути виявлені та виправлені усі помилки у фронт-енд та бек-енд частинах веб-сайту. Наявність привабливого дизайну сайту, що робить процес покупки приємнішим, також зручного і простого функціоналу. До нього належать: пошук, сортування, фільтрування і підбір товарів або послуг. Також це може бути публікація новин, комунікація з клієнтами та SEO-функції, що покращує якість обслуговування і приваблює клієнтів. Сайт інтернет-магазину відкривають з різних пристроїв. Тому

необхідно створити адаптовану верстку. Користуватися сторінками такого магазину буде набагато зручніше. Веб-сайт має відповідати умовам пошуку потрібних пошукових систем. [2][1)]

1.3 Вимоги до функціональних характеристик

Інтернет-магазин повинен видавати користувачеві готову веб сторінку, сформовану за запитом користувача на сервері, яка відображає необхідну інформацію та кнопки для здійснення наступних запитів.

Програмний продукт повинен виконувати наступні дії:

- при обиранні конкретної категорії, сайт повинен відображати всі наявні товари цієї категорії та їх ціни;
- при обиранні конкретного товару сайт повинен надавати користувачеві розширену інформацію про цей товар;
- при використанні спеціального пошуку, відображати конкретні товари, що відповідають обраним користувачем фільтрам;
- можливість авторизації на сайті для повного доступу до функціоналу останнього та більш швидкого процесу замовлення послуг;
- короткий опис товарів з посиланням на повний опис на сторінці з відомостями;
- вартість товару.

[1)] [2] Технології планування, розробки та тестування програм. URL: http://moodle.ipo.kpi.ua/moodle/mod/resource/view.php (Дата звернення 01.05.2020).

2 АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ГРАФІЧНОГО ІНТЕРФЕЙСУ

2.1 Опис мови HTML

HTML (Hypertext Markup Language − мова гіпертекстової розмітки) − це мова опису структури сторінок документів, яка дозволяє звичайний текст форматувати в абзаци, заголовки, списки та інші структури, створювати посилання на інші сторінки. Це текстова мова, в якій інструкції з форматування, що називаються тегами, вбудовані в розділи документа, які містять конкретну інформацію. Теги повідомляють браузерам, як форматувати і представляти інформацію на екрані. [3][1)]

Мова HTML дозволяє визначити структуру електронного документа з поліграфічним рівнем оформлення. Результуючий документ може містити різноманітні елементи: ілюстрації, аудіо і відео фрагменти. Мова HTML включає розвинені засоби для визначення кількох рівнів заголовків, шрифтових виділень, різних груп об’єктів та багато інших можливостей.

Важливим чинником, який вплинув на розвиток мови HTML, став її вибір за основу для гіпертекстової бази даних звичайного текстового файлу, який можна створювати у будь-якому текстовому редакторі на будь-якій апаратній платформі у середовищі будь-якої операційної системи.

Таким чином, гіпертекстова база даних у концепції WWW – це набір текстових файлів, розмічених мовою HTML, яка визначає форму представлення інформації (розмітка) і структуру зв’язків цих файлів (гіпертекстові посилання).

За основу моделі розмітки документів у HTML прийнята тегова модель. Тегова модель описує документ як сукупність контейнерів, кожен з яких починається і закінчується тегами. Тобто документ HTML є не чим іншим, як звичайним ASCII-файлом з доданими до нього керуючими HTML-кодами (тегами).

[1)] [3] Розробка на HTML. URL: http://htmlbook.ru (дата звернення 05.05.2020).

Теги HTML-документів в основному є простими і зрозумілими для використання, оскільки вони створені за допомогою загальновживаних слів англійської мови, зрозумілих скорочень і позначень.

HTML-тег складається з імені, за яким може слідувати необов'язковий список атрибутів тегу. Текст тегу вміщується у кутові дужки (<I>). Найпростіший варіант тегу – ім'я, вміщене у кутові дужки, наприклад, <HEAD>. Для більш складних тегів характерна наявність різних атрибутів, які можуть мати конкретні значення, визначені для видозмінення функцій тегу.

Атрибути тегу слідують за ім'ям і відділяються один від одного одним або кількома пропусками. Порядок запису атрибутів у тегу не має значення. Значення атрибута слідує за знаком рівняння, який стоїть після імені атрибута. Якщо значення атрибута – одне слово або число, його можна вказати безпосередньо після знаку рівняння, не виділяючи додатково. Решту значень необхідно вміщувати у одинарні або подвійні лапки, особливо якщо вони містять декілька розділених пропусками слів.

Найчастіше HTML-теги складаються з початкового і кінцевого компонентів, між якими розміщуються текст та інші елементи документа. Ім'я кінцевого тега ідентичне імені початкового тегу, але перед ім'ям ставиться коса риска (/) (наприклад, для тегу заголовка <TITLE> закриваючою парою буде </TITLE>). Кінцеві теги не містять атрибутів.

При використанні вкладених тегів їх слід закривати, починаючи з останнього і рухаючись до першого.

Деякі HTML-теги не мають кінцевого компонента, оскільки є автономними елементами. Наприклад, тег зображення <IMG>, призначений для вставки зображення у документ, не має кінцевого компонента.

Для створення HTML-документа можна застосувати редактор ASCII (зокрема, Блокнот системи Windows). Такі редактори дозволяють вводити HTML-теги, не додаючи до створеного нічого додатково. Створення документа

у такому редакторі дозволяє паралельно переглядати результат у програмі-браузері. Інший тип редакторів – візуальні HTML-редактори, наприклад, Microsoft FrontPage. Їх інтерфейс побудований за тим же принципом, що і інтерфейс текстового процесора, такого, як, наприклад, Word. Для роботи з візуальним редактором можна взагалі не володіти мовою HTML. Недоліком візуальних редакторів є те, що розмір створюваного ними HTML-документа у декілька разів більший, ніж документа, створеного звичайним Блокнотом системи Windows. В умовах низької пропускної здатності вітчизняних мереж цей недолік, який стосується швидкості завантаження сторінки (і, відповідно, вартості часу, який на це витрачається), є досить суттєвим недоліком (файл .htm, створений у WORD, в 4 − 9 разів більший, ніж файл аналогічного змісту, створений програмою Блокнот).

Детальний опис властивостей та значень CSS-правил можна знайти на сайті CSS довідник Основним поняттям CSS є стиль − набір правил оформлення і форматування, який може бути застосований до різних елементів сторінки.

У стандартному HTML для присвоєння елементу певних властивостей (таких, як колір, розмір, розташування на сторінці і т. д.) доводилося кожного разу описувати ці властивості, навіть якщо на одній сторінці повинно розташовуватися десятки однакових елементів.

CSS діє іншим, зручнішим і економнішим способом. Для присвоєння якому-небудь елементу певних характеристик потрібно один раз описати цей елемент і визначити цей опис як стиль, а надалі просто вказувати, що елемент, який необхідно оформити відповідним чином, повинен прийняти властивості описаного стилю.

Застосувати таблицю стилів до HTML-документу можна трьома способами:

– застосувати зовнішні стилі (у вигляді окремого текстового. css-файлу) за допомогою елементу link;

– вбудувати стилі безпосередньо в HTML-документ (у вигляді блоку css-тексту) за допомогою елемента style;
– застосувати inline-стиль, тобто призначити стиль конкретному HTML-елементу безпосередньо в документі, за допомогою HTML-атрибуту style.

На рис. 1 можна побачити один з типових HTML шаблонів.

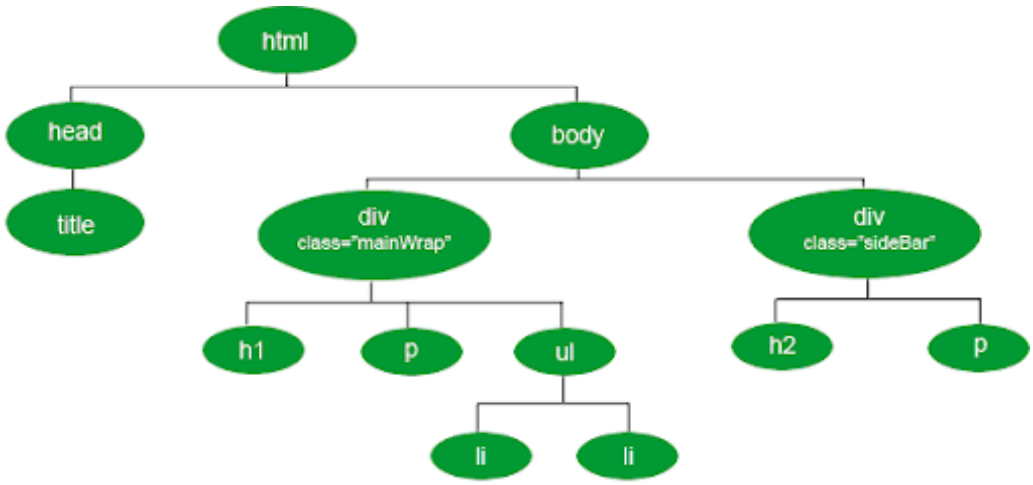


Рисунок 1 – Типовий HTML шаблон

2.2 Опис стилю CSS

CSS − зовнішні стилі (External Style Sheets). Застосовуються за допомогою елемента link, який повинен розташовуватися тільки всередині елемента head. [4] [1)]

Зустрівши в HTML-документі цей тег, браузер завантажить з сайту CSS-файл (у нашому випадку це mystyle.css) і застосує до документа стилі, що містяться в ньому. Файл не повинен містити нічого, крім CSS-інструкцій. Зовнішній файл зі стилями зручний тим, що одні й ті ж стилі можна

[1)] [4] Інформаційний ресурс з розробки CSS. URL: https://itchief.ru/ (дата звернення 12.05.2020).

застосовувати до множини документів на сайті − в кожному з них достатньо лише вписати один рядок з елементом link.

Таблиці стилів документа (document style sheets) називаються так тому, що розташовуються безпосередньо в HTML-документі і застосовуються лише до нього. Іноді називаються embedded style sheet (вбудований стиль). CSS-стилі та коментарі розташовуються між відкриваючим і закриваючим тегами елемента style:

- сам тег style (на відміну від link) може знаходитися в будь-якій частині документа, але звичайно його розміщують всередині елемента head;
- стилі, що підставляються в рядок (inline styles).

Іноді потрібно призначити стиль окремого елементу на сторінці, не застосовуючи зовнішніх стилів і елемента style. Типовий випадок − елемент зустрічається один раз в документі або на сайті, але вимагає особливого оформлення. Скористаємося атрибутом style (саме атрибутом елементів, а не елементом!)

Атрибут style є у всіх HTML-елементів, що розташовуються в елементі body. Усередині атрибуту style можна написати кілька CSS оголошень, розділених крапкою з комою, фігурні дужки не використовуються.

2.2.1 Структура CSS-правил

Всі CSS-правила складаються з селектора і блоку оголошень (укладеного у фігурні дужки). Усередині фігурних дужок блоку оголошень може знаходитися одне або кілька оголошень, розділених крапкою з комою. Оголошення − це рядок, складений з css-властивості і його значення. Вигляд css-правила

Кожне правило починається з селектора (покажчика), що вказує на ті html-елементи, до яких ми збираємося застосувати css-правило. У блоці оголошень встановлюються правила оформлення обраних нами елементів,

перевизначений їх властивості − розміри, колір, бордюри, поля, положення на екрані і т.д. Використовуються три основних види селекторів: HTML селектори, селектори класу, ID селектори (або ідентифікатори)

2.2.1.1 Огляд HTML селектору

Це найпростіший випадок − в якості селектора ми використовуємо ім'я html-елемента, який хочемо змінити. Наприклад, для тега селектором буде strong. Відповідно, для тега <h1> селектором буде h1, і так далі.

2.2.1.2 Селектор класу

Клас − це деяке ім'я, яке ми можемо застосувати до будь-яких HTML-тегів, щоб згодом посилатися на них по імені класу. Як ім'я класу можна використовувати практично будь-який рядок. Зручність таких селекторів в тому, що можна присвоїти одне ім'я класу множині html-тегів у документі і потім управляти їх зовнішнім виглядом, звертаючись до них по імені класу.

2.2.1.3 Огляд селектора ID

Будь-який ідентифікатор (ID) − це якесь ім'я, яке ви, так само, як і у випадку з класами, можете застосувати до будь-якого HTML-тегу. Основна відмінність − ID повинен бути унікальним в рамках html-документа:

Дуже поширений випадок − застосувати один набір правил до декількох різних селекторів. Це робиться елементарно − достатньо перерахувати селектори через кому.

2.2.2 Розробка інтерфейсу

Титульна сторінка (головна) будь-якого сайту повинна максимально інформативно і в стислому об'ємі відображати необхідну користувачеві інформацію про сайт. На головній сторінці необхідно помістити логотип веб-

сайту, основне меню сайту (для навігації по його структурі), форму аутентифікації (входу зареєстрованих користувачів), реєстраційне посилання (реєстрація нових клієнтів).

2.2.3 Препроцесор SASS

Sass (англ. Syntactically Awesome Stylesheets) – скриптова метамова, яка інтерпретується в каскадні таблиці стилів (CSS). Спроектована Гемптоном Кетліном та розроблена Наталі Вейзенбаум. Sass призначений для підвищення рівня абстракції коду та спрощення файлів CSS.

Мова Sass має два синтаксиси:

- Sass (оригінальний) – відрізняється відсутністю фігурних дужок, в ньому вкладені елементи реалізовані за допомогою відступів, а правила відокремлюються переведенням рядка;
- Scss (новий) – використовує фігурні дужки (подібно до CSS);
- файли sass-синтаксису мають розширення .sass, scss-синтаксису – .scss.

Sass розширює CSS, надаючи кілька механізмів, доступних в більш традиційних мовах програмування, зокрема об'єктно-орієнтованих мовах, але недоступних для CSS. Інтерпретатор Sass транслює SassScript у блоки правил CSS.

2.3 Опис мови програмування JavaScript

JavaScript – це мова програмування, що дозволяє зробити веб-сторінку інтерактивною, тобто такою що реагує на дії користувача.[5][1)]

JavaScript − об'єктно-орієнтована скриптова мова програмування і є діалектом мови ECMAScript.

JavaScript зазвичай використовується як вбудована мова для програмного

[1)] [5] Скриптова мова програмування JavaScript. URL: https:/learn.javascript.ru (дата звернення 06.05.2020).

доступу до об'єктів додатків. Найбільш широке застосування знаходить у браузерах як мова сценаріїв для надання інтерактивності веб-сторінкам.

Послідовність інструкцій (що називається програмою, скриптом або сценарієм) виконується інтерпретатором, вбудованим в звичайний веб-браузер. Іншими словами, код програми вбудовується в HTML − документ і виконується на боці клієнта. Для виконання програми не потрібно навіть перезавантажувати веб-сторінку, всі програми виконуються в відповідь на будь-яку подію. Наприклад, перед відправленням даних форми можна перевірити їх на допустимі значення і, якщо значення не відповідають очікуваним, заборонити відправлення даних.

Основні архітектурні риси:

– динамічна типізація;
– автоматичне керування пам'яттю;
– прототипне програмування;
– функції як об'єкти першого класу.

На JavaScript вплинули багато мов, при розробці була мета зробити мову схожою на Java, але при цьому легкою для використання не програмістами. JavaScript має низку властивостей об'єктно-орієнтованої мови, але реалізоване в мові прототипування обумовлює відмінності в роботі з об'єктами в порівнянні з традиційними об'єктно-орієнтованими мовами. Крім того, JavaScript має ряд властивостей, властивих функціональним мовам, − функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання − що додає мові додаткову гнучкість.

Незважаючи на схожий з Сі синтаксис, JavaScript у порівнянні з мовою Сі має корінні відмінності:

– об'єкти, з можливістю інтроспекції;
– функції як об'єкти першого класу;
– автоматичне приведення типів;

– автоматичне прибирання сміття;
– анонімні функції.

У мові відсутні такі корисні речі, як:

– модульна система − JavaScript не надає можливості управляти залежностями та ізоляцією областей видимості;
– стандартна бібліотека − зокрема, відсутній інтерфейс програмування додатків по роботі з файловою системою, управління потоками вводу/виводу, базових типів для бінарних даних;
– стандартні інтерфейси до веб-серверів та баз даних;
– система управління пакетами, яка б відстежувала залежності і автоматично встановлювала їх.

Синтаксис мови JavaScript дуже нагадує синтаксис Сі і Java, семантично ж мова набагато ближче до Self, Smalltalk або навіть Ліспу.

В JavaScript :

– всі ідентифікатори Реєстрозалежні;
– в назвах змінних можна використовувати літери, підкреслення, символ долара, арабські цифри;
– назви змінних не можуть починатися з цифри;
– для оформлення однорядкових коментарів використовуються //, багаторядкові і внутрішньорядкові коментарі починаються з/* і закінчуються */.

Структурно JavaScript можна представити у вигляді об'єднання трьох частин, що чітко різняться одна від одної:

– ядро (ECMAScript);
– об'єктна модель браузера (Browser Object Model або BOM);
– об'єктна модель документа (Document Object Model або DOM).

Об'єктну модель документа іноді розглядають як окрему від JavaScript сутність, що узгоджується з визначенням DOM як незалежного від мови

інтерфейсу документа.

ECMAScript не є браузерною мовою і насправді в ній не визначаються методи введення і виведення інформації. Це скоріше основа для побудови скриптових мов. Специфікація ECMAScript описує типи даних, інструкції, ключові і зарезервовані слова, оператори, об'єкти, регулярні вирази, не обмежуючи авторів похідних мов від розширення їх новими складовими.

Об'єктна модель браузера − браузероспеціфічна частина мови, яка являється прошарком між ядром і об'єктною моделлю документа. Основне призначення об'єктної моделі браузера − керування вікнами браузера і забезпечення їх взаємодії. Кожне з вікон браузера представляється об'єктом window, центральним об'єктом BOM. Об'єктна модель браузера на даний момент не стандартизована, проте специфікація знаходиться в розробці WHATWG та W3C. Крім управління вікнами, в рамках об'єктної моделі браузера, браузерами зазвичай забезпечується підтримка наступних сутностей:

– керування фреймами;
– підтримка затримки у виконанні коду і зациклювання з затримкою;
– системні діалоги;
– управління адресою відкритої сторінки;
– управління інформацією про браузер;
– управління інформацією про параметри монітора;
– обмежене керування історією перегляду сторінок;
– підтримка роботи з HTTP cookie.

Об'єктна модель документа − інтерфейс програмування додатків для HTML і XML-документів. Згідно DOM документом можна поставити у відповідність дерево об'єктів, які мають ряд властивостей, які дозволяють робити з ним різні маніпуляції:

– отримання вузлів;
– зміна вузлів;

– зміна зв'язків між вузлами;

– видалення вузлів.

Область застосування:

– використання на веб-сторінках;

– розташування всередині сторінки.

Щоб додати JavaScript-код на сторінку, можна використовувати теги.

Розташування всередині тега. Специфікація HTML описує набір атрибутів, використовуваних для завдання обробників подій.

Приклад використання:

```
<<a href="delete.php" onclick="return confirm ('Ви впевнені?');>"> Вилучити </a>
```

Відокремлення від розмітки у наведеному прикладі при натисненні на посилання функція confirm ('Ви впевнені?') викликає модальне вікно з написом «Ви впевнені?», a return false; блокує перехід за посиланням. Зрозуміло, цей код буде працювати тільки якщо в браузері є і включена підтримка JavaScript, інакше перехід за посиланням відбудеться без попередження.

Використання коду JavaScript в контексті розмітки сторінки в рамках ненав'язливого JavaScript розцінюється як погана практика. Аналогом (за умови надання посилання ідентифікатором alertLink).

Винесення в окремий файл. Є і третя можливість підключення JavaScript − написати скрипт в окремому файлі, а потім підключити його за допомогою конструкції

Тег script, що широко використовується для підключення до сторінки JavaScript, має декілька атрибутів.

Обов'язковий атрибут type для вказівки MIME-типу вмісту.

Проте, згідно специфікації HTML 4.01 в якості значення type повинно

бути зазначено застаріле "text/javascript". Тому що JavaScript є мовою програмування за замовчуванням у всіх браузерах, починаючи з Netscape 2, Дуглас Крокфорд дотримується думки про недоцільність використання атрибута type, рекомендуючи вказувати його в XHTML.

Необов'язковий атрибут src, що приймає в якості значення адресу до файлу зі скриптом.

Необов'язковий атрибут charset, використовуваний разом з src для вказівки кодування, що використовується для зовнішнього файлу.

Необов'язковий атрибут defer, що використовується для того, щоб показати, що скрипт не генерує ніякого вмісту (що означає, зокрема, те, що в цьому скрипті відсутній виклик document.write ()).

При цьому атрибут language language="JavaScript"), незважаючи на його активне використання (у 2008 році цей атрибут був найбільш часто використовуваний у тегах <script>), відноситься до нерекомендованих (deprecated), відсутній в DTD, тому вважається некоректним.

Букмарклети JavaScript використовується для створення невеликих програм, що розміщуються в закладках браузера. При цьому використовуються URL-адреси зі специфікатором javascript.

Користувацькі скрипти в браузері − це програми, написані на JavaScript, що виконуються в браузері користувача при завантаженні сторінки. Вони дозволяють автоматично заповнювати форми, переформовувати сторінки, приховувати небажаний вміст та вбудовувати бажаний для відображення вміст, змінювати поведінку клієнтської частини веб-додатків, додавати елементи керування на сторінку і т.д.

Для управління призначеними для користувача скриптами в Mozilla Firefox використовується розширення Greasemonkey; Opera надає засоби підтримки користувальницьких скриптів і можливості для виконання ряду скриптів Greasemonkey. Деякі скрипти Greasemonkey можуть виконуватися в

Google Chrome при використанні Greasemetal.

2.4 Опис технології AJAX

AJAX – це абревіатура, що в розгорнутому вигляді – Asynchronous JavaScript and XML. Говорячи простою мовою, це певна технологія, що дозволяє підвантажувати ряд даних на веб-сторінці без перевантаження самої сторінки, що, відповідно, зменшить кількість запитів до сервера, а також об’єм підвантажуваних даних.

При роботі з веб-сторінкою на етапі підвантаження із використанням вище згаданої технології викликається функція на стороні клієнта, що перенаправляє запит на сторону сервера. На сервері запускається скрипт, який в свою чергу поверне вам відповіть у вигляді даних. В основному технологія використовується для підвантаження окремих даних, відправки даних форм, а саме авторизація, добавлення коментарів чи відправки повідомлень. На рис. 2 можна побачити схему роботи технології AJAX.[5][1)]

AJAX – це не самостійна технологія, а швидше концепція використання декількох суміжних технологій. AJAX-підхід до розробки, який призначений для користувачів інтерфейсів, комбінує кілька основних методів і прийомів:

- Використання DHTML для динамічної зміни змісту сторінки;
- Використання XMLHttpRequest для звернення до сервера «на льоту», не перезавантажуючи всю сторінку повністю;
- Альтернативний метод – динамічне підвантаження коду JavaScript в тег <SCRIPT> з використанням DOM, що здійснюється із використанням формату JSON);
- динамічне створення дочірніх фреймів.

[1)] [5] Скриптова мова програмування JavaScript. URL: https:/learn.javascript.ru (дата звернення 06.05.2020).

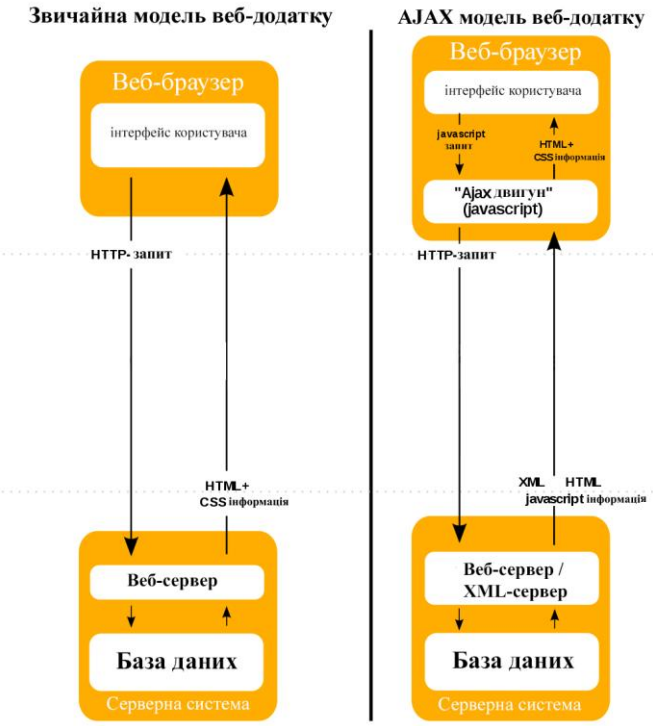


Рисунок 2 – Схема роботи технології AJAX

2.5 Опис бібліотеки JQUERY

З розвитком технологій функціональність веб-сторінок постійно зростає й наближається до функціональності настільних прикладних програм. Ця зростаюча функціональність реалізується за допомогою JavaScript. І зовсім не обов'язково особисто прописувати всі необхідні ефекти, якщо для цього вже написано кілька десятків або навіть сотень бібліотек, що дозволяють реалізовувати ці ефекти. І серед безлічі цих бібліотек по праву найбільш зручною й доступною для розуміння є бібліотека JQuery. [5][1)]

Давайте розберемося. JQuery-бібліотека Javascript, що фокусується на взаємодії JavaScript і HTML. Була опублікована на комп'ютерній конференції «Barcamp» у Нью-Йорку в 2006 році. У чому головна перевага JQuery? Вона

[1)] [5] Скриптова мова програмування JavaScript. URL: https:/learn.javascript.ru (дата звернення 06.05.2020).

закладена на рівні ядра – це вибір елементів об'єктної моделі документів. Крім того, завдяки наявності плагінів, базова функціональність JQuery може бути розширена. Для початку роботи з JQuery необхідно скачати саму бібліотеку з будь-якого доступного джерела. Далі її необхідно ініціалізувати. Після ініціалізації, для використання доступні всі можливості базового функціонала JQuery, серед яких:

- функції ядра;
- робота із селекторами;
- робота з атрибутами;
- обхід дерева DOM;
- маніпуляції елементами;
- робота з CSS-властивостями елементів;
- робота з подіями;
- візуальні ефекти;
- взаємодія з AJAX;
- утиліти.

Для маніпулювання потрібними елементами сторінки в Javascript є кілька способів знайти їх на сторінці серед безлічі інших об'єктів. Ці способи вимагають запам'ятовування великої кількості інформації, у той час як для пошуку елемента за допомогою JQuery необхідно лише пам'ятати ID елемента, з яким ви прагнете працювати. Код звернення в загальному випадку буде виглядати так: Ключовою функцією в JQuery є функція $() − вона тим чи іншим способом викликається всіма методами JQuery. Згідно із заявами вице-президента по розробці Скотта Гутрі (Майкрософт), бібліотека JQuery, стане основою ASP.Net AJAX Control Toolkit і буде поставлятися в складі Visual Studio. Розроблювач JQuery Джон Резиг заявив, що Нокіа теж використовує JQuery як частину своєї платформи для розробки Web Runtime, що базується на Webkit. Не відстають і російські компанії, наприклад, Яндекс уже давно активно

використовує JQuery у своїх додатках. Завдяки тому, що обсяг програмного коду JQuery менше, ніж обсяг стандартного коду Javascript, скорочуються часові витрати на розробку елементів веб-сторінки. Сам програмний код більш зрозумілий у порівнянні з Javascript. Наведемо приклад. Існує якась таблиця, непарні рядки якої пофарбовані відмінним від основного кольором. Припустимо, що існує також якась таблиця стилів CSS, у якій уже визначений клас (у прикладі, "odd") для такого стилю. Для реалізації цього ефекту необхідно наділити непарні рядки таблиці даним класом. Для створення простого AJAX-Запиту потрібно всього близько п'яти рядків коду, що значно спрощує використання цієї технології. [6][1)]

[1)] [6] Архітектура клієнт-сервер. URL: http://inter.ptngu.com/ (дата звернення 30.04.2020).

3 АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ БАЗИ ДАНИХ

3.1 Опис баз даних

База даних (БД) − це організована структура, яка призначена для зберігання, зміни та обробки взаємозалежної інформації, переважно великих обсягів. БД використовують для динамічних сайтів з великими обсягами (інтернет-магазин, портал, корпоративний сайт).[7][1)]

Бази даних для сайтів дають змогу зберігати інформацію, що виглядає як зв'язані між собою таблиці. Саме в БД зберігаються вся необхідна та корисна інформація для функціонування сайту (клієнтські дані, прайс-лист, список товарів). Щоб створити запит до бази даних часто використовують Structured Query Language. SQL дає змогу додавати, редагувати та видаляти інформацію, що міститься у таблицях. Під час програмування сайтів використовують різні системи управління БД. До основних СУБД, відносять:

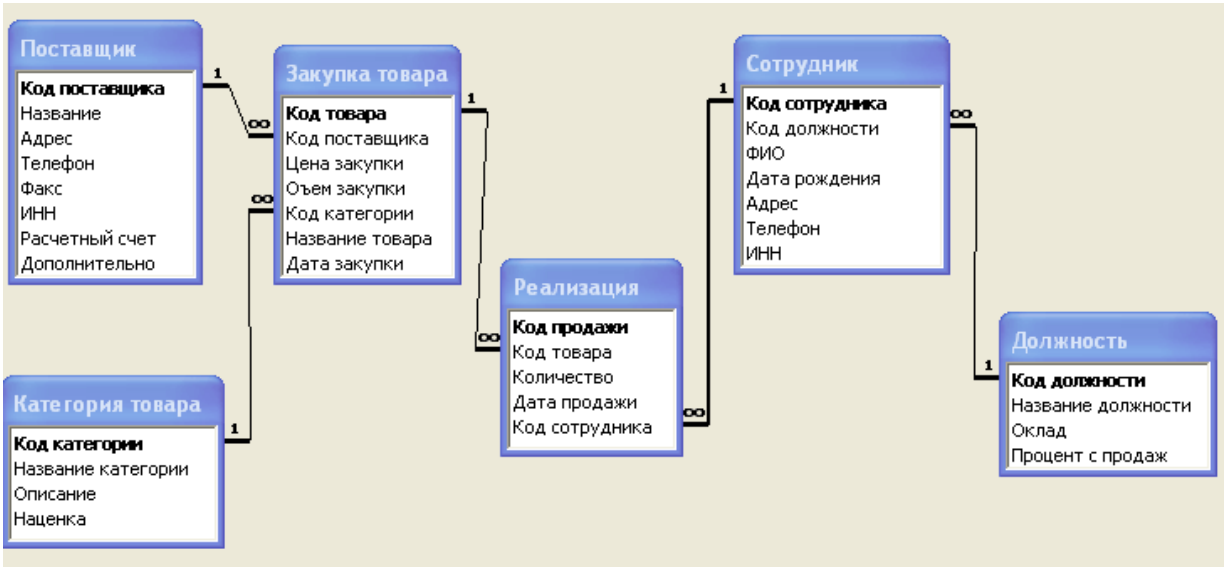


Рисунок 3 – Типова схема бази даних інтернет магазину

[1)] [7] Визначення база даних. URL: https://www.oracle.com/ru/database/what-is-database.html (дата звернення 01.05.2020).

– об'єктно-реляційна система управління базами даних Oracle Database;
– вільна система управління базами даних PostgreSQL;
– система керування базами даних Microsoft SQL Сервер;
– вільна система управління базами даних MySQL.

Такі системи управління відрізняються централізованою обробкою запитів, забезпечують надійність, доступність та безпеку БД. Найбільш популярною системою управління є MySQL, вона дає зручний доступ для управління БД та підтримує велику кількість таблиць різних типів. На рис. 3 можна побачити типову схему бази даних інтернет магазину.

3.2 Види баз даних

3.2.1 Прості структури даних

Перший і найпростіший спосіб зберігання даних − текстові файли. Метод застосовується і сьогодні для роботи з невеликими обсягами інформації. Для поділу полів використовується спеціальний символ: кома або крапка з комою в csv-файлах датасета, двокрапка або пробіл в * nix-подібних системах. [8][1)]

Наслідки:

– обмежений тип і рівень складності інформації, що зберігається;
– важко встановити зв'язки між компонентами даних;
– відсутність функцій паралелізму;
– практичні (для систем з невеликими вимогами до читання і запису);
– використовуються для зберігання конфігураційних даних;
– немає необхідності в сторонньому програмному забезпеченні.

[1)] [8] Інформаційний ресурс з розробки веб-сайтів. Види баз даних. Codenet.ru URL: https://codenet.ru/db/mysql (дата звернення 09.05.2020).

3.2.2 Огляд ієрархічних баз даних

На відміну від текстових таблиць, в наступному типі БД з'являються зв'язки між об'єктами. В ієрархічних базах даних кожен запис має одного «батька». Це створює деревоподібну структуру, в якій записи класифікуються за їхніми стосункам з ланцюжком батьківських записів.

Наслідки:

– інформація організована у вигляді дерева з відносинами «предок-нащадок»;
– кожен запис може мати не більше одного з батьків;
– зв'язки між записами виконані у вигляді фізичних покажчиків;
– неможливо реалізувати відносини «багатьох до багатьох».

3.2.3 Мережеві бази даних

Мережеві бази даних розширюють функціональність ієрархічних: записи можуть мати більше одного батька. А значить, можна моделювати складні відносини.

Наслідки:

– мережеві бази даних подаються не деревом, а загальним графом;
– обмежені тими ж шаблонами доступу, що ієрархічні БД.

3.2.4 Загальні відомості бази даних SQL

Реляційні бази даних − найстаріший тип і досі широко використовуваних БД загального призначення. Дані та зв'язки між даними організовані за допомогою таблиць. Кожен стовпець у таблиці має ім'я і тип. Кожен рядок представляє окремий запис або елемент даних в таблиці, який містить значення для кожного з стовпців.

Наслідки:

– поле в таблиці, зване зовнішнім ключем, може містити посилання на стовпці в інших таблицях, що дозволяє їх з'єднувати;
– високоорганізована структура і гнучкість робить реляційні БД потужними і такими, що адаптуються до різних типів даних;
– для доступу до даних використовується мова структурованих запитів (SQL);
– надійний вибір для багатьох додатків.

3.2.5 Характеритстика бази даних «ключ-значення»

У базах даних «ключ-значення» для зберігання інформації ви надаєте ключ і об'єкт даних, який потрібно зберегти. Наприклад, JSON-об'єкт, зображення або текст. Щоб зробити запит, відправляєте ключ і отримуєте blob-об'єкт .

Наслідки:

– сховища забезпечують швидкий доступ з незначними витратами;
– часто зберігають дані конфігурацій і інформацію про стан даних, представлених словниками або хешем;
– немає жорсткої схеми відносини між даними, тому в таких БД часто зберігають одночасно різні типи даних;
– розробник відповідає за визначення схеми іменування ключів і за те, щоб значення мало відповідний тип / формат.

3.2.6 Огляд документних баз даних

Документні бази даних (також документоорієнтовані БД або сховища документів), спільно використовують базову семантику доступу і пошуку сховищ ключів і значень. Такі БД також використовують ключ для унікальної ідентифікації даних. Різниця між сховищами «ключ-значення» і документними БД полягає в тому, що замість зберігання blob-об'єктів, документоорієнтовані

бази зберігають дані в структурованих форматах − JSON, BSON або XML.

Наслідки:

– база даних не виділяє окремий формат або схему;

– кожен документ може мати свою внутрішню структуру;

– документні БД є хорошим вибором для швидкої розробки;

– в будь-який момент можна змінювати властивості даних, не змінюючи структуру або самі дані.

3.2.7 Аналіз графовмх баз даних

Замість зіставлення зв'язків з таблицями і зовнішніми ключами, графові бази даних встановлюють зв'язки, використовуючи вузли, ребра і властивості.

Графові бази представляють дані у вигляді окремих вузлів, які можуть мати будь-яку кількість пов'язаних з ними властивостей.

Наслідки:

– виглядають аналогічно мережевим;

– фокусуються на зв'язках між елементами;

– явно відображають зв'язки між типами даних;

– не вимагають покрокового обходу для переміщення між елементами;

– немає обмежень в типах зв'язків.

3.3 Аналіз файлів JSON

Об'єктний запис JavaScript (JSON, JavaScript Object Notation) − це формат подання даних. JSON може подавати числа, тип boolean, рядки, null, масиви (впорядковані послідовності значень), а також об'єкти (містять пари ключ-значення), які й собі можуть містити значення всіх наведених типів (зокрема, інші об'єкти чи масиви). Складніші дані, як-от функції, регулярні вирази чи дати, в JSON неможливо подати як є − їх доведеться попередньо перетворювати на дані підтримуваних типів. Початково об'єкти Date перетворювано на рядок,

що містить запис дати у форматі ISO, тож не всі відомості буде загублено. Втім, якщо є потреба зберігати у JSON деякі додаткові типи даних, доведеться виконати належні перетворення до серіалізування та після десеріалізування.

JSON – формат обміну даними, що використовується як альтернатива XML та є на третину компактнішим від нього. Широко використовується в складних програмах, які часто обмінюються даними між браузером та веб-сервером (технологія AJAX). Підтримує чотири примітивних типи даних (String, Number, Boolean, Null) та два комплексних (Object – фігурні дужки, Array – квадратні дужки). На рис. 4 можна побачити типову JSON базу даних. [8][1)]

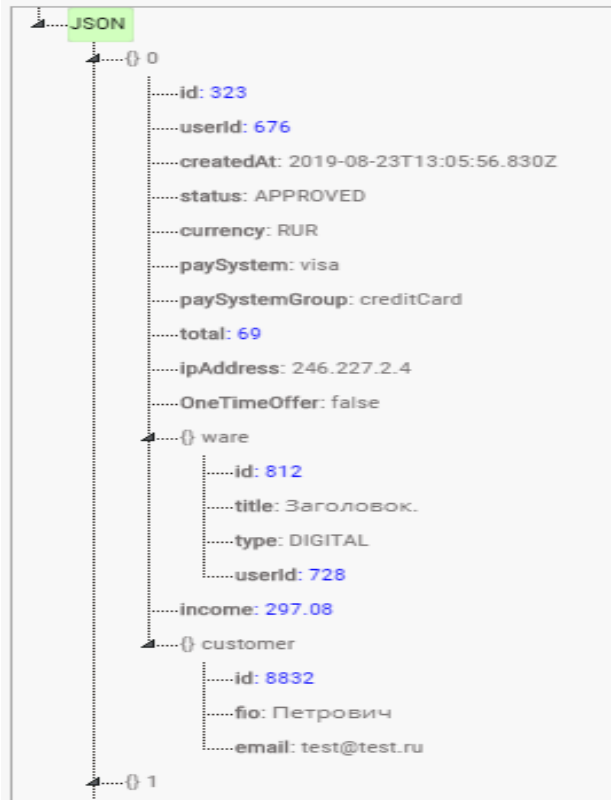


Рисунок 4 – Типова JSON база даних

[1)] [8] Інформаційний ресурс з JSON файлів. URL https://www.json.org/json-ru.html (дата звернення 09.05.2020).

4 ОПИС ІНТЕРНЕТ МАГАЗИНУ "ZYCSEL STORE"

4.1 Затвердження технічного завдання на розробку сайта

Перед розробником було поставлено завдання: розробити структурну схему проекту веб-сайту для використання в середовищі Internet. Веб-сайт, що розробляється також, повинен володіти наступними особливостями:

–гнучкістю, зручною для адміністраторів системою управління структурою;

–веб-сайт повинен підтримувати використання графічних вставок, анімації, які повинні підсилювати емоційно-ціннісний компонент змісту, формувати мотивацію.

Проте головним завданням проектування було створення структури, на яку виводяться товари одягу, та форми взаємодії з додатками. [9][1)]

4.2 Визначення структурної схеми сайту

Структура веб-сайту представлена у вигляді клієнтської частини та бази даних створеної с JSON файлів. Клієнтська частина доступна усім відвідувачам сайту. Вона забезпечує доступ до товарів, останніх новин, замовлення послуг. Сайт складається з п’яти сторінок. Вони пов’язані між собою посиланнями та пунктами головного меню. Головне меню на кожній сторінці та має незмінну структуру. Додаткове меню відсутнє. У середині сторінки знаходяться Пункти меню:

− "Верхній одяг";

− "Худі та світшоти";

− "Сорочки та футболки";

− "Штани та шорти".

[1)] [9] Технології планування, розробки та тестування програм. URL: http://moodle.ipo.kpi.ua/moodle/mod/resource/view.php (дата звернення 01.05.2020).

Зверху знаходиться хедер або шапка з контактною інформацією, посиланнями на сайти-партнери через які можливо замовити товари одягу, та полями пошуку товару за іменем та розміром.

Знизу знаходиться футер з контактною інформацією та посиланнями на сайти-партнери. Схему елементів веб-сторінок можна побачити на рис. 5.

Рисунок 5 – Схема веб-сайту.

На рис. 6 можна побачити структуру веб-сайту з усіма елементами верстки, JavaScript файлом з функціями, та бібліотекою JQuery, створену за допомогою функцій браузеру Google Chrome.

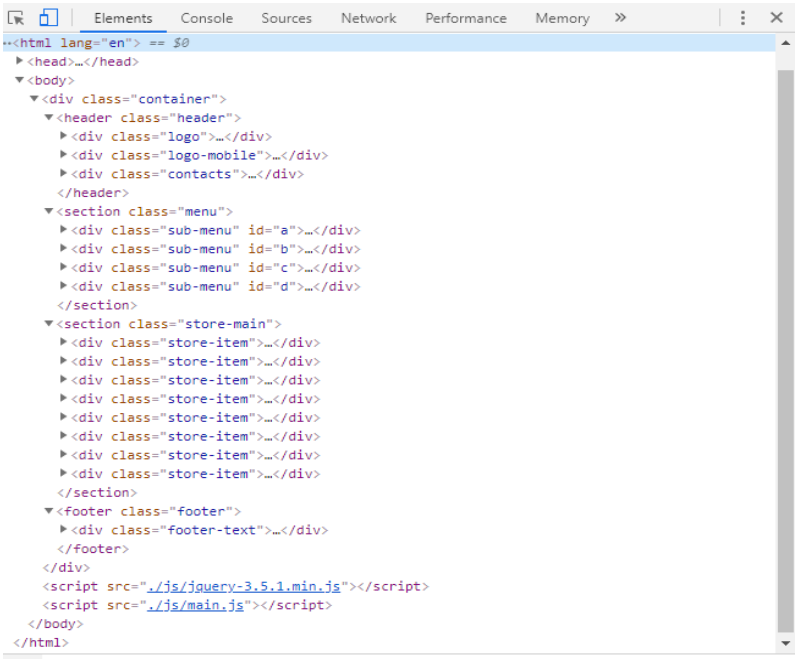


Рисунок 6 – Структура веб-сайту у браузері Google Chrome

На даний момент база даних веб-сервісу “Zycsel_Store” представлена у вигляді сукупності JSON файлів. Подібний варіант баз даних не має графічної оболонки, але як і усі бази даних має свою ієрархію та спосіб зберігання. На рис. 7 можна побачити ієрархію JSON даних, а на рис. 8 директорію у якій зберігаються JSON файли.

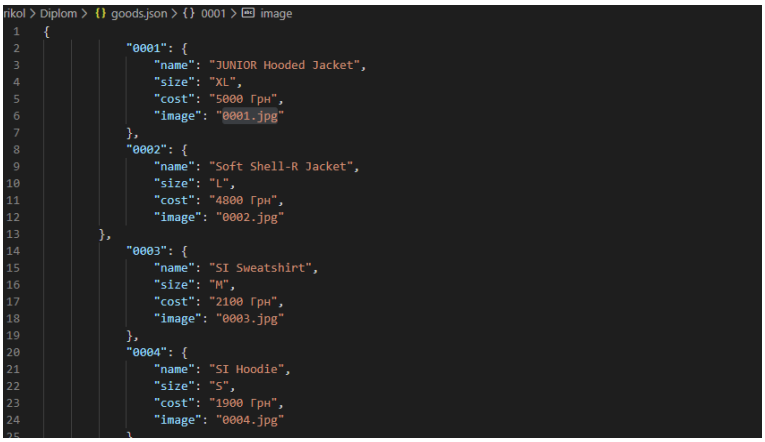


Рисунок 7 − Приклад коду JSON бази даних

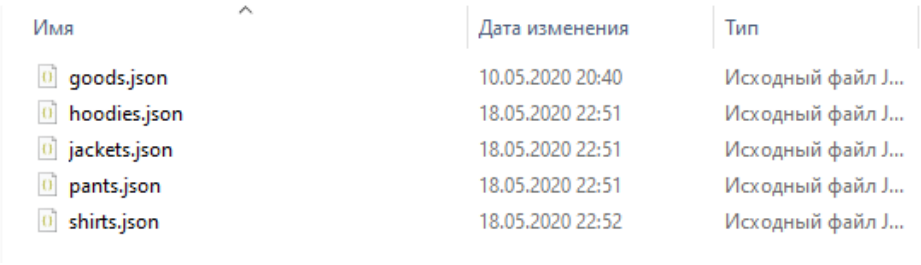


Рисунок 8 − Директорія JSON файлів

4.3 Створення графічних елементів системи

При переході за адресою на сторінці відобразиться головна сторінка сайту. Головна сторінка (Home page або main page) − початкова сторінка веб-сайту, яка, зазвичай, надає відомості про тематику веб-сайту та матеріали, які можна побачити на подальших сторінках (дозволяє переглянути зміст веб-сайту). Як правило, посилання робляться саме на домашню (головну) сторінку веб-сайту. На рис. 9 представлена головна сторінка сайту.[10][1)]

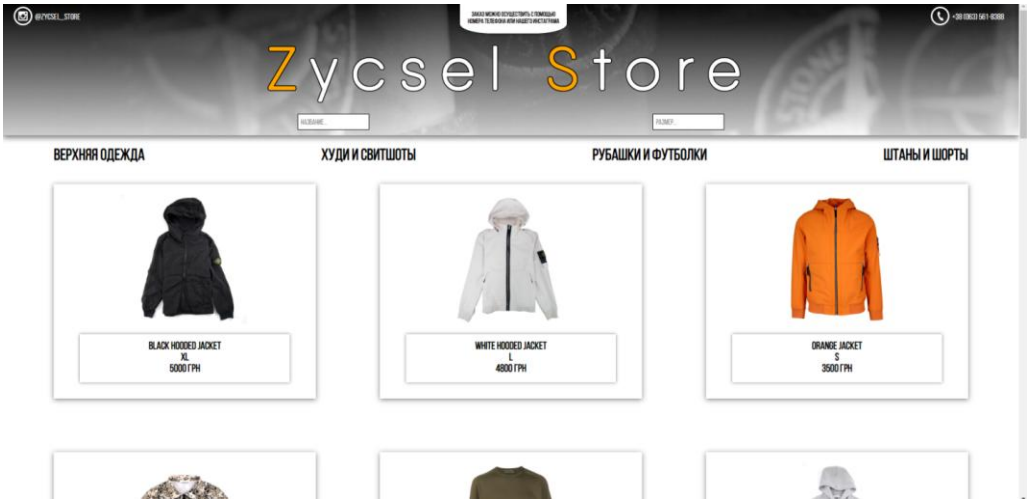


Рисунок 9 − Головна сторінка сайту

[1)] [10] Використання веб-ресурсів для покращення візуального сприйняття Інформації. URL: http://inmad.vntu.edu.ua/portal/index.php (дата звернення 28.04.2020).

Веб-сайт дає змогу переглядати інформацію про товари магазину. Обравши один з пунктів меню користувач побачить повний список товарів на сайті. На рис. 10 знаходиться пункт меню “Верхняя одежда”.

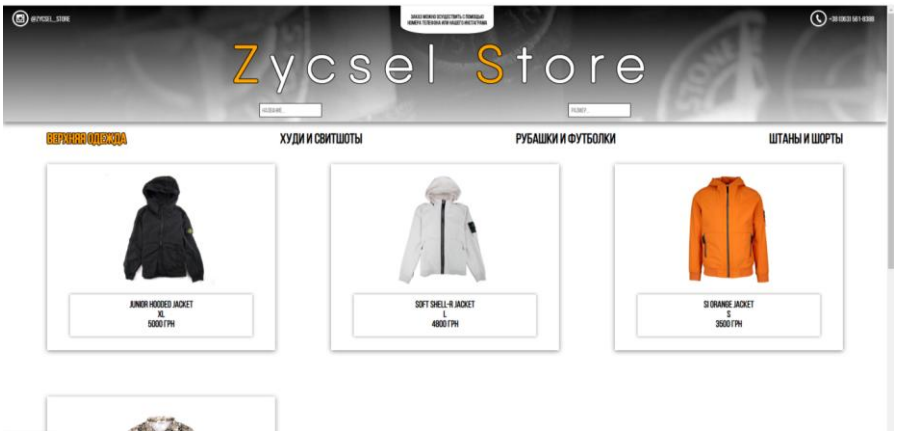


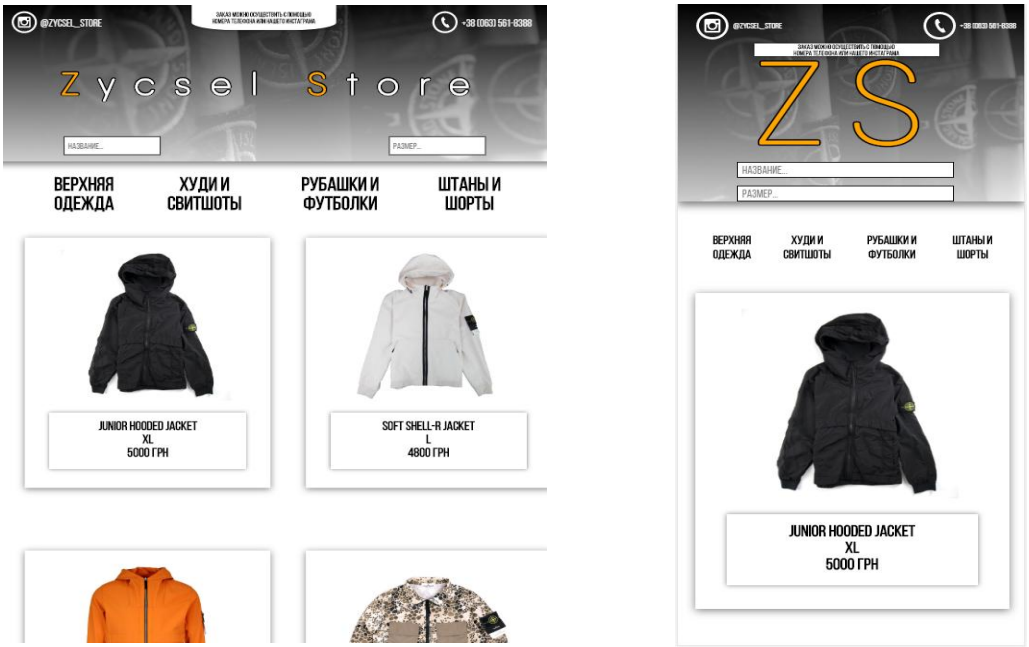
Рисунок 10 − Сторінка магазину “Верхняя одежда”

Обравши один із товарів, користувач має змогу замовити його за допомогою контактів які знаходяться у верхній частині веб-сайту.

Графічна частина сайту оформлена за принципами адаптивної верстки, та відповідає усім стандартам розмірів пристроїв для взаємодії з веб-ресурсом. На рис. 11 стандарти пристроїв.

Рисунок 11 − Стандарти розмірів пристроїв у пікселах

На рис. 12 (а,б) можна побачити адаптивний дизайн для зручного користування сайтом на мобільних пристроях та планшетах.



а) б)

Рисунок 12 − Адаптивний дизайн сайту: а) − для планшетних пристроїв; б) − для мобільних пристроїв

Кожнен товар генерується у індивідуальній картці, яку можна побачити на рис. 13.

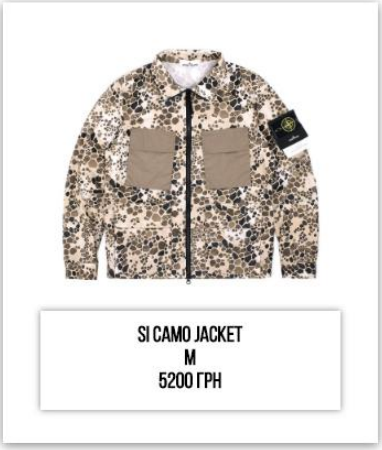


Рисунок 13 – Картка товару на веб-сайті

На картці можна побачити зображення товару, назву, розмір та ціну у гривнях.

Також на сайті присутні анімаційні елементи для приємного користування зі сторони клієнта. Анімаційні елементи продемонстровані на рисунках 14 (а,б).

а) б)

Рисунок 14 – Елемент: а) – без анімації; б) – з анімацією при наведенні

На рис. 15(а) можна побачити поле пошуку за іменем а на рис. 15(б) – активне поле пошуку за розміром.

а) б)

Рисунок 15 – Поле пошуку а) – за іменем; б) – активне поле за розміром

4.4 Розробка програмного коду та інших елементів сайту

На сьогодняшний день, веб-сайт не може працювати без додаткової мови програмування. Для виготовлення даного програмного продукту я використовував мову програмування JavaScript. К кожній сторінці сайту прив’язан файл с розширенням “.js”. У даних файлах сберігаються функції за допомогою яких на веб-сторінці здійснюються додаткові маніпуляції.

На рис. 16 можна побачити приклад коду Javascript файлу, який

використовує бібліотеку JQuery для виведення товарів на сторінку за допомогою JSON файлів які взаємодіють с директорією goods, у якій знаходяться зображення товару, зображена на рис. 17.

Рисунок 16 – Функція Init() для виведення товарів на сторінку сайту

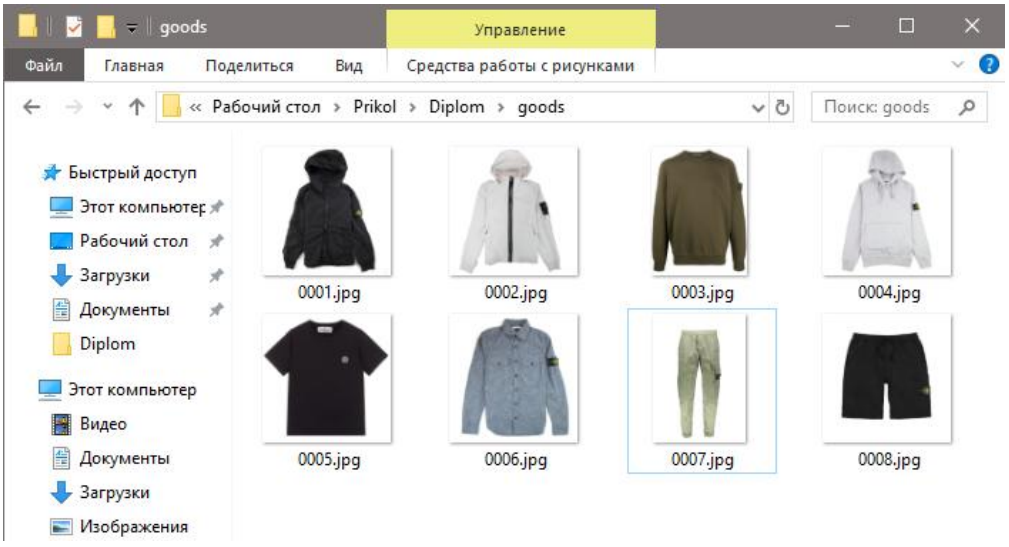


Рисунок 17 – Директорія goods з зображеннями товарів

Функція Init() користується директивою get.JSON для взяємодії с JSON

файлами з яких получає змінну data. У змінні дата зберігається значення типу String, яке береться за допомогою get.JSON. За допомогою змінної out яку повертає функція goodsOut ми виводимо ці значення у HTML клас (“.store-main”) генеруя у ньому картку з товаром, у якій знаходяться зображення, наіменування, розмір та ціна товару. Елементи функції можна побачити на рис. 18 представлена директива get.JSON та рис. 19 зображен цикл виводу на веб-сторінку.

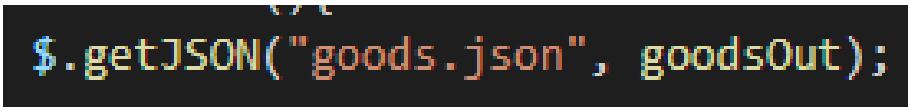


Рисунок 18 – Директива get.JSON

```
function goodsOut(data){
    console.log(data);
    var out='';
    for (var key in data) {
        out +='<div class="store-item">';
        out +=`<img class="item-image" src="./goods/${data[key].image}" alt="">`;
        out +='<div class="item-desctiption">';
        out +=`<span class="name">${data[key].name}</span>`;
        out +='<br>';
        out +=`<span class="size">${data[key].size}</span>`;
        out +='<br>';
        out +=`<span class="cost">${data[key].cost}</span>`;
        out +='</div>';
        out +='</div>';
```

Рисунок 19 – Функція goodsOut

Кожній сторінці сайту належать свої особисті JavaScript та JSON файли. Для JavaScript файлів створена особиста директорія “js”. Кожен JSON файл відповідає за особистий тип товару: верхній одяг, худі та світшоти, сорочки та футболки, штани та шорти. JSON файли мають бути в корні каталогу проекту для взаємодії з ними. На рис. 20 можна побачити директорію js, а на рис. 21 –

директорію з JSON файлами.

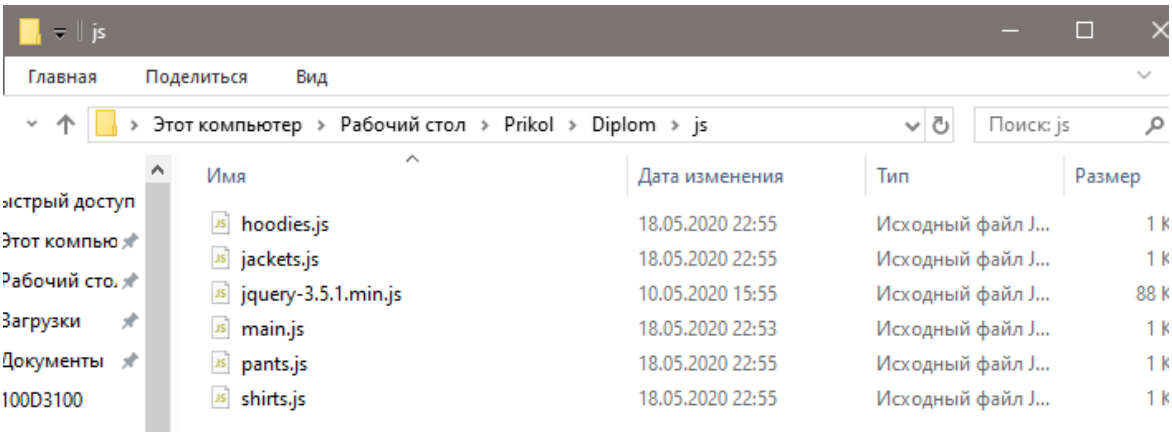


Рисунок 20 – Директорію js файлів

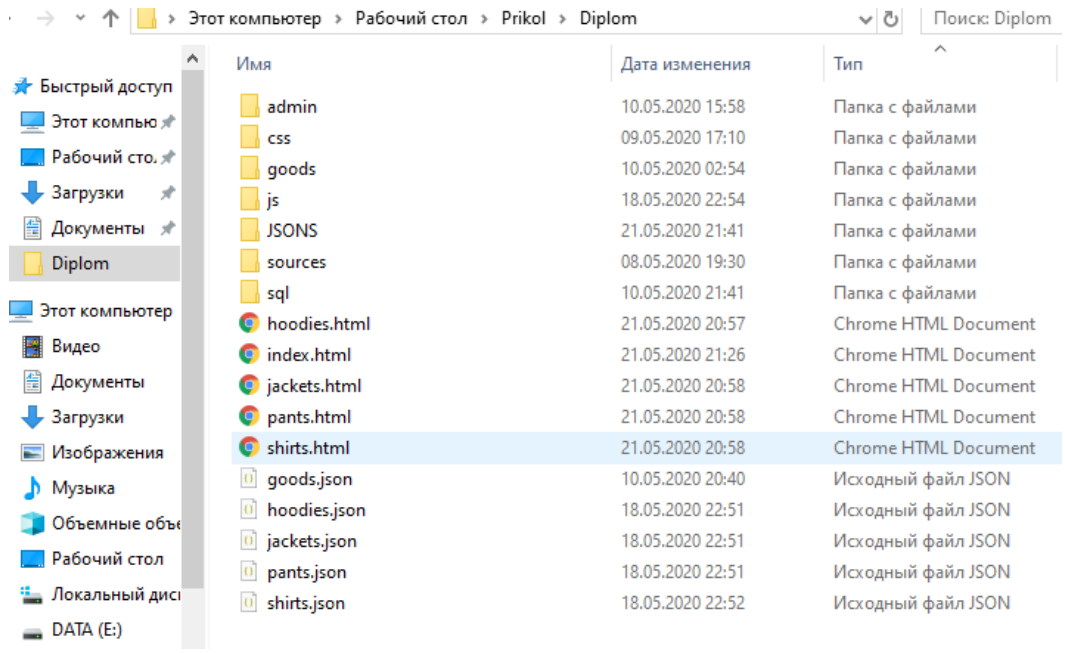


Рисунок 21 – Директорію з JSON файлами

На даний сайт було запроваджено фунцкію пошуку за ім’ям. Обидві функції використовують значення “name” і “size” для взаємодії з JSON файлами, і виводу відповідних товарів, яким відповідають запити у полях пошуку на рис.

15(а,б). Функції можуть працювати одночасно, їх можна побачити на рис. 22, та функцію пошуку за розміром рис. 23.

Рисунок 22 – Функція пошуку за ім’ям

Рисунок 23 – Функція пошуку за розміром

На рис. 24 можна побачити ще одну запроваджену функцію підбору

розміру для клієнтів, а на рис. 25 її код.

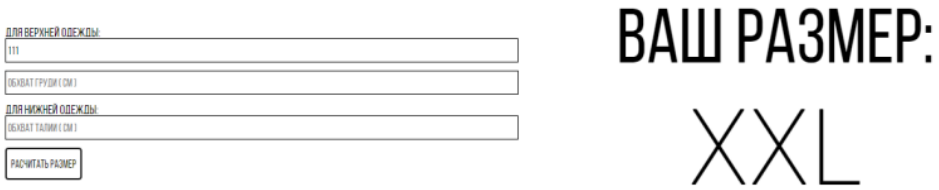


Рисунок 24 – Функція підбору розміру у дії

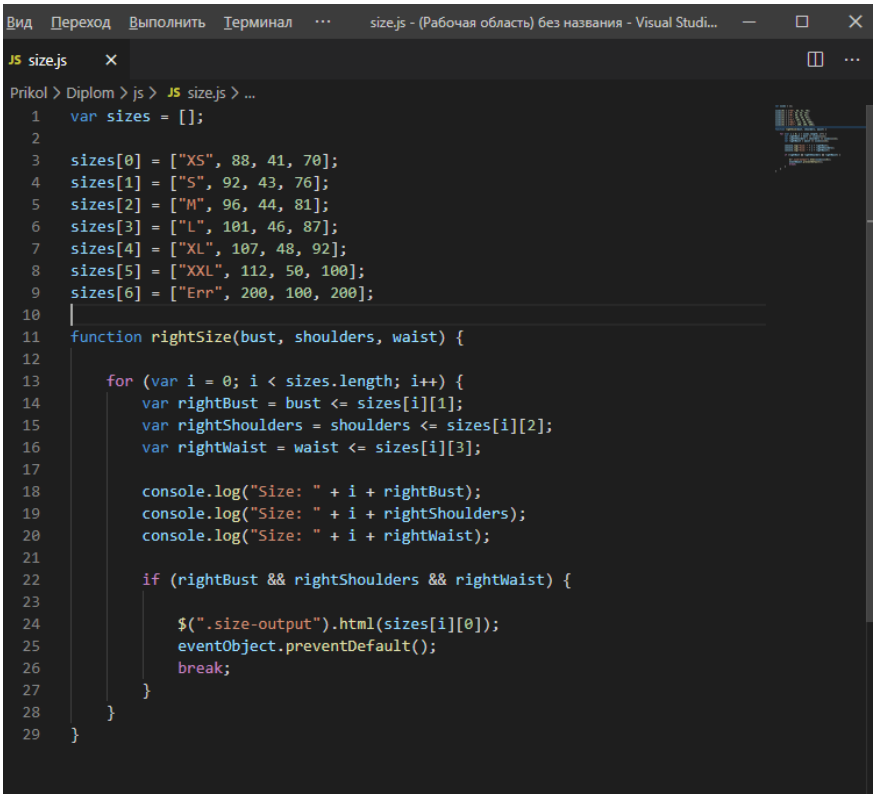


Рисунок 25 – Код функції підбору розміру

4.5 Тестування і розміщення сайту в мережі Інтернет

4.5.1 Web-хостинг

Створений веб-сайт можна розмістити в Інтернеті на Web-сервері. Web-сервер виконує збереження, пошук і обмін файлами в WWW. Наприклад, коли клієнт Web запитує файл із WWW, програма браузер відправляє цей запит на Web-сервер, на якому знаходиться даний файл.

Сервер відшукує файл на свої дисках і відправляє його комп'ютеру-клієнту, від якого був отриманий запит. Обмін інформацією між клієнтом і Web-сервером відбувається відповідно до протоколу HTTP − загального протоколу, що відповідає за функціонування World Wide Web.

Web-сервер − це комп'ютер, що працює під керуванням однієї з операційних систем UNIX, Windows, Macintosh, на який установлена спеціальна програма Web-серевр. Найбільш поширеною програмою Web-сервер є Apache Server, яка працює швидко і встановлюється безкоштовно (див. вузол www.apache.org). Часто використовуються також програми TomCat, Microsoft IIS, NCSA та ін. За допомогою однієї з таких програм свій Web-сервер може створити будь-яка організація, школа і навіть приватна особа. Однак це не завжди виправдано, оскільки повноцінний Web-сервер повинен мати швидке з'єднання з Інтернетом (наприклад, через виділену лінію) і повинен працювати цілодобово. Тому користувачі вирішують проблему Web-серверу за допомогою сторонніх організацій − фірми вашого провайдера чи спеціалізованих фірм, які займаються хостингом.

Веб-хостинг − це фізичне розміщення веб-сторінок на сервері. Від того, де буде розміщено сайт, залежить багато якісних характеристик, тому важливо вибрати оптимальний майданчик для сайту, що відповідає критеріям надійності та стабільності.

4.5.2 Види хостингу

Хостинг − це віртуальний аналог оренди приміщення, але орендується місце на диску, яке обчислюється мегабайтами. Хостинг умовно можна поділити на платний і безкоштовний. [11][1)]

Безкоштовний хостинг передбачає надання хостинг-провайдером безкоштовного дискового простору для розміщення в Інтернеті. Безкоштовний хостинг, зазвичай, існує за рахунок реклами, що розміщується на сторінках сайтів. Ця реклама може бути у вигляді банерів, текстових посилань, рекламних фреймів, спливаючих вікон, хоча існують безкоштовні хостинги, які не розміщують на сайтах жодної реклами, та різноманітні інтеграції.

Основні недоліки безкоштовного хостингу:

– невеликий об’єм, що надається для сайту;

– низька надійність і стабільність серверного майданчика;

– повільне завантаження сайтів;

– присутність реклами;

– часто відсутня підтримка PHP, баз даних та інших даних, що необхідні для повноцінного функціонування сайту;

– відсутність гарантій якісного та постійного надання послуг.

Хостинг є досить привабливим для малобюджетних, любительських чи тимчасових сайтів. Великим попитом користується серед юних розробників-початківців чи щойно створених спільнот.

Плюси безкоштовного хостингу:

Зрештою, інших плюсів безкоштовний хостинг не має, тому, якщо сайт скеровано на довге і стабільне існування, варто задуматися про надійний і

[1)] [11] Покрокове тестування веб-сайтів будь якої складності та напрямку. URL: https://sezzam.com.ua/testuvannya-saytiv/ (дата звернення 29.04.2020)

швидкий комерційний хостинг. [12][1)]

Залежно від країни розташування, хостинг може бути, наприклад: українським (технічний майданчик розташовано в Україні), російським (в Росії), американським (у США) тощо.

У платному хостингу, власник сайту оплачує певну суму за використання дискового простору та сервіси, що йому надаються.

У зв’язку з тестовим режимом, для даного проекту буде використовуватися безкоштовний хостинг. На рис. 26 можна побачити форму реєстрації на сайті www.beget.com. А на рис. 27 можна побачити характеристики хостингу який буде використовуватися.

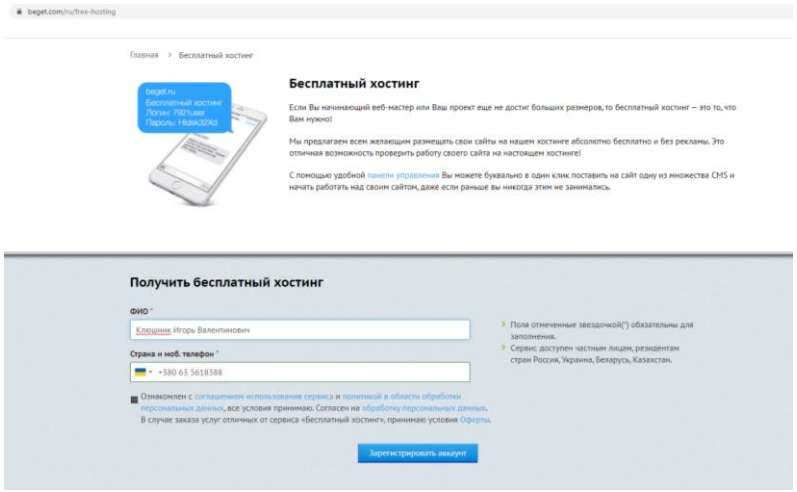
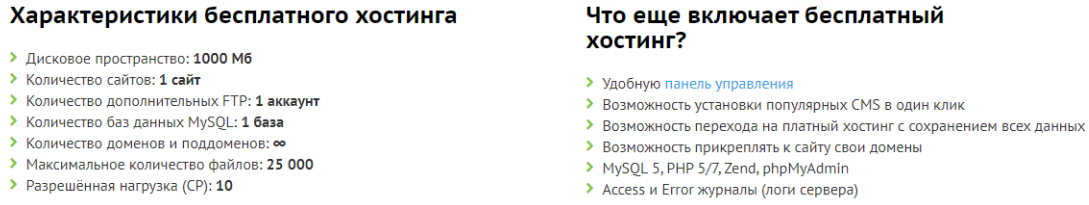


Рисунок 26 – Форма реєстрації для безкоштовного хостингу



[1)] [12] Види тестування ПО. URL: http://qalearning.com.ua/theory/lectures/material/testing-types-functional/ (дата звернення 29.04.2020).

Рисунок 27 – Характеристики безкоштовного хостинг

На рис. 28 можна побачити схему роботи хостингу нового покоління.

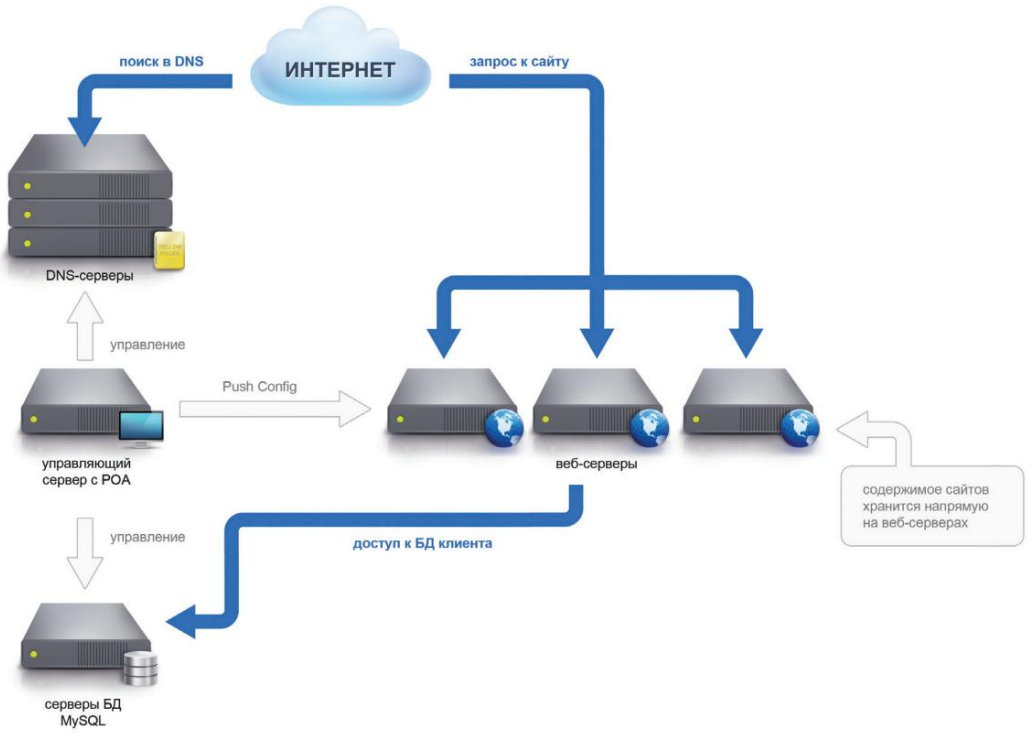


Рисунок 28 – Схема хостинга нового покоління

На рис. 29 можна побачити безкоштовне домене ім’я видане хостингом для веб-сайту Zycsel_Store.

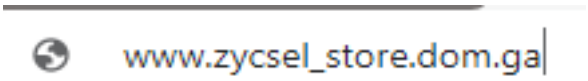


Рисунок 29 – Безкоштовне домене ім’я Zycsel_Store

ВИСНОВКИ

У кваліфікаційній роботі були виконані всі пункти мети. В роботі було досліджено та проаналізовано найбільш важливу і актуальну інформацію щодо розробки веб-ресурсів а саме основні принципи створення веб-ресурсів, їх структуру і функціональність, взаємодію основних компонентів. Також були розглянуті новітні та найбільш перспективні веб-технології, які з успіхом вже використовуються користувачами по всьому світі.

Веб-сайт виконує всі основні завдання та головне призначення програмного продукту − це надання інформації користувачам у мережі інтернет. Інтернет-магазин видає користувачеві готову веб-сторінку, сформовану за запитом користувача на сервері, яка відображає необхідну інформацію та кнопки для здійснення наступних запитів.

Були проведені тести в результаті яких були усунені усі баги та помилки і оптимізована робота веб-сайту. Наявність привабливого дизайну сайту, робить процес покупки приємнішим, також є зручний і простий функціоналу. До нього належать: сортування, фільтрування і підбір товарів або послуг. Також це може бути публікація новин, комунікація з клієнтами та SEO-функції, що покращує якість обслуговування і приваблює клієнтів.

Веб-сайтом інтернет-магазину користуються з різних пристроїв, тому сайт було створено за принципами адаптивної верстки. Користуватися сторінками такого магазину буде набагато зручніше. Веб-сайт має відповідати умовам пошуку потрібних пошукових систем.

Було виконано технічне завдання, та його пункти таке, як проектування системи управління вмістом, яка б дозволяла вносити зміни веб-сайта для людей, які не мають навичок в розробці веб-сайтів.

В процесі розробки веб-ресурсу та оформлення кваліфікаційної роботи на практиці були закріплені теоретичні знання, вдосконалені навички

програмування, проведено оформлення технічної документації в текстових редакторах.

В результаті ми отримали унікальний, який не має аналогів, повністю функціонуючий веб-сайт з адаптивною верстkою, що спеціалізується на продажу товарів одягу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Використання освітніх веб-ресурсів. URL: http://galanet.at.ua (Дата звернення 27.04.2020).
2. Технології планування, розробки та тестування програм. URL: http://moodle.ipo.kpi.ua/moodle/mod/resource/view.php (Дата звернення 01.05.2020).
3. Розробка веб-сайтів на HTML. URL: http://htmlbook.ru (Дата звернення 05.05.2020).
4. Інформаційний ресурс з розробки. CSS. Itchief.ru URL: https://itchief.ru/ (дата звернення 12.05.2020).
5. Скриптова мова програмування JavaScript. URL: https:/learn.javascript.ru (дата звернення 06.05.2020).
6. Архітектура клієнт-сервер. URL: http://inter.ptngu.com/ (Дата звернення 30.04.2020).
7. Визначення база даних. URL: https://www.oracle.com/ru/database/what-is-database.html (Дата звернення 01.05.2020).
8. Інформаційний ресурс з розробки веб-сайтів. Види баз даних. Codenet.ru URL: https://codenet.ru/db/mysql (дата звернення 09.05.2020).).
9. Технології планування, розробки та тестування програм. URL: http://moodle.ipo.kpi.ua/moodle/mod/resource/view.php (Дата звернення 01.05.2020).
10. Використання веб-ресурсів для покращення візуального сприйняття інформації. URL: http://inmad.vntu.edu.ua/portal/index.php (Дата звернення 28.04.2020).
11. Покрокове тестування веб-сайтів будь якої складності та напрямку. URL: https://sezzam.com.ua/testuvannya-saytiv/ (Дата звернення 29.04.2020)
12. Види тестування ПО. URL: http://qalearning.com.ua/theory/

lectures/material/testing-types-functional (Дата звернення 29.04.2020)

ДОДАТОК
ПРОГРАМНИЙ КОД МОДУЛІВ ПРОГРАМИ

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Zycsel_Store</title>
    <link rel="stylesheet" type="text/css" href="/css/style.css">
</head>
<body>


    <div class="container">

        <header class="header">
            <div class=logo><a href=#>Zycsel Store</a></div>
            <div class=logo-mobile><a href=#>ZS</a></div>
            <div class="contacts">
                <div class=contact1>
                <svg xmlns="http://www.w3.org/2000/svg" width="35" height="35" viewBox="0
 0 24 24"><path d="M12 2c5.514 0 10 4.486 10 10s-4.486 10-10 10-10-4.486-10-10 4.486-
10 10-10zm0-2c-6.627 0-12 5.373-12 12s5.373 12 12 12 12-5.373 12-12-5.373-12-12-
12zm4.615 6h-9.23c-.766 0-1.385.62-
1.385 1.384v9.23c0 .766.619 1.386 1.385 1.386h9.23c.766 0 1.385-.62 1.385-1.385v-9.23c0-
.765-.619-1.385-1.385-1.385zm-4.615 3.693c1.274 0 2.309 1.032 2.309 2.307s-1.035 2.307-
2.309 2.307-2.307-1.033-2.307-2.307 1.033-2.307 2.307-2.307zm4.5 6.346c0 .255-.207.461-
.461.461h-8.078c-.254 0-.461-.207-.461-.461v-5.039h.949c-.045.158-.078.32-.102.486-
.023.168-.038.339-.038.514 0 2.04 1.652 3.693 3.691 3.693s3.691-1.653 3.691-3.693c0-.174-
.015-.346-.039-.514-.023-.166-.058-.328-.102-.486h.95v5.039zm0-6.991c0 .255-.207.462-
.461.462h-1.088c-.256 0-.461-.208-.461-.462v-1.087c0-.255.205-.461.461-
.461h1.088c.254 0 .461.207.461.461v1.087z"/></svg> 
                <a href="https://www.instagram.com/zycsel_store/">@Zycsel_Store</a>
                </div>
                <div class=contact2>
                <svg xmlns="http://www.w3.org/2000/svg" width="35" height="35" viewBox="0
 0 24 24"><path d="M12 2c5.514 0 10 4.486 10 10s-4.486 10-10 10-10-4.486-10-10 4.486-
10 10-10zm0-2c-6.627 0-12 5.373-12 12s5.373 12 12 12 12-5.373 12-12-5.373-12-12-
12zm4.5 17.311l-1.76-3.397-1.032.505c-1.12.543-3.4-3.91-2.305-4.497l1.042-.513-1.747-
3.409-1.053.52c-3.601 1.877 2.117 12.991 5.8 11.308l1.055-.517z"/></svg> 
                <span> +38 (063) 561-8388</span>
                </div>
            </div>
        </header>

        <section class="menu">
            <div class="sub-menu" id="a"><a href="index.html">Верхняя одежда</a></div>
            <div class="sub-menu" id="b"><a href="page2.html">Худи и свитшоты</a></div>
            <div class="sub-
menu" id="c"><a href="page3.html">Рубашки и футболки</a></div>
            <div class="sub-menu" id="d"><a href="page4.html">Штаны и шорты</a></div>
            <div class="sub-
menu" id="e"><a href="page5.html">Узнай свой размер!</a></div>
        </section>
        <section class="store-main">
        </section>
<footer class="footer">
    <div class="footer-text"><a href="#">© since 2020<br>Zycsel_Store®</a></div>
</footer>
    </div>
    <script src="./js/jquery-3.5.1.min.js"></script>
    <script src="./js/main.js"></script>
</body>
</html>
```

CSS

```
@font-face {
  font-family: "Font-main";
  src: url(../sources/fonts/Font-main.ttf) format("truetype");
  font-style: normal;
  font-weight: normal;
```

```
}

@font-face {
  font-family: "Font-sub";
  src: url(../sources/fonts/Font-sub.ttf) format("truetype");
  font-style: normal;
  font-weight: normal;
}

html,
body,
.container {
  padding: 0;
  margin: 0;
  border: 0;
  width: 100%;
}

.header {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  width: 100%;
  height: 250px;
  background-image: -webkit-
gradient(linear, left top, left bottom, from(rgba(0, 0, 0, 0.8)), to(rgba(255, 255, 255,
0.8))), url(../sources/images/Header2.jpg);
  background-image: linear-
gradient(to bottom, rgba(0, 0, 0, 0.8) 0%, rgba(255, 255, 255, 0.8) 100%), url(../sources
/images/Header2.jpg);
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
  -webkit-box-shadow: 0 5px 10px rgba(0, 0, 0, 0.5);
          box-shadow: 0 5px 10px rgba(0, 0, 0, 0.5);
}

@media only screen and (min-device-width: 0px) and (max-device-width: 767px) {
  .header {
    margin-bottom: 40px;
  }
}

.header .logo {
  position: relative;
  margin: auto;
  height: -webkit-fit-content;
  height: -moz-fit-content;
  height: fit-content;
  width: -webkit-fit-content;
  width: -moz-fit-content;
  width: fit-content;
  font-size: 120px;
  font-family: "Font-main";
  text-shadow: -0 -1px 1px #000000,
 0 -1px 1px #000000,
 -0 1px 1px #000000,
 0 1px 1px #000000,
 -1px -0 1px #000000,
 1px -0 1px #000000,
 -1px 0 1px #000000,
 1px 0 1px #000000,
 -1px -1px 1px #000000,
 1px -1px 1px #000000,
 -1px 1px 1px #000000,
 1px 1px 1px #000000,
 -1px -1px 1px #000000,
 1px -1px 1px #000000,
 -1px 1px 1px #000000,
 1px 1px 1px #000000;
  color: white;
  letter-spacing: 27px;
}
```

```
@media only screen and (min-device-width: 0px) and (max-device-width: 767px) {
  .header .logo {
    display: none;
  }
}

.header .logo a,
.header .logo a:visited {
  text-decoration: none;
  color: white;
}

.header .logo-mobile {
  height: -webkit-fit-content;
  height: -moz-fit-content;
  height: fit-content;
  width: -webkit-fit-content;
  width: -moz-fit-content;
  width: fit-content;
  margin: auto;
  font-size: 150px;
  font-family: "Font-main";
  color: white;
  text-shadow: -0 -1px 1px #000000,
 0 -1px 1px #000000,
 -0 1px 1px #000000,
 0 1px 1px #000000,
 -1px -0 1px #000000,
 1px -0 1px #000000,
 -1px 0 1px #000000,
 1px 0 1px #000000,
 -1px -1px 1px #000000,
 1px -1px 1px #000000,
 -1px 1px 1px #000000,
 1px 1px 1px #000000,
 -1px -1px 1px #000000,
 1px -1px 1px #000000,
 -1px 1px 1px #000000,
 1px 1px 1px #000000;
  letter-spacing: 27px;
}

.header .logo-mobile a,
.header .logo-mobile a:visited {
  text-decoration: none;
  color: white;
}

@media only screen and (min-width: 1224px) {
  .header .logo-mobile {
    display: none;
  }
}

@media only screen and (min-device-width: 768px) and (max-device-width: 1223px) {
  .header .logo-mobile {
    display: none;
  }
}

.header .contacts {
  position: absolute;
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  -webkit-box-pack: justify;
      -ms-flex-pack: justify;
          justify-content: space-between;
  width: 98%;
  height: -webkit-fit-content;
  height: -moz-fit-content;
  height: fit-content;
  margin: 10px 30px 10px 20px;
}
```

```
@media only screen and (min-device-width: 0px) and (max-device-width: 767px) {
  .header .contacts {
    width: 90%;
  }
}

.header .contacts .contact2 {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  text-align: center;
  height: -webkit-fit-content;
  height: -moz-fit-content;
  height: fit-content;
}

@media only screen and (min-device-width: 0px) and (max-device-width: 767px) {
  .header .contacts .contact2 {
    font-size: 10px;
  }
}

.header .contacts .contact2 svg {
  fill: white;
}

.header .contacts .contact2 span {
  -ms-flex-item-align: center;
      -ms-grid-row-align: center;
      align-self: center;
  color: white;
  font-family: "Font-sub";
}

.header .contacts .contact2 span:hover {
  -webkit-transition: .5s ease-in-out;
  transition: .5s ease-in-out;
  color: orange;
  cursor: pointer;
  text-shadow: -0 -1px 1px #000000,
 0 -1px 1px #000000,
 -0 1px 1px #000000,
 0 1px 1px #000000,
 -1px -0 1px #000000,
 1px -0 1px #000000,
 -1px 0 1px #000000,
 1px 0 1px #000000,
 -1px -1px 1px #000000,
 1px -1px 1px #000000,
 -1px 1px 1px #000000,
 1px 1px 1px #000000,
 -1px -1px 1px #000000,
 1px -1px 1px #000000,
 -1px 1px 1px #000000,
 1px 1px 1px #000000;
}

.header .contacts .contact1 {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  -webkit-box-align: center;
      -ms-flex-align: center;
          align-items: center;
  height: -webkit-fit-content;
  height: -moz-fit-content;
  height: fit-content;
}

@media only screen and (min-device-width: 0px) and (max-device-width: 767px) {
  .header .contacts .contact1 {
    font-size: 10px;
  }
```

```
}

.header .contacts .contact1 svg {
  fill: white;
}

.header .contacts .contact1 a,
.header .contacts .contact1 a:visited {
  -ms-flex-item-align: center;
      -ms-grid-row-align: center;
      align-self: center;
  text-decoration: none;
  color: white;
  font-family: "Font-sub";
}

.header .contacts .contact1 a:hover {
  -webkit-transition: .5s ease-in-out;
  transition: .5s ease-in-out;
  color: orange;
  cursor: pointer;
  text-shadow: -0 -1px 1px #000000,
 0 -1px 1px #000000,
 -0 1px 1px #000000,
 0 1px 1px #000000,
 -1px -0 1px #000000,
 1px -0 1px #000000,
 -1px 0 1px #000000,
 1px 0 1px #000000,
 -1px -1px 1px #000000,
 1px -1px 1px #000000,
 -1px 1px 1px #000000,
 1px 1px 1px #000000,
 -1px -1px 1px #000000,
 1px -1px 1px #000000,
 -1px 1px 1px #000000,
 1px 1px 1px #000000;
}

.menu {
  height: -webkit-fit-content;
  height: -moz-fit-content;
  height: fit-content;
  margin: auto;
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  width: 90%;
  -webkit-box-pack: justify;
      -ms-flex-pack: justify;
          justify-content: space-between;
  margin-top: 20px;
  margin-bottom: 40px;
}

@media only screen and (min-device-width: 0px) and (max-device-width: 767px) {
  .menu {
    display: none;
  }
}

.menu .sub-menu {
  width: -webkit-fit-content;
  width: -moz-fit-content;
  width: fit-content;
  text-align: center;
}

.menu .sub-menu a,
.menu .sub-menu a:visited {
  font-size: 21px;
  color: black;
  font-family: "Font-sub";
  text-decoration: none;
```

```
}

.menu .sub-menu a:hover {
  -webkit-transition: .5s ease-in-out;
  transition: .5s ease-in-out;
  color: orange;
  cursor: pointer;
  text-shadow: -0 -1px 1px #000000,
 0 -1px 1px #000000,
 -0 1px 1px #000000,
 0 1px 1px #000000,
 -1px -0 1px #000000,
 1px -0 1px #000000,
 -1px 0 1px #000000,
 1px 0 1px #000000,
 -1px -1px 1px #000000,
 1px -1px 1px #000000,
 -1px 1px 1px #000000,
 1px 1px 1px #000000,
 -1px -1px 1px #000000,
 1px -1px 1px #000000,
 -1px 1px 1px #000000,
 1px 1px 1px #000000;
}

.store-main {
  display: -ms-grid;
  display: grid;
  width: 90%;
  margin: auto;
  justify-items: center;
  -ms-grid-columns: 1fr 1fr 1fr;
      grid-template-columns: 1fr 1fr 1fr;
  -webkit-column-gap: 7%;
          column-gap: 7%;
  row-gap: 100px;
  margin-bottom: 150px;
  max-width: 2000px;
}

@media only screen and (min-device-width: 768px) and (max-device-width: 1223px) {
  .store-main {
    -ms-grid-columns: 1fr 1fr;
        grid-template-columns: 1fr 1fr;
  }
}

@media only screen and (min-device-width: 0px) and (max-device-width: 767px) {
  .store-main {
    -ms-grid-columns: 1fr;
        grid-template-columns: 1fr;
  }
}

.store-main .store-item {
  width: 100%;
  height: 400px;
  background-color: white;
  -webkit-box-shadow: 0px 2px 10px rgba(0, 0, 0, 0.5);
          box-shadow: 0px 2px 10px rgba(0, 0, 0, 0.5);
  min-width: 340px;
}

.store-main .store-item .item-image {
  display: block;
  margin: 0 auto;
  height: 60%;
  margin-top: 20px;
  margin-bottom: 20px;
}

.store-main .store-item .item-desctiption {
  padding-top: 10px;
  font-family: "Font-sub";
```

```
  font-size: 20px;
  color: black;
  text-align: center;
  margin: auto;
  width: 80%;
  height: 20%;
  -webkit-box-shadow: 0 0px 6px rgba(0, 0, 0, 0.5);
          box-shadow: 0 0px 6px rgba(0, 0, 0, 0.5);
}

.footer {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
  -webkit-box-align: end;
      -ms-flex-align: end;
          align-items: flex-end;
  -webkit-box-pack: center;
      -ms-flex-pack: center;
          justify-content: center;
  height: 150px;
  width: 100%;
  background-image: -webkit-
gradient(linear, left bottom, left top, from(rgba(0, 0, 0, 0.4)), to(rgba(255, 255, 255,
0.4))), url(../sources/images/footer.jpg);
  background-image: linear-
gradient(to top, rgba(0, 0, 0, 0.4) 0%, rgba(255, 255, 255, 0.4) 100%), url(../sources/im
ages/footer.jpg);
  background-size: cover;
  background-position: center;
  background-repeat: no-repeat;
  -webkit-box-shadow: 0px -5px 10px rgba(0, 0, 0, 0.5);
          box-shadow: 0px -5px 10px rgba(0, 0, 0, 0.5);
}

.footer .footer-text {
  font-family: "Font-sub";
  color: white;
  text-align: center;
}

.footer .footer-text a,
.footer .footer-text :active {
  color: white;
  text-decoration: none;
}

.footer .footer-text a:hover {
  -webkit-transition: .5s ease-in-out;
  transition: .5s ease-in-out;
  color: orange;
  cursor: pointer;
  text-shadow: -0 -1px 1px #000000,
 0 -1px 1px #000000,
 -0 1px 1px #000000,
 0 1px 1px #000000,
 -1px -0 1px #000000,
 1px -0 1px #000000,
 -1px 0 1px #000000,
 1px 0 1px #000000,
 -1px -1px 1px #000000,
 1px -1px 1px #000000,
 -1px 1px 1px #000000,
 1px 1px 1px #000000,
 -1px -1px 1px #000000,
 1px -1px 1px #000000,
 -1px 1px 1px #000000,
 1px 1px 1px #000000;
}

.footer .footer-text:hover {
  -webkit-transition: .5s ease-in-out;
  transition: .5s ease-in-out;
  color: orange;
```

```
  cursor: pointer;
  text-shadow: -0 -1px 1px #000000,
 0 -1px 1px #000000,
 -0 1px 1px #000000,
 0 1px 1px #000000,
 -1px -0 1px #000000,
 1px -0 1px #000000,
 -1px 0 1px #000000,
 1px 0 1px #000000,
 -1px -1px 1px #000000,
 1px -1px 1px #000000,
 -1px 1px 1px #000000,
 1px 1px 1px #000000,
 -1px -1px 1px #000000,
 1px -1px 1px #000000,
 -1px 1px 1px #000000,
 1px 1px 1px #000000;
}
/*# sourceMappingURL=style.css.map */
```

SCSS

```
@import "./mixins.scss";

@font-face {
    font-family: "Font-main";
    src: url(../sources/fonts/Font-main.ttf) format("truetype");
    font-style: normal;
    font-weight: normal;
}

@font-face {
    font-family: "Font-sub";
    src: url(../sources/fonts/Font-sub.ttf) format("truetype");
    font-style: normal;
    font-weight: normal;
}

$main-gradient: linear-gradient(to bottom, rgba(0, 0, 0, 0.8) 0%, rgba(255, 255, 255,
0.8) 100%);
$sub-gradient: linear-gradient(to top, rgba(0, 0, 0, 0.4) 0%, rgba(255, 255, 255, 0.4)
100%);
$main-font: "Font-main";
$sub-font:"Font-sub";
$shadow: 0 0px 6px rgba(0, 0, 0, 0.5);

html,
body,
.container {
    padding: 0;
    margin: 0;
    border: 0;
    width: 100%;
}

.header {
    display: flex;
    width: 100%;
    height: 250px;
    background-image: $main-gradient,
        url(../sources/images/Header2.jpg);
    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;
    box-shadow: 0 5px 10px rgba(0, 0, 0, 0.5);

    @include mobile {
        margin-bottom: 40px;
    }

    .logo {
        position: relative;
        margin: auto;
        height: fit-content;
        width: fit-content;
```

```
    font-size: 120px;
    font-family: $main-font;
    @include text-stroke;
    color: white;
    letter-spacing: 27px;

    @include mobile {
        display: none;
    }

    a,
    a:visited {
        text-decoration: none;
        color: white;

    }
}

.logo-mobile {
    height: fit-content;
    width: fit-content;
    margin: auto;
    font-size: 150px;
    font-family: $main-font;
    color: white;
    @include text-stroke;
    letter-spacing: 27px;

    a,
    a:visited {
        text-decoration: none;
        color: white;
    }

    @include desctop {
        display: none;
    }

    @include tablet {
        display: none;
    }
}

.contacts {
    position: absolute;
    display: flex;
    justify-content: space-between;
    width: 98%;
    height: fit-content;
    margin: 10px 30px 10px 20px;

    @include mobile {
        width: 90%;
    }

    .contact2 {

        display: flex;
        text-align: center;
        height: fit-content;

        @include mobile {
            font-size: 10px;
        }

        svg {
            fill: white;
        }

        span {
            align-self: center;
            color: white;
            font-family: $sub-font;
```

```
            }

            span:hover {
                @include animation;
            }
        }

        .contact1 {
            display: flex;
            align-items: center;
            height: fit-content;

            @include mobile {
                font-size: 10px;
            }

            svg {

                fill: white;
            }

            a,
            a:visited {
                align-self: center;
                text-decoration: none;
                color: white;
                font-family: $sub-font;
            }

            a:hover {
                @include animation;
            }
        }

    }
}

.menu {
    height: fit-content;
    margin: auto;
    display: flex;
    width: 90%;
    justify-content: space-between;
    margin-top: 20px;
    margin-bottom: 40px;

    @include mobile {
        display: none;
    }


    .sub-menu {
        width: fit-content;
        text-align: center;

        a,
        a:visited {
            font-size: 21px;
            color: black;
            font-family: $sub-font;
            text-decoration: none;
        }

        a:hover {
            @include animation;
        }
    }
}

.store-main {
    display: grid;
    width: 90%;
    margin: auto;
    justify-items: center;
```

```
    grid-template-columns: 1fr 1fr 1fr;
    column-gap: 7%;
    row-gap: 100px;
    margin-bottom: 150px;
    max-width: 2000px;

    @include tablet {
        grid-template-columns: 1fr 1fr;
    }

    @include mobile {
        grid-template-columns: 1fr;
    }

    .store-item {
        width: 100%;
        height: 400px;
        background-color: white;
        box-shadow: 0px 2px 10px rgba(0, 0, 0, 0.5);
        min-width: 340px;

        .item-image {
            display: block;

            margin: 0 auto;
            height: 60%;
            margin-top: 20px;
            margin-bottom: 20px;


        }

        .item-desctiption {
            padding-top: 10px;
            @include sub-font;
            text-align: center;
            margin: auto;
            width: 80%;
            height: 20%;
            box-shadow: $shadow;
        }
    }
}

.footer {
    display: flex;
    align-items: flex-end;
    justify-content: center;
    height: 150px;
    width: 100%;
    background-image: $sub-gradient,
        url(../sources/images/footer.jpg);
    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;
    box-shadow: 0px -5px 10px rgba(0, 0, 0, 0.5);
    ;

    .footer-text {
        font-family: $sub-font;
        color: white;
        text-align: center;

        a,
        :active {
            color: white;
            text-decoration: none;
        }

        a:hover {
            @include animation;
        }
    }
```

```
    .footer-text:hover {
        @include animation;
    }
}

MIXINS
@mixin desctop {
    @media only screen and (min-width : 1224px) {
        @content;
    }
}

@mixin tablet {
    @media only screen and (min-device-width : 768px) and (max-device-width : 1223px) {
        @content;
    }
}

@mixin mobile {
    @media only screen and (min-device-width : 0px) and (max-device-width : 767px) {
        @content;
    }
}

@mixin text-stroke {
    text-shadow:
        -0 -1px 1px #000000,
        0 -1px 1px #000000,
        -0 1px 1px #000000,
        0 1px 1px #000000,
        -1px -0 1px #000000,
        1px -0 1px #000000,
        -1px 0 1px #000000,
        1px 0 1px #000000,
        -1px -1px 1px #000000,
        1px -1px 1px #000000,
        -1px 1px 1px #000000,
        1px 1px 1px #000000,
        -1px -1px 1px #000000,
        1px -1px 1px #000000,
        -1px 1px 1px #000000,
        1px 1px 1px #000000;
}

@mixin animation {
    transition: .5s ease-in-out;
    color: orange;
    cursor: pointer;
       @include text-stroke();
}

@mixin sub-font {
    font-family: $sub-font;
    font-size: 20px;
    color: black;
}

JAVASCRIPT
function init(){
    $.getJSON("goods.json", goodsOut);
}
    function goodsOut(data){
        console.log(data);
        var out='';
        for (var key in data) {
            out +='<div class="store-item">';
            out +=`<img class="item-image" src="./goods/${data[key].image}" alt="">`;
            out +='<div class="item-desctiption">';
            out +=`<span class="name">${data[key].name}</span>`;
            out +='<br>';
            out +=`<span class="size">${data[key].size}</span>`;
            out +='<br>';
            out +=`<span class="cost">${data[key].cost}</span>`;
            out +='</div>';
```

```
            out +='</div>';

        }
        $('.store-main').html(out);
    }
    $(document).ready(function () {
        init();
});
```

JSON (Альтернатива SQL)

```
{
            "0001": {
                "name": "JUNIOR Hooded Jacket",
                "size": "XL",
                "cost": "5000 Грн",
                "image": "0001.jpg"
            },
            "0002": {
                "name": "Soft Shell-R Jacket",
                "size": "L",
                "cost": "4800 Грн",
                "image": "0002.jpg"
        },
            "0003": {
                "name": "SI Sweatshirt",
                "size": "M",
                "cost": "2100 Грн",
                "image": "0003.jpg"
            },
            "0004": {
                "name": "SI Hoodie",
                "size": "S",
                "cost": "1900 Грн",
                "image": "0004.jpg"
            },
            "0005": {
                "name": "SI T-shirt",
                "size": "M",
                "cost": "1500 Грн",
                "image": "0005.jpg"
            },
            "0006": {
                "name": "Shirt",
                "size": "XXL",
                "cost": "1700 Грн",
                "image": "0006.jpg"
        },

            "0007": {
                "name": "SI Pants",
                "size": "XL",
                "cost": "1900 Грн",
                "image": "0007.jpg"
            },
            "0008": {
                "name": "SI Shorts",
                "size": "S",
                "cost": "1500 Грн",
                "image": "0008.jpg"
            }
}
-- phpMyAdmin SQL Dump
-- version 5.0.2
-- https://www.phpmyadmin.net/
--
-- Хост: 127.0.0.1:3306
-- Время создания: Май 10 2020 г., 21:09
-- Версия сервера: 10.3.22-MariaDB
-- Версия PHP: 7.1.33

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- База данных: `goods`
--

-- --------------------------------------------------------

--
-- Структура таблицы `goods`
--

CREATE TABLE `goods` (
  `id` int(4) NOT NULL,
  `name` varchar(100) NOT NULL,
  `cost` varchar(20) DEFAULT NULL,
  `size` text DEFAULT NULL,
  `type` varchar(20) DEFAULT NULL,
  `image` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Дамп данных таблицы `goods`
--

INSERT INTO `goods` (`id`, `name`, `cost`, `size`, `type`, `image`) VALUES
(1, 'SI JUNIOR Hooded Jacket', '5000 UAH', 'XL', 'jacket', '0001.jpg'),
(2, 'SI Light Soft Shell-R Jacket', '4800 UAH', 'L', 'jacket', '0002.jpg'),
(3, 'SI Sweatshirt', '3100 UAH', 'M', 'sweater', '0003.jpg'),
(4, 'SI Hoodie', '3600 UAH', 'S', 'sweater', '0004.jpg'),
(5, 'SI T-Shirt', '1500 UAH', 'M', 'shirt', '0005.jpg'),
(6, 'SI Shirt', '1700 UAH', 'XXL', 'shirt', '0006.jpg'),
(7, 'SI Pants', '2300 UAH', 'XL', 'pants', '0007.jpg'),
(8, 'SI Shorts', '1500 UAH', 'S', 'pants', '0008.jpg');

--
-- Индексы сохранённых таблиц
--

--
-- Индексы таблицы `goods`
--
ALTER TABLE `goods`
  ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT для сохранённых таблиц
--

--
-- AUTO_INCREMENT для таблицы `goods`
--
ALTER TABLE `goods`
  MODIFY `id` int(4) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```