|  |
| --- |
| **Міністерство освіти і науки України**  **Відокремлений структурний підрозділ «Тернопільський фаховий коледж Тернопільського національного технічного університету імені Івана Пулюя»** |
| (повне найменування вищого навчального закладу) |
| відділення телекомунікацій та електронних систем |
| (назва відділення) |
| циклова комісія комп’ютерних наук |
| (повна назва циклової комісії) |

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

**до курсової роботи**

|  |
| --- |
|  |
| (освітньо-кваліфікаційний рівень) |

**на тему:** Розробка веб-додатку управління компонентами інтернет-магазину

принтерів

Виконав: студент (ка) 3 курсу, групи КН-321

напряму підготовки (спеціальності)

122 «Комп’ютерні науки»

(шифр і назва напряму підготовки, спеціальності)

Шозда Ю.Я.

(прізвище та ініціали)

Керівник Лісовий В.М

(прізвище та ініціали)

Оцінка:

Члени комісії:

(підпис) (прізвище та ініціали)

м. Тернопіль – 2022

Завдання на курсовий проект

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ «ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО

ТЕХНІЧНОГО УНІВЕРСИТЕТУ ІМЕНІ ІВАНА ПУЛЮЯ»

Відділення телекомунікацій та електронних систем

Циклова комісія комп’ютерних наук

Освітньо-кваліфікаційний рівень молодший спеціаліст

Напрям підготовки 122 « Комп’ютерні науки»

(шифр і назва)

Галузь знань 12 «Інформаційні технології»

(шифр і назва)

**ЗАТВЕРДЖУЮ**

Голова циклової комісіїпрограмних систем і комплексів

**\_\_\_\_\_\_\_\_\_\_\_\_** Г.Я. Марціяш “\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ року

**З А В Д А Н Я**

**НА КУРСОВУ РОБОТУ СТУДЕНТУ**

Шозда Юлія Ярославівна

( прізвище, ім’я, по батькові)

1. Тема роботи: Розробка веб-додатку управління компонентами

інтернет-магазину принтерів

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

керівник роботи: Лісовий В. М.

( прізвище, ім’я, по батькові)

2. Строк подання студентом роботи: \_17 травня 2022р\_\_\_\_\_\_\_\_

3. Вихідні дані до роботи \_\_Інформація по темі курсової роботи, мови HTML5, CSS3, JavaScript, стандарти мов програмування \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Висновки\_\_\_Виконати висновки по результатах виконаної роботи та прийнятих рішень\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Додаткові вказівки:

Виконання проекту (з виготовленням макета, стенда, приладу і т.д.)

\_\_\_\_\_\_з написанням сайту та розміщенням його в мережі інтернет і вихідних кодів в github\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата видачі завдання « 9 » 03 2022 р.

Студент Шозда Ю.Я.

( підпис ) (прізвище та ініціали)

Керівник роботи Лісовий В.М.

( підпис ) (прізвище та ініціали)

ЗМІСТ

[ВСТУП …………………………………………………………………………….. 6](#_Toc105849217)

[1 ЗАГАЛЬНИЙ РОЗДІЛ 7](#_Toc105849218)

[1.1 Аналітичний огляд існуючих рішень 7](#_Toc105849219)

[1.2 Технічне завдання 7](#_Toc105849220)

[2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ 11](#_Toc105849221)

[2.1 Розробка структури сайту і web-сторінок 11](#_Toc105849222)

[2.2 Створення та верстка сторінок сайту 15](#_Toc105849223)

[2.3 Розробка структури бази даних сайту 19](#_Toc105849224)

[2.4 Програмування сайту 22](#_Toc105849225)

[2.5 Тестування сайту 25](#_Toc105849226)

[3 СПЕЦІАЛЬНИЙ РОЗДІЛ 27](#_Toc105849227)

[3.1 Розміщення сайту в Інтернеті 27](#_Toc105849228)

[3.2 Інструкція з обслуговування та наповнення сайту 27](#_Toc105849229)

[3.3 Інструкція з популяризації та підтримки сайту 29](#_Toc105849230)

[ВИСНОВКИ 30](#_Toc105849231)

[ПЕРЕЛІК ПОСИЛАНЬ 31](#_Toc105849232)

[Додаток А Лістинг методу «buildElementToPage» 32](#_Toc105849233)

[Додаток Б Тестування додатку «Головне меню» 33](#_Toc105849234)

[Додаток В Тестування додатку «Створення принтеру» 34](#_Toc105849235)

[Додаток Д Тестування додатку «Навігаційне меню» 35](#_Toc105849236)

[Додаток Е Тестування додатку «Категорії принтеру» 36](#_Toc105849237)

[Додаток Е Тестування додатку «Підменю» 37](#_Toc105849238)

[Додаток Є Тестування додатку «Головний екран» 38](#_Toc105849239)

[Додаток Ж Тестування додатку «Асортимент» 39](#_Toc105849240)

[Додаток З Лістинг файлу «basic.html» 40](#_Toc105849241)

[Додаток И Лістинг файлу «basic.js» 44](#_Toc105849242)

[Додаток І Лістинг файлу «audit.js» 51](#_Toc105849243)

[Додаток Й Лістинг файлу «localStorage.js» 52](#_Toc105849244)

[Додаток К Лістинг файлу «basic.css» 55](#_Toc105849245)

[Додаток Л Лістинг файлу «button.css» 61](#_Toc105849246)

[Додаток М Лістинг файлу «modal.css» 64](#_Toc105849247)

[Додаток Н Лістинг файлу «component.css» 66](#_Toc105849248)

[Додаток Компакт-диск із програмним продуктом 68](#_Toc105849249)

ВСТУП

Сучасна практика програмування вимагає вдосконалення прийомів розробки web-сторінок, прискорення та полегшення процесу їх проектування. При цьому програмні засоби повинні бути орієнтовані на тривалу роботу з користувачем. Це ставить перед розробником-програмістом завдання, що спрямовані не тільки на технологічні вимоги організації інтерфейсу програми, а й на психолого-фізіологічні особливості сприймання людиною інформації.

Ріст приватного сектору економіки призвів до перегляду об’ємів витрат на комп’ютерну техніку та на спеціалізоване програмне забезпечення. Набагато вигідніше стало розробляти нескладні програмні засоби, орієнтовані на розв’язання вузько спеціалізованих завдань. Тому, перед навчальними закладами постала мета підготувати спеціалістів, які спроможні створювати гнучкі програмні засоби, не витрачаючи на їх розробку та впровадження багато фінансів та часу.

Як приклад, в даній курсовій роботі представлено розробку одного із таких спеціалізованих програмних рішень мовою HTML, CSS та Java Script. Ціль його розробки – оптимізація процесу вибору/купівлі принтеру для користувачів.

# ЗАГАЛЬНИЙ РОЗДІЛ

## Аналітичний огляд існуючих рішень

Із-за підвищення ефективності праці людини в буденному житті, інноваційних нововведень в обліку, заповнення різних форм та проблематика психології людини у повсякденному часі: невпинний рух обставин або «швидка ейфорія» – виникла потреба у швидкому та легко читабельному друці на папері. Необхідність людини виражати свої думки на папері заставляє її задуматись над покупкою принтера.

З вище зазначеного тексту можна виразити наступні проблематики, які варто рішити, для полегшення користувачам використання сайту:

* Виробник/Модель;
* Максимальний формат паперу (А4/А3…) ;
* Швидкість друку, ст./хв.;
* Ресурс картриджа, ст.

Сайт виконує опцію інтернет-магазину та являється експлуатаційним призначенням даної розробки.

Відповідно функціональне призначення – оптимізація і спрощення ведення обліку і виведення наявних товарів в магазині, та можливість отримати їх через посередництво пошти України.

## Технічне завдання

### Найменування та область застосування

Найменування web-додатку – управління компонентами інтернет-магазину принтерів.

Стисла назва - "MyPrinters".

Область застосування додатку – веб-сайт для продажі/покупки принтеру в доступній та легко зрозумілій. Підвищення кількості покупців та налагодженні легкої реалізації розмови покупця та продавця.

### Вимоги до функціоналу web-сайту

#### Вимоги до функціональних характеристик

Web-додаток «управління компонентами інтернет-магазину принтерів» повинен забезпечити оптимізацію і спрощення ведення таких форм обліку:

* Перелік узагальнених назв принтерів. Тут містяться їхні id, назва категорії. Таблиця складається з 2 стовпців, де кожен новий рядок ­– нова категорія. Дану таблицю представлено нижче.

Таблиця 1.1 – Перелік узагальнених назв принтерів

|  |  |
| --- | --- |
| id | Name |
| 1 | Brother |
| 2 | Canon |
| 3 | Dell |
| 4 | Epson |
| 5 | HP |
| 6 | Jetix |
| 7 | Kodak |
| 8 | Kyocera |
| 9 | Polaroid |
| 10 | Samsung |

* Перелік самих принтерів. Їхні id, повна назва; максимальний формат паперу, А4, А5 і т.д.; швидкість друку, ст./хв.; ресурс картриджа, ст.; ціна, UAH. Таблиця складається з 7 стовпців, де 7 стовпець ­– зв'язаний по id з таблицею 1.1, тобто є під категорією. Дану таблицю представлено нижче.

Таблиця 1.2 – Перелік принтерів

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| id | Name | Size | Speed | Resource | Price | link |
| 1 | 3D FlashForge | A1 | 1 | 20 | 24 000 | 1 |
| 2 | Brother DCP-1510E | A4 | 24 | 1000 | 7 139 | 1 |
| 3 | Brother DCP-L2500DR | A4 | 26 | 1200 | 7 999 | 1 |
| 4 | Brother DCP-L3550CDW | A4 | 18 | 2000 | 28 674 | 1 |
| 5 | Brother DCP-T420W | A4 | 16 | 1000 | 6 299 | 1 |
| 6 | BROTHER HL-1110R | A4 | 10 | 700 | 5 663 | 1 |
| 7 | BROTHER HL-1210WE | A4 | 20 | 650 | 5 229 | 1 |
| 8 | Brother HL-L2300DR | A4 | 26 | 650 | 5 334 | 1 |
| 9 | Brother HL-L2340DWR with Wi-Fi | A4 | 26 | 700 | 6 521 | 1 |
| 10 | Brother MFC-L5750DWR з Wi-Fi | A4 | 40 | 2500 | 25 658 | 1 |
| 14 | Canon i-Sensys | A4 | 18 | 1150 | 11 599 | 2 |
| 11 | CANON i-Sensys LBP6030B | A4 | 18 | 960 | 6 300 | 2 |
| 12 | Canon I-SENSYS MF267dw | A4 | 25 | 1200 | 15 499 | 2 |
| 13 | Canon i-Sensys MF3010 | A4 | 18 | 1200 | 12 100 | 2 |
| 15 | Canon Laser T140 | A4 | 16 | 840 | 8 300 | 2 |
| 16 | Canon MAXIFY GX6040 | A4 | 5 | 300 | 12 000 | 2 |
| 17 | CANON PIXMA E414 | A6 | 3 | 120 | 4 199 | 2 |
| 18 | Canon PIXMA G540 | A6 | 4 | 250 | 9 299 | 2 |
| 19 | Canon SELPHY CP-1300 Pink | A4 | 5 | 20 | 5 990 | 2 |
| 20 | Canon SELPHY CP-1300 White | A4 | 5 | 30 | 6 120 | 2 |
| 21 | Dell B2360Dn Mono | A4 | 16 | 800 | 7 597 | 3 |
| 22 | Dell E310dw | A4 | 26 | 600 | 4 589 | 3 |
| 23 | Epson L110 | A4 | 18 | 750 | 6 300 | 4 |
| 27 | Epson L1300 | A4 | 6 | 650 | 7 500 | 4 |
| 24 | Epson L222 | A4 | 12 | 420 | 9 000 | 4 |
| 25 | Epson L300 | A4 | 11 | 560 | 8 999 | 4 |
| 28 | Epson L3156 | A4 | 14 | 430 | 9 600 | 4 |
| 29 | Epson L6490 with Wi-Fi | A4 | 18 | 1200 | 26 030 | 4 |
| 30 | Epson L6550 A4 with Wi-Fi | A4 | 12 | 850 | 21 350 | 4 |
| 26 | Epson L810 | A4 | 14 | 350 | 14 200 | 4 |
| 31 | Epson M15140 | A4 | 6 | 450 | 7 599 | 4 |
| 32 | EPSON SureColor SC-F501 | A4 | 4 | 300 | 4 560 | 4 |

Зовнішній вид форм обліку принтерів повинен якомога точніше відтворювати вид представлених таблиць.

#### Вимоги до надійності

Надійне функціонування додатку повинне забезпечуватись наявністю блоку перевірки правильності введених даних. Це збереже від помилок, що виникають при виконанні рутино роботи користувачем.

Також необхідно забезпечити достатню стійкість сайту до некваліфікованих дій користувача, що досягається супроводом будь-яких операцій в додатку чіткими і зрозумілими повідомленнями.

#### Умови до експлуатації

Розроблений програмний продукт по потребуватиме яких-небудь видів обслуговування, для його використання достатньо буде навиків взаємодії користувача з ПК.

#### Часові характеристики

В інформаційно-пошукових системах вимоги до часових характеристик становлять до 10с на відповідь на запит; в системах керування виробничим процесом - до 1с, а в web-сайтах, що відвідуються раз на день, жорстких вимог не потребує.

# РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

## Розробка структури сайту і web-сторінок

В даному підрозділі пояснювальної записки описується такий етап розробки програми, як аналіз. Метою аналізу є максимально повний опис поставленої задачі, відповідно до технічного завдання.

Під час проходження етапу аналізу спочатку було визначено такі питання:

* Хто буде діючими суб'єктами?
* Хто вводитиме інформацію?
* Хто запрошуватиме?
* Чи буде хто-небудь ще взаємодіяти з програмою?
* Чи буде сама програма взаємодіяти з іншими програмами?

В даному випадку з додатком MyPrinters має працювати багато користувачів. Таким чином, вони вводять інформацію, підбирають для себе і переглядають.

Наступне, що потрібно було з'ясувати, це групу дій, які ініціюватиме діючий суб'єкт. Виходячи із технічного завдання, діючий суб'єкт – користувач буде виконувати такі дії:

* відкрити початкову сторінку сайту;
* переглянути властивості принтерів наявних в базі;
* вибрати один з переліку;
* вивести на головний екран вибрані принтери;
* додати/оновити/видалити принтер;
* відсортувати за певним параметром;
* подивитись загальну суму вибраних принтерів;
* перейти за навігаційним меню на інші вкладки;
* пошук потрібного принтера за будь-яким параметром;
* купити принтер ввівши свої дані в окреме вікно.

Таким чином, діаграму варіантів використання, яка отримується внаслідок наведеного переліку дій, можна подати так, як це показано на рисунку 2.1.

Вивести перелік принтерів

Переглянути властивості

Вибрати з переліку принтер

Додати/оновити/ видалити принтер

Перейти на інші вкладки

Запустити додаток

Пошук принтера

Покупка товару

Відсортувати, переглянути суму

Рисунок 2.1 – Діаграма варіантів використання для додатку MyPrinters

Наступний крок етапу аналізу це опис варіантів використання, де

необхідно детально описати всі варіанти використання.

**Запустити додаток:** ця дія, здавалося б, є дуже очевидною для того,

щоб про неї зовсім не згадувати, але все ж таки... Коли запускається додаток, у Local Storage автоматично записуються потрібні дані, які витягуються з бази даних.

**Пошук потрібного принтера за будь-яким параметром:** на головному екрані подій зверху відображається пошуковий показник. За його допомогою користувач може з легкістю найти потрібний йому принтер з будь-яким параметром.

**Навігаційне меню:** сценарій 1. У ньому представлено чотири події. Перша «Home» вертає користувача на головний екран. Друга «Catalog» дає доступ користувачу до бази даних всіх принтерів. Виводиться зноска категорій брендів, при чому, якщо нажати один з них, користувач переходить в іншу вкладку, де йому демонструється весь асортимент того бренду.

**Вивести на головний екран вибрані принтери:** якщо користувач вибрав необхідний пристрій він добавляє його в корзину, ця корзина відображається на головному екрані.

**Додати/оновити/видалити принтер:** сценарій 1. Додаток має виводити на екран короткий перелік принтерів, які вибрав користувач. Кожен рядок переліку складається з п’ятьох полів: назва принтеру, максимальний розмір паперу, швидкість друкування одного аркуша за хвилину, ресурс картриджа. Це може називатися екраном інтерфейсу користувача. В форму можна добавити новий принтер або оновити наявний, також видалити існуючий. Якщо користувач визначився з вибором, то можна завершити покупку і тоді програма сумує загальну ціну та відкриває вікно, де можна ввести поштові дані.

**Відсортувати за певним параметром:** цей же перелік можна відсортувати за потрібним користувачу параметром.

**Подивитись загальну суму вибраних принтерів:** додаток має виводити загальну суму вибраних принтерів, якщо нажати для цього кнопку.

**Купити принтер ввівши свої дані в окреме вікно:** web-розробка виводить вікно «Покупка пристроя», де просить користувача ввести дані для відправки посилки та оплатити покупку.

Варіант використання може складатися з декількох альтернативних сценаріїв. Вище було описано тільки основний сценарій для кожного варіанта використання. Це сценарій безпомилкової роботи, коли все йде гладко, і мета операції досягається так само, як це вимагається в ідеалі. Проте необхідно передбачити більш загальні варіанти подій у програмі. Як приклад можна навести випадок спроби створення принтеру не ввівши дані.

**Навігаційне меню:** сценарій 2. У ньому представлено чотири події. Третя «Menu» дозволяє користувачу переходити між вкладками, які надають докладнішу інформацію як оплатити/отримати посилку. Четверта «About us» відкриває нову вкладку, на якій надається інформація про магазин.

**Додати/оновити/видалити принтер:** сценарій 2. Додаток має виводити на екран короткий перелік принтерів, які вибрав користувач. Кожен рядок переліку складається з п’ятьох полів: назва принтеру, максимальний розмір паперу, швидкість друкування одного аркуша за хвилину, ресурс картриджа. В форму можна добавити новий принтер або оновити наявний, також видалити існуючий., але якщо в формі хоч одна комірка пуста, то користувачу виводиться повідомлення про помилку.

Також на етапі аналізу для моделювання варіантів використання, що мають декілька сценаріїв доцільно розробити UML-діаграми дій. Цей тип діаграм демонструє керівні потоки від одних дій до інших.

Навігаційне меню повідомлення про введення нового челленджу

ційне меню

«Home»

«About us»

«Catalog»

«Menu»

Перейти до вибраної вкладки

Перейти до вибраної вкладки

до вибраної вкладки

Перейти до вибраної вкладки

Перейти до вибраної вкладки

Перейти до вибраної вкладки

Рисунок 2.2 – Діаграма дій «Відповісти на питання»

На рисунку 2.2 показано варіант використання «Навігаційне меню», що включає обидва раніше описані сценарії. Вибір гілки діаграми залежить від того, що вибрав користувач. Один із запропонованих варіантів відкриє нову вкладку користувачу. Також на рисунку 2.3 зображено UML-діаграму дій варіанту використання «Додати/оновити/видалити принтер».

Вивести повідомлення про введення нового принтеру

Отримати назву та

інші дані

Помістити дані в базу даних принтерів

Вивести повідомлення про помилку

[Пуста комірка]

Рисунок 2.3 – Діаграма дій «Додати/оновити/видалити принтер»

## Створення та верстка сторінок сайту

Наступний етап розробки web-додатку – етап проектування. Даний етап поділяється на логічне та фізичне проектування. Логічне полягає у скриптуванні елементів дизайну, коли визначаються html-шаблони всіх сторінок (елементів) сайту, складові об'єктів, алгоритми методів, що описують поведінку об'єктів. Насамперед необхідно передбачити конструкції, з яких буде складатися програма. Для цього спочатку проаналізуємо перелік іменників з опису варіантів використання, потім внесемо деякі їх уточнення, визначимо їх функціональні елементи, перейдемо від дієслів до повідомлень, і, на завершення, розробимо діаграми конструкцій послідовностей.

**Аналіз переліку іменників з опису варіантів використання.** Розглянемо перелік усіх іменників, які візьмемо з опису варіантів використання:

1. Функція взаємодії з користувачем.
2. Навігаційне меню.
3. Вікно взаємодії з таблицею вибраних принтерів.
4. Перелік принтерів.
5. Властивості принтера.
6. Сума.
7. Сортування.
8. Додавання.
9. Оновлення.
10. Видалення.
11. Footer.

**Уточнення переліку іменників.** З різних причин багато іменників не зможуть стати конструкціями. Давайте проведемо відбір тільки тих іменників, які можуть претендувати на те, щоб бути функцією.

Було виписано назви рядків різних div-ів: перелік принтерів, властивості принтера. Іноді з рядків можуть виходити чудові конструкції, якщо вони складені або містять складні дані. Також хотілось би згадати: функція взаємодії з користувачем та навігаційне меню. Складні конструкції, які мають находитись на кожній вкладці web-додатку та безперервно виконувати своє завдання. Рядок суми, сортування та оновлення сторінки також можна винести як окремо конструкцію.

У додатку div Footer немає важких обчислень, та не виконує важливої ролі, але задля чистої верстки сайту та чистого коду, потрібно винести у відокремлений div. Функції додавання, апгрейду, та видалення не обов’язково створювати поруч, тому їх рознесемо по різних куточках сайту.

Отже, складемо перелік конструкцій, які було тільки що уточнено:

1. Голова вкладки.
2. Модальне вікно..
3. Навігаційне меню.
4. Головне фото сайту.
5. Вибрані принтери користувачем.
6. Контент сайту.
7. Вікно з властивостями принтеру.

**Визначення CSS-селекторів.** Всі конструктори html-сторінки мають мати красивий зовнішній вигляд, задля зручного користування, та розуміння сайту. Більшість div-ам надано свій клас, проте в роботі часто застосовується селектор: .container > ul. Також, як вище зазначеному тексті, у кожного конструктора свій клас, щоб мати змогу підступити до його редагування.

**Перехід від дієслів до повідомлень.** Тепер подивимося, що нам дають варіанти використання для з'ясування того, якими повідомленнями обмінюватимуться конструктори. Оскільки повідомлення – це, по суті справи, є викликом методу в об'єкті, то визначення повідомлень зводиться до визначення методів класу, що приймає те або інше повідомлення. Як і у випадку з іменниками, далеко не кожне дієслово стає кандидатом у повідомлення. Деякі з них, замість прийняття даних від користувачів, зв'язуються з такими операціями, як виведення інформації на екран, або з якими-небудь ще діями. Розглянемо опис варіанта використання функції «Вивести перелік вибраних принтерів»: «Програма виводить на екран перелік принтерів, кожен рядок переліку складається з деяких полів: ім'я принтеру, ціна за нього і т.д.». Під словом «додаток» ми насправді маємо на увазі вікно інтерфейсу користувача, отже, слово «виводить» означає, що об'єкт «Вікно інтерфейсу користувача» посилає повідомлення об'єкту «Перелік принтерів» (тобто викликає його метод). У повідомленні має міститися вказівка вивести самого себе на екран. Даний метод назвемо get(). Дієслово «складається» не належить ні до якого повідомлення. Він просто приблизно визначає склад рядка об'єкта «Перелік принтерів». Розглянемо опис варіанту використання «Додати новий принтер»: «У вікні має відобразитися повідомлення, у якому програма просить користувача ввести ім’я принтеру, ціну за нього та й інші дані. Ця інформація повинна заноситися в таблицю». Тут, екран інтерфейсу користувача повинен послати повідомлення класу «Екран введення принтеру», наказуючи йому вивести себе і отримати дані від користувача. Це повідомлення може бути викликом методу класу з іменем, наприклад getName(). Дієслова «просить» і «ввести» стосуються взаємодії класу «Екран введення принтеру» з користувачем. Вони не стають повідомленнями в об'єктному сенсі. Справді getName () виводить запрошення і записує відповіді користувача (ім'я принтеру, ціну за нього та інші дані). Дієслово «заноситися» означає, що об'єкт класу «Екран введення принтеру» посилає повідомлення об'єктові класу «Перелік принтерів». Можливо, як аргумент використовується новий об'єкт класу «Принтери». Об'єкт «Перелік принтерів» може потім вставити цей новий об'єкт у свій перелік. Ця функція може мати ім'я типу inserIntoTable(). Усі інші варіанти використання були проаналізовані аналогічним чином, це дозволило створити основу для зв'язування класів, тобто дозволило виконати фізичне проектування.

**Побудова діаграм конструкторів.** Знаючи, які класи буде включено в розроблювану програму і як вони пов'язані між собою, можна побудувати спрощену діаграму структурної схеми, яка зображуватиме взаємозв’язки між класами. Дану діаграму зображено на рисунку 2.4.

Головна вкладка

Сортування

Керуюче меню

Навігаційне меню

Виведення

Назад до головної

Створення

Сумування

Категорії

Редагування

Пошук

Оплата

Видалення

Доставка

Демонстрація

Повернення

Поради

Контакти

Про нас

Рисунок 2.4 – Діаграма структурної схеми додатку MyPrinters

## Розробка структури бази даних сайту

В даному підрозділі описується наступний етап розробки програмного забезпечення – еволюцію, що представляє собою етап послідовної реалізації та підключення бази даних до проекту. Сам процес проектування бази даних являє собою складний процес проектування відображення опису предметної області у схему внутрішньої моделі даних. Перебіг цього процесу є послідовністю більш простих процесів проектування менш складних відображень. Ця послідовність у процесі проектування весь час уточняється, вдосконалюється таким чином, щоб були визначені об'єкти, їх властивості та зв'язки, які будуть потрібні майбутнім користувачам системи.

Етапи проектування (створення) бази даних:

* **Визначення таблиць, що їх повинна містити база даних.**

Для цього проекту, було створено базу даних з безліччю таблицю, які називаються згідно назви принтеру. В кожній з цих таблиць міститься інформація про кожен принтер наявний у магазина.

* **Визначення необхідних у таблиці полів.**

Для того, щоб точніше описати характеристику принтерів у таблиці інсують такі поля:

* + Id
  + Name
  + Size
  + Speed
  + Resource
  + Price

Кожне з цих полів має бути заповнене, для необхідної інформації, яка надається користувачам, задля зручного та швидкого пошуку і вибору необхідної речі для спожитку.

* **Завдання індивідуального значення кожному полю.**

Кожне поле індивідуальне, та має відповідати своїй характеристиці. Тому, виходячи з вище сказаних слів, надаєм кожному полю своє значення. Кожен параметр повинен мати функцію NOT NULL, щоб користувач зміг витягнути звідти необхідне. Хлопці як десерт, тобі цього хочеться, но тобі цього не потрібно. Перше поле ID повинен бути, для зручного користування інтерфейса, йому надано тип INTEGER. Наступному параметру було надано значення STRING із-за необіхності внести дані назви принтера. Полю size також було надано властивість STRING, проте наступні три поля мають значення INTEGER.

* **Визначення зв'язків між таблицями.**

Зв'язків між таблицями не існує, бо вони не залежні як Україна. Вони не потребують, щоб хтось з ними зв'язкувався. А особливо спасав.

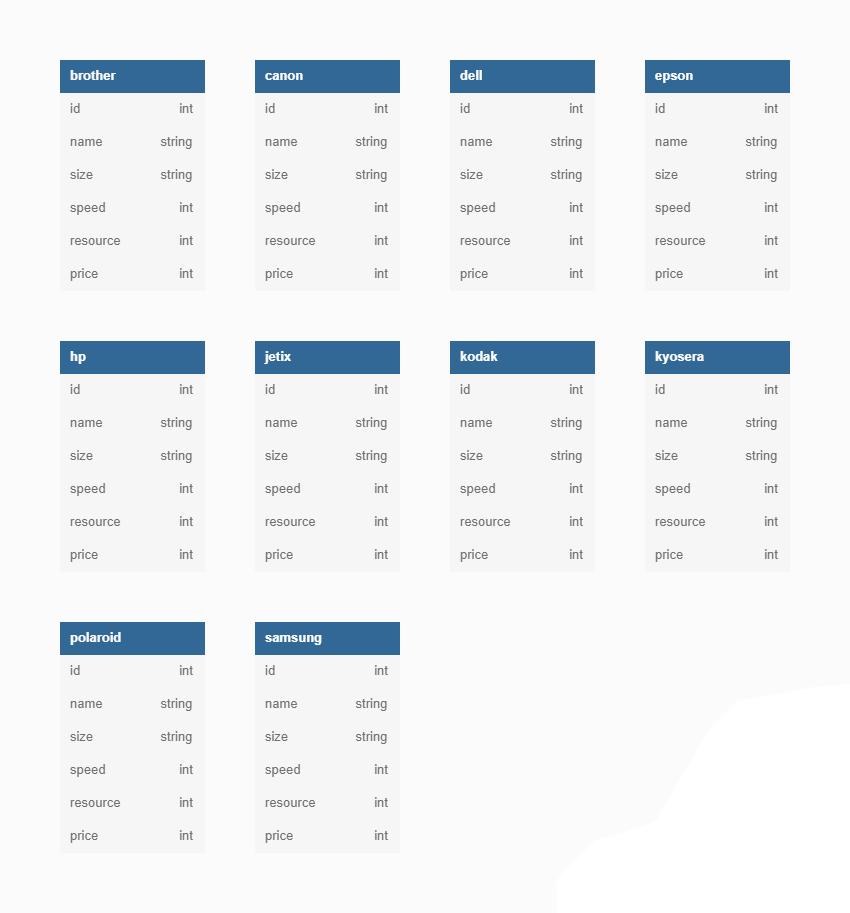


Рисунок 2.5 – ER-діаграма бази даних додатку MyPrinters

* **Додавання даних і створення запитів, форм, звітів та інших об'єктів бази даних.**

Базу заповнено всіма принтерами, які наявні у власника, та описано всі характеристики. Доступ до даних получаємо через окремі функції, такі як buildElementToPage(), search(), sort(), count(), deleteElement(), getArrayFromStorage(), editElement(), addElementToLocalStorage(), defaultValues().

Всі ці функції використовуються, щоб мати змогу підступити до баази даних та вміти нею орудувати.

## Програмування сайту

Маючи діаграми варіантів використання, детальні їх описи, діаграми таблиць, а також попередні плани щодо створення коду програми, можна приступити до написання коду програми.

Написання коду програми починається із написання заголовного (\*.html) файлу у якому необхідно визначити як інтерфейсні частини класів, так і подробиці їх реалізації. Оголошення в заголовному файлі – це загальнодоступна частина класів. Зовнішній вигляд функцій, які є розташовані в \*.css-файлах, називаються реалізацією і користувачам (стороннім програмістам) недоступні.

Написання заголовного файлу – це проміжний крок між етапами плануванням і звичайним кодуванням методів. Розглянемо вміст заголовного файлу «basic.html».

Більшість оголошень появляються безпосередньо з div-ів, створених за допомогою взятих з описів варіантів використання іменників, і відображаються на діаграмі класів. Тільки імена з багатослівних потрібно зробити однослівними. Наприклад, ім'я «Перелік принтерів» (Printer List) перетворюється в PrinterList.

Написання коду програми розглянутих методів має починатися тільки на даному етапі розроблення ПЗ і ні кроком раніше, тому що тільки зараз відомі ім'я кожного класу чи методу, їх призначення і навіть, можливо, передбачити аргументи, що їм передаються.

Завдання програмування полягає в тому, щоб об'єднати всі вживані технології воєдино в розроблений сайт і надати користувачеві зручний інтерфейс для використання функціональних можливостей веб-ресурсу.

Веб-додаток інтернет-магазин MyPrinters складно розроблений і не без допомоги Java Script. Всі основні функції роботи з базою даних лежать в методах та класах. Також, здавалось би на перший погляд, звичайне відкриття вкладки застосоване через мову скриптування.

В першу чергу, хотілось би звернути увагу на методи доступу до Local Storage. Дальше зазначенні функції це все реалізовують: buildElementToPage(), search(), sort(), count(), deleteElement(), getArrayFromStorage(), editElement(), addElementToLocalStorage(), defaultValues(). Саме як, продемонстровано в додатках від А

Метод buildElementToPage() записує нові дані в Local Storage. Код даної функції продемонстровано в додатку А.

Метод search() шукає дані в Local Storage.

Метод sort() сортує дані в певному порядку.

Метод count() сумує ціну принтерів.

Метод deleteElement() видаляє не потрібний принтер з Local Storage.

Метод getArrayFromStorage() переводить масив в JSON стрічку.

Метод editElement() редагує певний елемент в Local Storage.

Метод addElementToLocalStorage() додає новий елемент в Local Storage.

Метод defaultValues() автоматично записує деякі параметри в Local Storage.

Крім цих методів існують інші, шо є необіхдними для графічного керування сайту, такі як:

* + rebootSite() – перезагрузка сайту;
  + showModal() – показує модальне вікно;
  + showImageInModal() – показує зображення в модальному вікні;
  + closeModal() – приховує модальне вікно;
  + showCatalog() – показує каталог в навігаційному меню;
  + closeCatalog() – приховує каталог в навігаційному меню;
  + showMenu() – показує меню в навігаційному меню;
  + closeMenu() – приховує меню в навігаційному меню;
  + showImg() – показує фото в каталозі;
  + closeImg() – приховує фото;
  + showContent() – відкрває вікно з інформацією;
  + showComponent() – приховує вікно з інформацією;
  + checkImg() – перевіряє чи є зображення в модалці;
  + checkValues() – перевіряє чи є всі дані в модалці.

Ці всі методи забезпечують коректність сайту, та його функціональну можливість.

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.

Користувацькі скрипти в браузері - це програми, написані на JavaScript, що виконуються в браузері користувача при завантаженні сторінки.

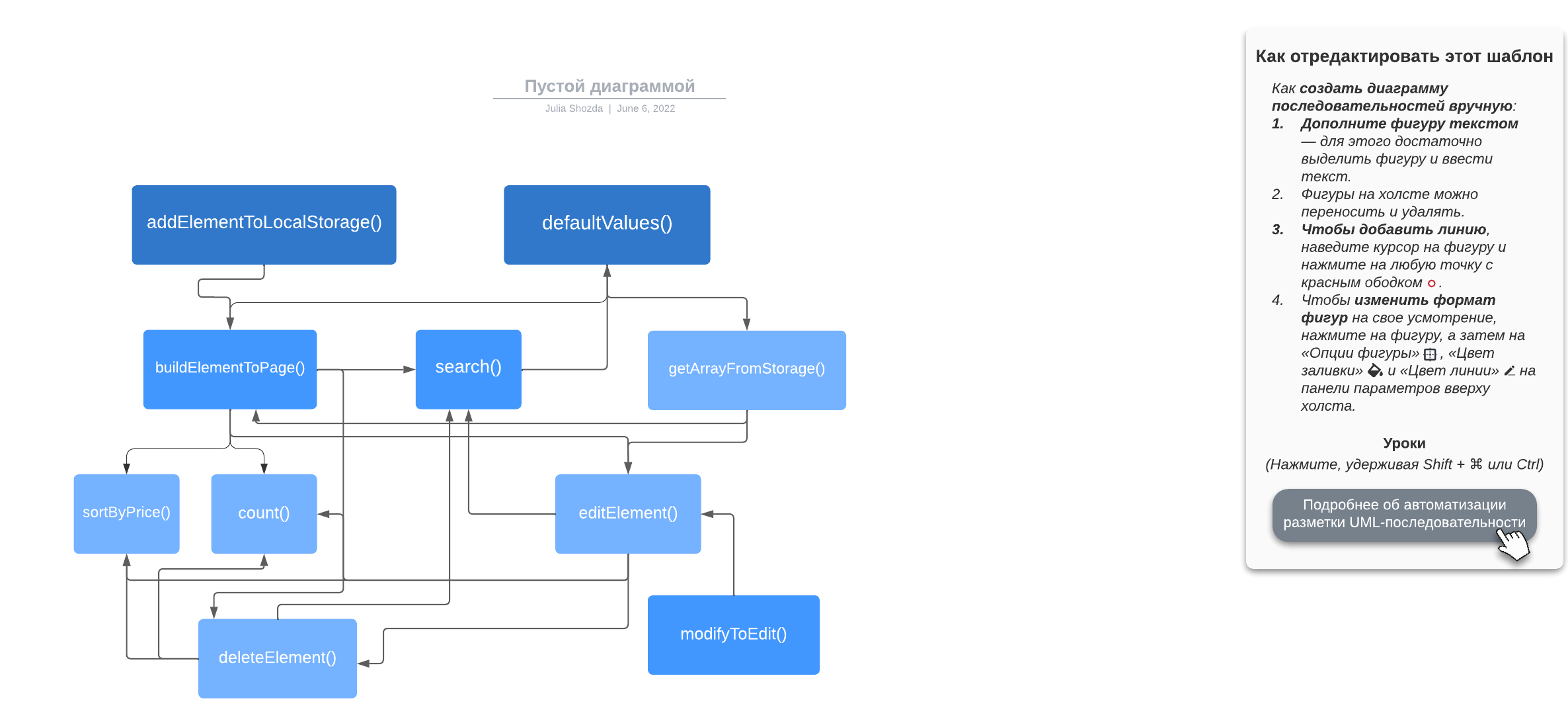


Рисунок 2.6 – UML-діаграма взаємозв’язків функцій додатку MyPrinters

|  |  |  |
| --- | --- | --- |
| № | Назва | Призначення |
| 1 | search() | шукає дані в Local Storage |
| 2 | sort() | сортує дані в певному порядку |
| 3 | count() | сумує ціну принтерів |
| 4 | deleteElement() | видаляє не потрібний принтер з Local Storage |
| 5 | getArrayFromStorage() | переводить масив в JSON стрічку |
| 6 | editElement() | редагує певний елемент в Local Storage |
| 7 | addElementToLocalStorage() | додає новий елемент в Local Storage |
| 8 | defaultValues() | автоматично записує деякі параметри в Local Storage. |
| 9 | modifyToEdit() | модифікує одні значення в інші |
| 10 | buildElementToPage() | записує нові дані в Local Storage |

Таблиця 2.1 – Таблиця взаємозв’язків функцій додатку MyPrinters

## Тестування сайту

В ході тестування програми було виявлено і виправлено незначні помилки. Запустивши програму на екрані монітора відображається головне меню програми, яке представлене в додатку Б. З його допомогою користувач може виконати такі дії:

* Створити новий принтер (Додаток В);

Після натиснення кнопки, програма перейде до наступного вікна, де необхідно ввести інформацію про принтер. Також, якщо користувач не вів хоча би одну із комірок, то вилізе повідомлення про помилку.

* Пошук необхідного принтеру.

Після введення символів додаток шукає необхідний принтер, та виводить нижче.

Нижче від головного меню відображається навігаційне меню. Це представлено в додатку Г. З його допомогою користувач може виконати такі дії:

* Вернутись на головну вкладку;

Користувач повертається на головний екран.

* Перейти до певної категорії принтерів (Додаток Д);

Користувач має змогу перейти до потрібної категорії принтерів, та прочитати про них інформацію та, якщо потрібно, вибрати до кошика.

* Перейти до підменю (Додаток Е);

Користувач може перейти до потрібної вкладки, та прочитати про це інформацію.

* Перейти до вкладки «Про нас».

Дальше йде головний екран де висвітлюються всі вибрані принтери. З його допомогою користувач може виконати такі дії:

* Редагувати принтер (Додаток Є);

Користувач може відредагувати деякі з параметрів принтера.

* Видаляти принтер (Додаток Є);

Користувач може видалити не потрібний йому принтер.

* Сортувати принтери (Додаток Є);

Користувач може відсортувати принтер за потрібним йому параметром.

* Підсумувати ціну принтерів (Додаток Є);

Користувач може підсумувати ціну принтерів.

* Оновити сторінку (Додаток Є).

Після йде асортимент магазину (Додаток Ж). Користувач може переглянути функції принтеру, та має можливість добавити до свого кошику.

# СПЕЦІАЛЬНИЙ РОЗДІЛ

## Розміщення сайту в Інтернеті

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

При виборі стороннього Web-серверу керуються двома основними критеріями: ціною і продуктивністю. У професійних розробників Web-сайтів вибір визначається можливостями замовника і стосунками з компаніями, що володіють Інтернетсерверами.

Нині з'являється усе більше Web-серверів, що здійснюють безкоштовний хостінг.

Такі сервери не вимагають оплати за підтримку Web-сайтів клієнтів та існують за рахунок рекламодавців. Зареєстрованому клієнту дається обмежений, але достатній простір на диску серверу — від 5 до 50 Мбайт. На ньому клієнт може розмістити як невеликий, так і солідний сайт. Єдине зобов'язання, що бере на себе клієнт безкоштовного Web-серверу, це розміщення на своєму сайті *банерів*— маленьких помітних зображень (часто анімованих), призначених для реклами.

## Інструкція з обслуговування та наповнення сайту

У даному підрозділі описується інструкція експлуатації адміністративної частини сайту, основні функції та меню сайту, інструкція з редагування існуючого та створення нового контенту. Також визначається мінімальна конфігурація серверу для хостингу сайту.

Наповнення сайту, контент - це одна з найважливіших складових сайту; це весь обсяг інформації, яка міститься на сайті. Основна мета розміщення інформації полягає в тому, що б надати можливість користувачеві ознайомитися з товарами і послугами, а також для підтримки інтересу ресурсу. Вся інформація, подана на сайті, повинна бути правильно підібрана, достовірна та актуальна, грамотно структурована і розбита на розділи.

Види контенту і вимоги до нього:

* Текст. Всі тексти на сайті повинні в повному обсязі і коротко викладати необхідну інформацію. Слід створювати текст максимально інформативним при мінімальному обсязі. Часто для текстової інформації на сайті є необхідність графічного виділення важливих моментів, на які користувач повинен звернути увагу.
* Графічні зображення (фотографії, малюнки). Всі графічні зображення, що розміщуються на сайті, повинні виконувати функцію ілюстрації до текстової інформації, а не бути її заміною.
* Табличні дані. Без необхідності, сайт не слід заповнювати великою кількістю таблиць. Їх слід використовувати для цифрових даних (прайс-листи) або для даних які постійно оновлюються (ціни, тарифи).
* Посилання. До них пред'являються прості вимоги: посилання повинні бути виділені графічно і обов'язково вести до потрібної інформації.

Контент - це складова частина сайту, яка працює на імідж і підтримує постійний інтерес до сайту. Якісний і унікальний контент створює позитивну думку у відвідувачів.

Обслуговування сайту включає в себе два основних напрямки по роботі з сайтом: технічна підтримку та супровід сайту.

До технічної підтримки ресурсу слід віднести повний комплекс послуг з технічної підтримки сайту та забезпечення безперебійної роботи ресурсу. Підтримка працездатності скриптів, та багато інших робіт необхідні для ефективної роботи ресурсу. Серед таких робіт можна виділити оновлення або доповнення програмного забезпечення для працездатності сайту, періодичне резервне копіювання, доповнення або зміна функціоналу сайту.

Супроводу сайту - це роботи, які пов'язані з контентом: періодичне доповнення на сайті розділу "новини", "події" і тд, оновлення табличних даних та зміна зображень на більш актуальні.

## Інструкція з популяризації та підтримки сайту

Популяризація та підтримка сайту - це комплекс дій, спрямованих на підвищення позиції сайту у видачі пошукових систем і, відповідно, на збільшення відвідуваності.

До основних етапів розкрутки сайту відносять:

Аудит - комплексне дослідження сайту, що проводиться для виявлення можливих невідповідностей з внутрішніми переходами і посиланнями, з необхідністю зміни або доповнення контенту. А так само для діагностики зручності користування і можливості швидко знайти необхідну інформацію.

Внутрішня оптимізація сайту - виправленням помилок html-коду, змінами контенту, зміною внутрішніх переходів на сайті. Внутрішня оптимізація робить сайт «видимим» для пошукових систем.

Пошукова оптимізація сайту - етап, що включає в себе: складання семантичного ядра, впровадження в тексти необхідних слів і словосполучень, відповідних семантичному ядру, створення метатегів для кожної сторінки сайту, та інше.

Збільшення кількості поклилань на сайт з інших ресурсів, або збільшення цитованості сайту, значно поліпшує позиції у видачі пошукових машин.

Ефективно побудована стратегія з пошукової оптимізації сайту веде до зростання трафіку. Розкрутка сайту - це один з найважливіших кроків для просування вашого бізнесу в інтернеті.

ВИСНОВКИ

В даній курсовій роботі було описано етапи розробки програми MyPrinters, яка призначена для підвищення ефективності заробітку людини в буденному житті, за рахунок сайту, що збільшує продуктивність кожноденних замовлень.

Під час виконання курсової роботи було закріплено і поглиблено знання, одержанні за час вивчення предмету «Web-технології та web-дизайн», більш детально вивчено мови програмування HTML, CSS, Java Script.

Для розробки даного програмного забезпечення використовувались принципи і концепції об’єктно-орієнтованого програмування. Використане середовище програмування Visual Studio Code, його особливі можливості і переваги, що дозволило швидко конструювати програмний засіб. Також під час розробки використано стандартну бібліотеку html, зокрема класи, що дозволило в значній мірі пришвидшити розробку даної програми.

ПЕРЕЛІК ПОСИЛАНЬ

1. Правильність функцій [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com> - Дата доступу: 13.05.2022
2. Наповнення сайту [Електронний ресурс] – Режим доступу до ресурсу: https://www.can-all.com/6-steps-to-conquer-the-internet/site-content.html - Дата доступу: 13.05.2022
3. Обслуговування сайту [Електронний ресурс] – Режим доступу до ресурсу: https://www.can-all.com/6-steps-to-conquer-the-internet/support-site.html - Дата доступу: 13.05.2022
4. Популяризація сайту [Електронний ресурс] – Режим доступу до ресурсу: https://www.can-all.com/6-steps-to-conquer-the-internet/site-promotion.html- Дата доступу: 13.05.2022

Додаток А  
Лістинг методу «buildElementToPage»

function buildElementToPage(id, elem) {

const element = document.createElement('div');

element.classList.add('element');

element.insertAdjacentHTML('afterbegin', `

<div class="compContentSelect">

<img src="img/${elem.img}" class="compImgSel">

<p class="compTitle">${elem.name}</p>

<p class="compLine">Maximum paper size: <span>${elem.size}</span> </p>

<p class="compLine">Color print speed, ppm: <span>${elem.speed}</span> </p>

<p class="compLine">Cartridge resource, page: <span>${elem.resource }</span> </p>

<p class="compLine">Price, UAH: <span>${elem.price}</span> </p>

<div class="contBut">

<button id="editElement" class="contEdit" onclick="modifyToEdit(${id})">Edit</button><span> </span>

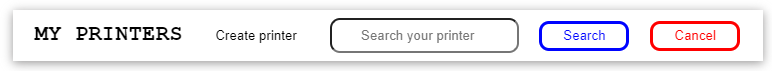
<button id="deleteElement" class="contDelete" onclick="deleteElement(${id})">Delete</button>

</div>

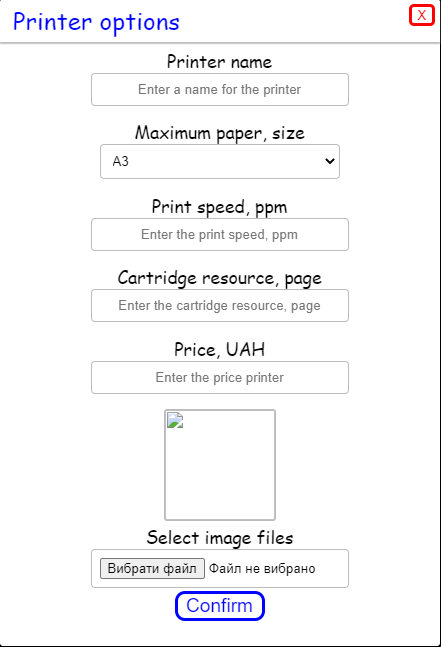
</div>`);

document.getElementsByClassName("displayzone")[0].appendChild(element);

}

Додаток Б  
Тестування додатку «Головне меню»

Додаток В  
Тестування додатку «Створення принтеру»



 Додаток Д  
Тестування додатку «Навігаційне меню»

Додаток Е  
Тестування додатку «Категорії принтеру»



Додаток Е  
Тестування додатку «Підменю»

Додаток Є  
Тестування додатку «Головний екран»



Додаток Ж  
Тестування додатку «Асортимент»

Додаток З  
Лістинг файлу «basic.html»

<!DOCTYPE html>

<html lang="uk">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>MY PRITNERS</title>

<link rel="stylesheet" href="css/basis.css">

<link rel="stylesheet" href="css/buttons.css">

<link rel="stylesheet" href="css/modal.css">

<link rel="stylesheet" href="css/component.css">

</head>

<body>

<div class="empty"></div>

<!-- Header -->

<div class="Header">

<div>

<span class="headerFat">MY PRINTERS</span>

<button type="button" class="butCreateHeader" onclick=showModal()>

Create printer

</button>

<label for="headerEdit">

<input type="search" id="headerEdit" class="headerEdit" placeholder="Search your printer">

</label>

<button class="butSearchHeader" id="butSearchHeader" onclick="search()">Search</button>

<button class="butCancelHeader" onclick="rebootSite()">Cancel</button>

</div> </div>

<!-- Menu -->

<div class="menu" id="menu">

<nav>

<ul class="menuNavbar">

<li onmouseover=closeMenu()><a href="basic.html" onmouseover=closeCatalog()>Home</a></li>

<li onmouseover=closeMenu()><a onmouseover=showCatalog()>Catalog</a></li>

<li onmouseover=closeCatalog()><a onmouseover=showMenu()>Menu</a></li>

<li onmouseover=closeMenu()><a href="html/aboutUs.html" onmouseover=closeCatalog()>About us</a></li>

</ul>

</nav>

</div>

<div class="menuDropDownCatalog" id="menuDropDownCatalog" onmouseover=showCatalog() onmouseout=closeCatalog()>

<div class="menuDropDownCatalogDiv">

<div>

<li onmouseover=showImg(1) onmouseout=closeImg(1)><a href="html/brother.html" class=>Brother</a></li>

<li onmouseover=showImg(2) onmouseout=closeImg(2)><a href="html/canon.html">Canon</a></li>

<li onmouseover=showImg(3) onmouseout=closeImg(3)><a href="html/dell.html">Dell</a></li>

<li onmouseover=showImg(4) onmouseout=closeImg(4)><a href="html/epson.html">Epson</a></li>

<li onmouseover=showImg(5) onmouseout=closeImg(5)><a href="html/hp.html">HP</a></li>

<li onmouseover=showImg(6) onmouseout=closeImg(6)><a href="html/kodak.html">Kodak</a></li>

<li onmouseover=showImg(7) onmouseout=closeImg(7)><a href="html/kyocera.html">Kyocera</a></li>

<li onmouseover=showImg(8) onmouseout=closeImg(8)><a href="html/panasonic.html">Jetix</a></li>

<li onmouseover=showImg(9) onmouseout=closeImg(9)><a href="html/palaroid.html">Polaroid</a></li>

<li onmouseover=showImg(10) onmouseout=closeImg(10)><a href="html/smartKids.html">Samsung</a></li>

</div> <div>

<img src="img/homePrinter.png" class="always" id="always" style="margin: 10px; max-width: 300px;">

<img src="img/Brother/BROTHER HL-1110R.jpg" id="brother" class="brotherImg">

<img src="img/Canon/Canon Laser T140.jpg" id="canon" class="canonImg">

<img src="img/Dell/Dell B2360Dn Mono.jpg" id="dell" class="dellImg">

<img src="img/Epson/EPSON SureColor SC-F501.jpg" id="epson" class="epsonImg">

<img src="img/HP/HP Ink Tank 410 with Wi-Fi.jpg" id="hp" class="hpImg">

<img src="img/Kodak/Kodak Photo Printer Dock PD450W.jpg" id="kodak" class="kodakImg">

<img src="img/Kyocera/Kyocera M2040dn.jpg" id="kyocera" class="kyoceraImg">

<img src="img/Jetix/JETIX Mini Printer Student.jpg" id="jetix" class="jetixImg">

<img src="img/Polaroid/Polaroid Lab.jpg" id="polaroid" class="polaroidImg"

style="margin: 10px; display: none; max-width: 280px;">

<img src="img/Samsung/Samsung Xpress M2070W.jpg" id="samsung" class="samsungImg">

</div>

</div>

</div>

<div class="menuDropDownMenu" id="menuDropDownMenu" onmouseover=showMenu() onmouseout=closeMenu()>

<div class="menuDropDownMenuDiv">

<div class="menuIkon">

<a href="html/payment.html">

<img src="img/ikon/menuPayment.png" class="menuIkonImg">

<div>Payment</div>

</a>

</div>

<div class="menuIkon">

<a href="html/delivery.html">

<img src="img/ikon/menuDelivery.png" class="menuIkonImg">

<div>Delivery</div>

</a>

</div>

<div class="menuIkon">

<a href="html/return.html">

<img src="img/ikon/menuReturn.png" class="menuIkonImg" height="1500px">

<div>Return</div>

</a>

</div>

Додаток И  
Лістинг файлу «basic.js»

// Header

function rebootSite() {

location.reload();}

function search() {

document.getElementsByClassName("displayzone")[0].innerHTML = '';

let toSearchArr = getArrayFromStorage();

let toSearchStr = document.querySelector("#headerEdit").value;

let toSearchStrLow = toSearchStr.toLowerCase();

let checkValueSearch = new RegExp(`${toSearchStrLow}`);

let isFounded = false;

for (let i = 0; i < toSearchArr.length; i++) {

let item = toSearchArr[i];

let elemN = item.name.toLowerCase();

let elemSiz = item.size.toLowerCase();

let elemSp = item.speed;

let elemRes = item.resource;

let elemP = item.price;

if (checkValueSearch.test(elemN) || checkValueSearch.test(elemSiz) || checkValueSearch.test(elemSp) || checkValueSearch.test(elemRes) || checkValueSearch.test(elemP)) {

buildElementToPage(item.id, item);

isFounded = true; }}

if (!isFounded) {

document.getElementsByClassName("displayzone")[0].innerHTML = alert("is not founded");}}

//Modal

function showImageInModal(imageElement) {

let image = document.getElementById("modalFile");

image.src = URL.createObjectURL(imageElement.files[0]);}

function showModal() {

document.getElementById("modal").style.display = "block";}

function closeModal() {

document.getElementById("modal").style.display = "none";}

// My selected

function buildElementToPage(id, elem) {

const element = document.createElement('div');

element.classList.add('element');

element.insertAdjacentHTML('afterbegin', `

<div class="compContentSelect">

<img src="img/${elem.img}" class="compImgSel">

<p class="compTitle">${elem.name}</p>

<p class="compLine">Maximum paper size: <span>${elem.size}</span> </p>

<p class="compLine">Color print speed, ppm: <span>${elem.speed}</span> </p>

<p class="compLine">Cartridge resource, page: <span>${elem.resource }</span> </p>

<p class="compLine">Price, UAH: <span>${elem.price}</span> </p>

<div class="contBut">

<button id="editElement" class="contEdit" onclick="modifyToEdit(${id})">Edit</button><span> </span>

<button id="deleteElement" class="contDelete" onclick="deleteElement(${id})">Delete</button>

</div>

</div>`);

document.getElementsByClassName("displayzone")[0].appendChild(element);}

let keyNumbers = Object.keys(localStorage).length;

for (let k = 0; k < keyNumbers; k++) {

let keyName = localStorage.key(k);

let row = JSON.parse(localStorage.getItem(keyName));

buildElementToPage(keyName, row);}

// сортування по ціні

function sortByPrice() {

let checkBox = document.getElementById("sort").value;

if (checkBox === "Price") {

let sortArr = getArrayFromStorage()

function byField(field) {

return (a, b) => +a[field] > +b[field] ? 1 : -1;}

// параметр для сортування

sortArr.sort(byField(checkBox));

document.getElementsByClassName("displayzone")[0].innerHTML = ''

for (let n = 0; n < sortArr.length; n++) {

let temp = sortArr[n];

buildElementToPage(temp.id, temp)}} else {

setTimeout(location.reload(), 1000)}}

function count() {

let priceArr = Object.keys(localStorage).length;

let result = 0

for (let i = 0; i < priceArr; i++) {

let keyID = localStorage.key(i)

let strObj = JSON.parse(localStorage.getItem(keyID))

result += Number(strObj.price);

document.getElementById("result").innerHTML = `${result} UAN`}}

sort.addEventListener('click', sortByPrice)

butSearchHeader.addEventListener('click', search)

//Menu

function showCatalog() {

document.getElementById("menuDropDownCatalog").style.display = "block";}

function closeCatalog() {

document.getElementById("menuDropDownCatalog").style.display = "none";}

function showMenu() {

document.getElementById("menuDropDownMenu").style.display = "block";}

function closeMenu() {

document.getElementById("menuDropDownMenu").style.display = "none";}

function showImg(key) {

switch (key) {

case 1:

document.getElementById("brother").style.display = "block";

document.getElementById("always").style.display = "none";

break;

case 2:

document.getElementById("canon").style.display = "block";

document.getElementById("always").style.display = "none";

break;

case 3:

document.getElementById("dell").style.display = "block";

document.getElementById("always").style.display = "none";

break;

case 4:

document.getElementById("epson").style.display = "block";

document.getElementById("always").style.display = "none";

break;

case 5:

document.getElementById("hp").style.display = "block";

document.getElementById("always").style.display = "none";

break;

case 6:

document.getElementById("kodak").style.display = "block";

document.getElementById("always").style.display = "none";

break;

case 7:

document.getElementById("kyocera").style.display = "block";

document.getElementById("always").style.display = "none";

break;

case 8:

document.getElementById("jetix").style.display = "block";

document.getElementById("always").style.display = "none";

break;

case 9:

document.getElementById("polaroid").style.display = "block";

document.getElementById("always").style.display = "none";

break;

case 10:

document.getElementById("samsung").style.display = "block";

document.getElementById("always").style.display = "none";

break;}}

function closeImg(key) {

switch (key) {

case 1:

document.getElementById("brother").style.display = "none";

document.getElementById("always").style.display = "block";

break;

case 2:

document.getElementById("canon").style.display = "none";

document.getElementById("always").style.display = "block";

break;

case 3:

document.getElementById("dell").style.display = "none";

document.getElementById("always").style.display = "block";

break;

case 4:

document.getElementById("epson").style.display = "none";

document.getElementById("always").style.display = "block";

break;

case 5:

document.getElementById("hp").style.display = "none";

document.getElementById("always").style.display = "block";

break;

case 6:

document.getElementById("kodak").style.display = "none";

document.getElementById("always").style.display = "block";

break;

case 7:

document.getElementById("kyocera").style.display = "none";

document.getElementById("always").style.display = "block";

break;

case 8:

document.getElementById("jetix").style.display = "none";

document.getElementById("always").style.display = "block";

break;

case 9:

document.getElementById("polaroid").style.display = "none";

document.getElementById("always").style.display = "block";

break;

case 10:

document.getElementById("samsung").style.display = "none";

document.getElementById("always").style.display = "block";

break;}

}

// Content

function showContent() {

document.getElementById("compTable").style.display = "none";}

function showComponent(key) {

switch (key) {

case 1:

document.getElementById("compTable").style.display = "block";

break;

case 2:

document.getElementById("compTable").style.display = "block";

break;

case 3:

document.getElementById("compTable").style.display = "block";

break;

case 4:

document.getElementById("compTable").style.display = "block";

break;

case 5:

document.getElementById("compTable").style.display = "block";

break;

case 6:

document.getElementById("compTable").style.display = "block";

break;

case 7:

document.getElementById("compTable").style.display = "block";

break;

case 8:

document.getElementById("compTable").style.display = "block";

break;

case 9:

document.getElementById("compTable").style.display = "block";

break;

case 10:

document.getElementById("compTable").style.display = "block";

break;

}

}

Додаток І  
Лістинг файлу «audit.js»

function checkImg() {

let valid = true;

if (document.getElementById("img").value) { valid } else {

alert("The image was not selected!");

valid = false; }

return valid; }

function checkValues() {

let valid = true;

if (document.getElementById("name").value) { valid } else {

alert("The name was not entered!")

valid = false;}

if (document.getElementById("size").value) { valid } else {

alert("The size was not entered!")

valid = false;}

if (document.getElementById("speed").value) { valid } else {

alert("The speed was not entered!")

valid = false;}

if (document.getElementById("resource").value) { valid } else {

alert("The resource was not entered!")

valid = false;}

if (document.getElementById("price").value) { valid } else {

alert("The price was not entered!")

valid = false;}

return valid;

}

Додаток Й  
Лістинг файлу «localStorage.js»

//елементи, наявні по дефолту в локас сторедж

function defaultValues() {

let printerDate = [

{ name: "Kodak PD-450", size: "A6", speed: 11, resource: 6, price: 1962, img: "Kodak/Kodak PD450.jpg" },

{ name: "Kyocera M2040dn", size: "A3", speed: 40, resource: 850, price: 22194, img: "Kyocera/Kyocera M2040dn.jpg" },

{ name: "Dell B2360Dn Mono", size: "A4", speed: 16, resource: 200, price: 7597, img: "Dell/Dell B2360Dn Mono.jpg" },

{ name: "Sprocket Photo Print", size: "A6", speed: 1, resource: 5, price: 4945, img: "HP/HP Sprocket Photo Print Black Noir.jpg" },

{ name: "Samsung SL-M2026", size: "A4", speed: 20, resource: 120, price: 2295, img: "Samsung/Samsung SL-M2026.jpg" }]

localStorage.clear();

for (let i = 0; i < printerDate.length; i++) {

let item = printerDate[i];

let itemToStorage = JSON.stringify(item);

localStorage.setItem(`${i}`, itemToStorage);}

location.reload();}

// додати елемент в локал сторедж

function addElementToLocalStorage() {

if (checkValues() && checkImg()) {

let keyArr = [];

for (let i = 0; i < localStorage.length; i++) {

let key = Number(localStorage.key(i));

keyArr[i] = key;}

const freeKey = Math.max(...keyArr) + 1;

let filename = document.getElementById("img").value.replace(/C:\\fakepath\\/, '');

//Build new element.

const newElement = {};

newElement.name = document.getElementById("name").value;

newElement.size = document.getElementById("size").value;

newElement.speed = document.getElementById("speed").value;

newElement.resource = document.getElementById("resource").value;

newElement.price = document.getElementById("price").value;

newElement.img = filename;

//Convert to JSON.

let newObj = JSON.stringify(newElement);

localStorage.setItem(`${freeKey}`, newObj);

closeModal();

setTimeout(location.reload(), 1000);}}

//редагувати в локал сторедж

function editElement(id) {

if (checkValues()) {

let edElem = JSON.parse(localStorage.getItem(id));

edElem.name = document.getElementById("name").value;

edElem.size = document.getElementById("size").value;

edElem.speed = document.getElementById("speed").value;

edElem.resource = document.getElementById("resource").value;

edElem.price = document.getElementById("price").value;

if (document.getElementById("img").value) {

let filename = document.getElementById("img").value.replace(/C:\\fakepath\\/, '');

edElem.img = filename;}

let strEdit = JSON.stringify(edElem);

localStorage.setItem(`${id}`, strEdit);

closeModal();

setTimeout(location.reload(), 1000);

} else closeModal();}

// масив елементів з локалу

function getArrayFromStorage() {

let keynumber = Object.keys(localStorage).length;

let element = {};

let inputArray = [];

for (let i = 0; i < keynumber; i++) {

let keyId = localStorage.key(i);

let item = JSON.parse(localStorage.getItem(keyId));

element = {};

element.id = keyId;

element.name = item.name;

element.size = item.size;

element.speed = item.speed;

element.resource = item.resource;

element.price = item.price;

element.img = item.img;

inputArray.push(element);}

return inputArray;}

//видаляти

function deleteElement(id) {

localStorage.removeItem(id)

setTimeout(location.reload(), 1000)}

//зміна для редагування

function modifyToEdit(id) {

document.getElementsByClassName("modalTitle")[0].innerText = "Edit printer";

document.getElementById("butConfirmModal").innerText = "Update";

document.getElementById("butConfirmModal").setAttribute("onclick", `editElement(${id})`);

let edElem = JSON.parse(localStorage.getItem(id));

document.getElementById("name").value = edElem.name;

document.getElementById("size").value = edElem.size;

document.getElementById("speed").value = edElem.speed;

document.getElementById("resource").value = edElem.resource;

document.getElementById("price").value = edElem.price;

document.getElementById("img").setAttribute("src", `img/${edElem.img}`);

showModal();

}

Додаток К  
Лістинг файлу «basic.css»

body {

font-family: Bradley Hand, cursive;

overflow: auto;

text-align: center;

margin: 10px;}

/\* Header \*/

.empty {

display: block;

width: 100%;

height: 40px;}

.Header {

display: inline-block;

position: fixed;

top: 10px;

left: 0;

right: 0;}

.Header div{

box-shadow: 0 2px 10px #969595;

background-color: rgb(255, 255, 255);

width: 750px;

margin: auto;}

.headerFat {

font-size: 140%;

font-weight: bold;

margin: auto auto;

margin: 8px;

font-family: 'Courier New', Courier, monospace;}

.headerCreate {

margin: 8px;

padding-bottom: 3px;}

.headerEdit {

border-radius: 10px;

padding: 8px;

text-align: center;

margin: 8px;}

/\* Menu \*/

.menu {

margin-left: -50px;

text-align: left;}

.menuNavbar {

display: flex;

flex-direction: row;}

.menuNavbar \* {

list-style-type: none;

margin: 0;

padding: 0;}

.menuNavbar li {

display: block;

height: 25px;

position: relative;}

.menuNavbar li:hover {

background-color: CornflowerBlue;}

.menuNavbar li a {

display: table-cell;

min-width: 80px;

text-align: center;

color: black;

text-decoration: none;}

.menuNavbar li a:hover {

color: white;

cursor: pointer;}

.menuDropDownMenu {

display: none;

position: fixed;

left: 30px;

right: 30px;

text-align: center;}

.menuDropDownMenuDiv {

min-width: 80%;

height: 150px;

box-shadow: 2px 2px 10px #969595;

background-color: white;

display: flex;

flex-flow: row wrap;

justify-content: space-around;

text-align: center;

position: relative;}

.menuDropDownMenuDiv div a {

text-align: center;

color: black;

text-decoration: none;}

.menuDropDownMenuDiv div:hover {

background-color: CornflowerBlue;

cursor: pointer;

color: white;}

.menuIkon {

text-align: center;

width: 200px;

object-fit: cover;}

.menuIkonImg {

min-width: 100px;

height: 100px;

margin: 10px;}

.menuDropDownCatalog {

display: none;

position: fixed;

left: 30px;

text-align: center;}

.menuDropDownCatalogDiv {

min-width: 500px;

height: auto;

box-shadow: 2px 2px 10px #969595;

background-color: white;

display: flex;

justify-content: space-between;

text-align: left;

position: relative;

align-items: center;}

.menuDropDownCatalogDiv \* {

list-style-type: none;

margin: 0;

padding: 0;}

.menuDropDownCatalogDiv div li:hover a {

background-color: CornflowerBlue;

cursor: pointer;

color: white;}

.menuDropDownCatalogDiv div li a {

text-align: center;

color: black;

text-decoration: none;}

.menuDropDownCatalogDiv div li {

margin: 10px}

.menuDropDownCatalogDiv div li:hover a {

background-color: CornflowerBlue;

cursor: pointer;

color: white;}

.brotherImg,

.canonImg,

.dellImg,

.epsonImg,

.hpImg,

.kodakImg,

.kyoceraImg,

.jetixImg,

.polaroidImg,

.samsungImg {

margin: 10px;

display: none;

max-width: 300px;}

/\* Main foto \*/

.mainFotoImg {

width: 100%;

object-fit: cover;}

/\* Selected \*/

.manage {

display: flex;

flex-flow: row wrap;

justify-content: space-evenly;

align-items: baseline;

font-size: 20px;}

.btnManage {

background-color: white;

color: rgb(0, 4, 255);

border: solid rgb(0, 4, 255);

border-radius: 5px;

height: 30px;

width: 80px;}

.btnManage:hover {

background-color: rgb(0, 4, 255);

color: white;}

/\* Footer \*/

footer {

display: inline-block;

box-shadow: 0 2px 10px #969595;

width: 721.5px;

height: 51px;}

/\* addition \*/

.content {

text-align: left;}

.h1 {

background-color: rgb(0, 4, 255);

color: white;

padding: 8px;

box-shadow: 10px 10px 10px #969595;}

.imgLeft {

float: left;

margin: 5px 10px 0 0;}

.imgRight {

float: right;

margin: 5px 0 0 5px;}

.manage,

.h3 {

background-color: CornflowerBlue;

color: white;

padding: 8px;}

.map {

width: 900px;

height: 520px;

display: flex;}

.ikon {

width: 100px;

border-radius: 100px;

margin-right: 30px;}

.container {

align-items: center;

display: flex;}

Додаток Л  
Лістинг файлу «button.css»

.butCreateHeader {

color: black;

border: none;

background-color: white;

padding: 0 20px;}

.butCreateHeader:hover {

cursor: pointer;}

.butSearchHeader {

background-color: white;

color: rgb(0, 4, 255);

border: solid rgb(0, 4, 255);

border-radius: 10px;

height: 30px;

width: 90px;

margin: 8px;}

.butCancelHeader {

background-color: white;

color: rgb(255, 0, 0);

border: solid rgb(255, 0, 0);

border-radius: 10px;

height: 30px;

width: 90px;

margin: 8px;}

.butSearchHeader:hover {

background-color: rgb(0, 4, 255);

color: white;}

.butCancelHeader:hover {

background-color: rgb(255, 0, 0);

color: white;}

.butConfirmModal {

background-color: white;

color: rgb(0, 4, 255);

border: solid rgb(0, 4, 255);

border-radius: 10px;

height: 30px;

width: 90px;

font-size: 120%;

margin-bottom: 5px;}

.butCloseModal {

background-color: white;

color: rgb(255, 0, 0);

border: solid rgb(255, 0, 0);

border-radius: 5px;

height: 20px;

width: 26px;

padding-bottom: 15px;}

.butConfirmModal:hover {

background-color: rgb(0, 4, 255);

color: white;}

.butCloseModal:hover {

background-color: rgb(255, 0, 0);

color: white;}

.contBut {

display: flex;

flex-flow: row wrap;

justify-content: space-evenly;

align-items: baseline;}

.contEdit {

background-color: white;

color: rgb(0, 4, 255);

border: solid rgb(0, 4, 255);

border-radius: 5px;

height: 30px;

width: 80px;}

.contDelete {

background-color: white;

color: rgb(255, 0, 0);

border: solid rgb(255, 0, 0);

border-radius: 5px;

height: 30px;

width: 80px;

margin-left: 13%;

}

.contEdit:hover {

background-color: rgb(0, 4, 255);

color: white;

}

.contDelete:hover {

background-color: rgb(255, 0, 0);

color: white;

}

.butContImg {

padding: 10px 5px;

border-radius: 10px;

background-color: white;

border: white 2px;

margin: 10px;

}

.butContImg:hover {

box-shadow: 2px 2px 20px #969595;

}

Додаток М  
Лістинг файлу «modal.css»

.overlay {

background-color: black;

width: 100%;

height: 110%;

opacity: 50%;

top: 0%;

left: 0%;

position: fixed;

}

.modalWindow {

margin: auto;

width: 400px;

height: auto;

background-color: rgb(255, 255, 255);

border: solid black 2px;

border-radius: 5px;

padding: 20px;

position: relative;

}

.modal {

display: none;

position: fixed;

left: 0;

right: 0;

top: 1%;

}

.modalTitle {

color: rgb(0, 4, 255);

font-size: 150%;

margin-left: 2%;

}

.modalHead {

display: flex;

justify-content: space-between;

box-shadow: 0 2px 1px #c0c0c0;

margin: -20px -20px 25px;

padding: 5px;

}

.modalBody {

font-size: 110%;

margin-top: -18px;

}

.modalEdit {

border-radius: 4px;

border: 1px solid #bcbcbc;

padding: 8px;

width: 60%;

text-align: center;

margin-bottom: -3px;

}

#modalFile {

object-fit: cover;

width: 110px;

height: 110px;

border: 1px solid #bcbcbc;

border-radius: 3px;

}

.modalBut {

margin-top: -11px;

}

Додаток Н  
Лістинг файлу «component.css»

.flex-container {

display: flex;

flex-flow: row-reverse wrap;

justify-content: space-around;

align-items: baseline;

}

.flex-container \* {

list-style-type: none;

}

.contImg {

width: 220px;

min-height: 220px;

object-fit: scale-down;

}

.compTable {

text-align: center;

display: none;

position: fixed;

left: 0;

right: 0;

}

.compContent {

border: solid black 2px;

background-color: white;

border-radius: 10px;

width: 240px;

margin: auto;

margin-top: -1700px;

position: relative;

}

.compImgSel,

.compImg {

padding-top: 10px;

box-shadow: 1px 1px 10px #969595;

width: 220px;

display: block;

margin: auto;

margin-top: 5px;

}

.compTitle {

text-align: center;

font-size: larger;

font-weight: bolder;

}

.compLine {

text-align: center;

font-size: smaller;}

.displayzone {

display: flex;

flex-flow: row wrap;

justify-content: space-evenly;}

.compContentSelect {

border: solid black 2px;

padding: 10px 5px;

border-radius: 10px;

background-color: white;

margin: 10px;

width: 220px;

height: 460px; }

.compContentSelect:hover {

box-shadow: 2px 2px 20px #969595;}

.compImgSel {

min-height: 220px;

object-fit: contain;

}

Додаток   
Компакт-диск із програмним продуктом