

Київський національний університет імені Тараса Шевченка
Факультет радіофізики, електроніки та комп'ютерних систем
Кафедра комп'ютерної інженерії

**МЕТОДИ АВТОМАТИЗАЦІЇ ФОРМУВАННЯ ЗАЯВОК НА
ЗАКУПІВЛЮ ОБЛАДНАННЯ**

Дипломна робота магістра
студента 2 року навчання
спеціальність: 123 Комп'ютерна інженерія
Тараса ВОНСА

Науковий керівник
канд. фіз.-мат. наук Юрій БОЙКО,
доцент кафедри комп'ютерної інженерії

Рецензент
доктор фіз.-мат. наук Євген ІВОХІН,
професор кафедри системного аналізу
та теорії прийняття рішень
факультету комп'ютерних наук та кібернетики

До захисту допускаю:

Завідувач кафедрою

Юрій БОЙКО /

Ухвалено на засіданні кафедри “_____” _____ 2022 р., протокол
№ _____

Київ - 2022

РЕФЕРАТ

Обсяг роботи 48 сторінок, 16 ілюстрацій, 4 таблиці, 14 джерел посилань, 7 додатків.

ЗАКУПІВЛЯ ТОВАРІВ, БРАУЗЕР, РОЗШИРЕННЯ ДЛЯ БРАУЗЕРА, ІНТЕРФЕЙС КОРИСТУВАЧА, ВЕБСАЙТ АГРЕГАТОР, STORAGE, SINGLE PAGE APPLICATION.

Актуальність роботи обґрунтовується автоматизацією основних процесів в закладах вищої освіти, в тому й числі організації електронного документообігу. Одним із цих процесів є формування заявок для публічних закупівель товарів, робіт та послуг структурними підрозділами університету.

Предметом роботи є програмний засіб для автоматизації формування вище згаданих заявок на придбання обладнання.

Метою роботи є дослідження можливих методів формування заявок на закупівлю та створення інструментарію для автоматизації цього процесу.

Методи розроблення: метод ручного створення програмного продукту за допомогою платформи розробки вебзастосунків. Інструменти розроблення: безкоштовне, вільно поширюване інтегроване середовище розробки Visual Studio Code, мова програмування JavaScript.

Результати роботи: виконано загальний огляд вимог для створення заявок на публічну закупівлю товарів, досліджено основні вимоги та компоненти розширень для браузера, розроблено розширення для браузера Google Chrome (запропонована назва «KNU Orders»), яке дозволяє отримувати із сайту-агрегатора загальну інформацію про обладнання, його технічні характеристики, діапазон цін, запропонові різними постачальниками та створювати заявки на закупівлю.

Зміст

	С.
РЕФЕРАТ	2
СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	5
ВСТУП	6
РОЗДІЛ 1. ПРИНЦИПИ ПРОВЕДЕННЯ ПУБЛІЧНИХ ЗАКУПІВЕЛЬ ДЕРЖАВНИМИ ОРГАНІЗАЦІЯМИ	8
1.1 Вимоги законодавства України щодо публічних закупівель	8
1.2 Порядок проведення закупівель в університеті	9
РОЗДІЛ 2. ПРОГРАМНИЙ ІНСТРУМЕНТАРІЙ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ ФОРМУВАННЯ ЗАЯВОК	12
2.1 Аналіз існуючих рішень щодо формування заявок на закупівлю .	12
2.2. Технічні вимоги щодо інструментарію	16
РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ ДЛЯ ФОРМУВАННЯ ЗАЯВОК	20
3.1 Визначення стеку технологій для розроблення додатку для браузера	20
3.2 Побудова програмного забезпечення для збереження інформації	22
3.2.1 Створення файлу маніфесту	22
3.2.2 Структура проєкту	23
3.2.3 Взаємодія між частинами проєкту	28
3.3 Додавання розширення на панель інструментів	31
ВИСНОВКИ	33
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	34
ДОДАТКИ	36
Додаток А. Вміст файлу manifest.json	36

Додаток Б. Вміст файлу rorur.html	37
Додаток В. Вміст файлу rorur.css	38
Додаток Г. Вміст файлу rorur.js	41
Додаток Г. Вміст файлу background.js.....	45
Додаток Д. Вміст файлу content.js.....	46
Додаток Е. Програмний код експорту заявки в CSV-файл	48

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

ІК	– Інтерфейс користувача
ПЗ	– Програмне забезпечення
РБ	– Розширення для браузера
API	– Application programming interface
DOM	– Document Object Model
JS	– JavaScript
HTML	– Hypertext Markup Language
CSS	– Cascade StaleSheet
JSON	– JavaScript Object Notation
SPA	– Single Page Application
ST	– Browser storage (сховище браузера)

ВСТУП

В час автоматизації та комп'ютеризації управлінських процесів і зберіганні інформації на електронних носіях все більшої популярності набуває електронний документообіг. Використання електронних засобів взаємодії або систем документообігу є однієї із основних частин функціонування будь-яких сучасних установ. Електронний документообіг [12] в порівнянні з паперовим дозволяє вивести планування діяльності організації, облік та контроль на всіх етапах управління на новий рівень – комплексного підходу, системного аналізу та прогнозування [6]. Він дозволяє інтегрувати усі документальні системи організації в єдиний інформаційний простір. При цьому це здійснюється без будь-яких втрат якості роботи із інформацією. Основним аспектом такої інтеграції виступає сховище документів та системи документообігу, що можуть з ним взаємодіяти.

«Електронний документообіг (обіг електронних документів) – сукупність процесів створення, оброблення, відправлення, передавання, одержання, зберігання, використання та знищення електронних документів, які виконуються із застосуванням перевірки цілісності та у разі необхідності з підтвердженням факту одержання таких документів» [8].

Основні переваги та особливості електронного документообігу:

- дозволяє виконувати різні операції над документом паралельно, щоб скоротити час руху файлу та підвищити ефективність його використання;
- забезпечує постійний (безперервний) руху файлів;
- надає єдину базу даних для централізованого зберігання та дедуплікації документів;
- організовує ефективну систему пошуку файлів.

Використання систем електронного документообігу є автоматизацією основних процесів в закладах вищої освіти. Одним із цих процесів є

організація публічних закупівель товарів, робіт та послуг структурними підрозділами університету. Саме для автоматизації процесу створення заявок на закупівлі технічного обладнання та інтегрування цього в системи документообігу було спроектовано та реалізовано систему «KNU Orders».

РОЗДІЛ 1. ПРИНЦИПИ ПРОВЕДЕННЯ ПУБЛІЧНИХ ЗАКУПІВЕЛЬ ДЕРЖАВНИМИ ОРГАНІЗАЦІЯМИ

Публічні закупівлі — це придбання товарів, робіт і послуг, які здійснюються за кошти платників податків.

1.1 Вимоги законодавства України щодо публічних закупівель

З метою забезпечення ефективного і прозорого здійснення закупівель, попередження проявів корупції в цій сфері та розвитку добросовісної конкуренції, будь-які державні некомерційні організації, в тому числі й університети, мають здійснювати закупівлі товарів, робіт і послуг виключно на основі та в рамках Закону України «Про публічні закупівлі».

Обов'язковою складовою публічних закупівель в державному секторі є тендер. Згідно із законодавством України державні підприємства повинні здійснювати будь-які публічні закупівлі на авторизованих електронних майданчиках. Ця процедура відбувається виключно через тендери в системі Prozorro.

Відповідно до Закону України «Про публічні закупівлі» усі закупівлі товарів, послуг та робіт класифікуються за рівнем вартості. У цьому нормативному-правовому акті зазначаються наступна гранична вартість («пороги»):

«1. Допорогові:

- вартість товарів та послуг, які планується придбати, становить до 200 тис. грн. (в окремих сферах господарювання – до 1 млн. грн.);
- вартість робіт – до 1,5 млн. грн. (для окремих сфер – до 5 млн. грн.).

2. Надпорогові:

- вартість товарів та послуг, які планується придбати, складає від 200 тис. грн. (або від 1 млн. грн.);

- вартість робіт – від 1,5 млн. грн. (від 5 млн. грн. для окремих сфер)» [13].

1.2 Порядок проведення закупівель в університеті

В рамках Київського національного університету імені Тараса Шевченка розроблена структура технологічний ланцюжок публічних закупівель. Цей процес показаний на схемі нижче.

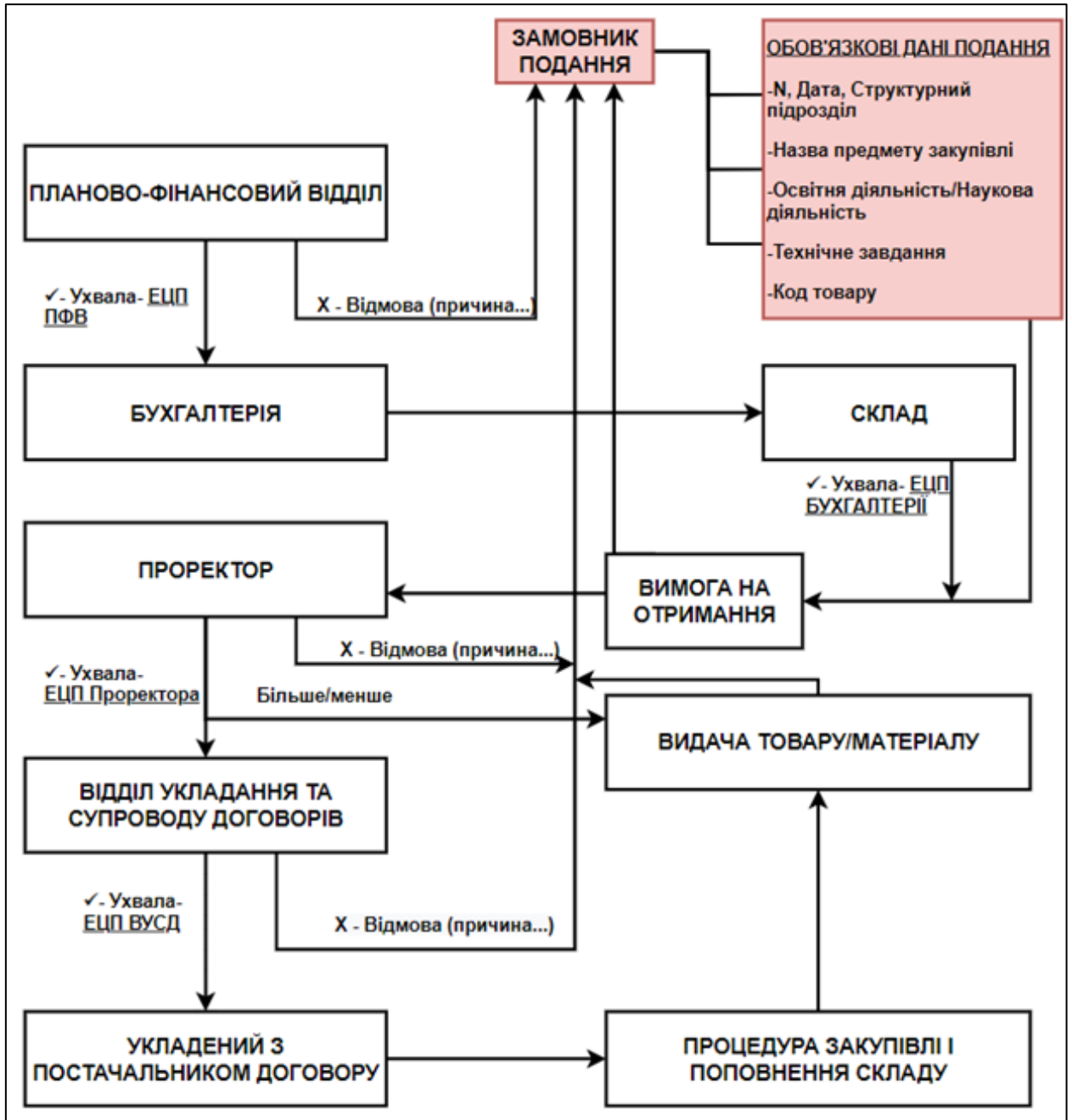


Рис. 1 Технологічний ланцюжок публічних закупівель університету

Для проведення закупівель в рамках Київського національного університету імені Тараса Шевченка створено відповідний підрозділ. Він займається організацією та здійсненням процедур закупівель товарів, робіт і послуг за державні кошти. Метою його створення є організація та проведення процедур закупівель на засадах колегіальності та неупередженості.

Основою для проведення публічної закупівлі товарів, робіт та послуг є подання заявок від структурних підрозділів університету на здійснення даної закупівлі. Подання регулюється розпорядженням університету від 17.10.2020 № 47 «Щодо порядку визначення предмету закупівлі товарів, робіт та послуг структурними підрозділами університету»

Відповідно до пунктів 4-5 цього розпорядження:

«4. У поданні на закупівлю товару **ОБОВ'ЯЗКОВО** зазначаються такі технічні характеристики:

- конкретна назва товару;
- марка товару, ДСТУ, ГОСТ (у разі наявності);
- країна виробник товару;
- розміри, колір товару, одиниці виміру, необхідна кількість товару, фасування товару; матеріал, з якого виготовлено товар, потужність, робочий тиск тощо (у разі наявності).

5. У поданні обов'язково вказується орієнтовна ціна товару, роботи чи послуги за одиницю. Ціна визначається структурним підрозділом самостійно, виходячи з моніторингу ціни на конкретний вид товару, роботи чи послуги. До подання **ОБОВ'ЯЗКОВО** долучається **ТРИ** комерційні пропозиції від різних постачальників/ надавачів послуг як підтвердження проведення моніторингу цін та їх відповідності ринковим цінам» [14].

Якщо подання не відповідає встановленим вимогам, то воно не буде розглядатись та потребує повторного створення.

Поставлені завдання у рамках дипломної роботи передбачають наступні пункти:

- визначення основного методу проведення моніторингу цін;
- порівняння можливих рішень формування заявок;
- розробка програмного забезпечення, для автоматизації цього процесу на основі проведеного аналізу.

РОЗДІЛ 2. ПРОГРАМНИЙ ІНСТРУМЕНТАРІЙ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ ФОРМУВАННЯ ЗАЯВОК

2.1 Аналіз існуючих рішень щодо формування заявок на закупівлю

Відповідно до пункту 5 розпорядження університету (див. підрозділ 1.2) ключовим елементом процедури подання заявки на закупівлю є проведення моніторингу цін та їх відповідності ринковим цінам.

Для того, щоб визначити очікувану вартість кожного товару закупівлі потрібно здійснити кілька кроків:

- 1) визначте повну назву товару, який потрібно закупити;
- 2) сформуйте опис товару, де потрібно вказати усі технічні та якісні характеристики, що можуть вплинути на ціну (якщо потрібно, то за погодженням із спеціалістом, відповідальність за якість продукту);
- 3) виберіть необхідний уніфікований профіль товару (опис експлуатаційних, якісних і технічних характеристик однакових або подібних продуктів різних виробників і постачальників товарної групи в межах цінового діапазону) при здійсненні закупівлі з використанням такого профілю;
- 4) визначте особливості та умови доставки;
- 5) визначте умови оплати.

Визначення очікуваної вартості об'єкта закупівлі в основному здійснюється шляхом порівняння ринкових цін.

«Метод порівняння (співставлення) ринкових цін – це метод визначення очікуваної вартості товару на підставі даних ринку, а саме загальнодоступної відкритої цінової інформації та інформації з отриманих комерційних пропозицій та прайс-листів на момент вивчення ринку» [7].

Точність отриманих оцінок очікуваної вартості безпосередньо залежить від достовірності використаної інформації про ціни.

Щоб розрахувати очікувану вартість шляхом співставлення ринкових цін, потрібно виконати наступні кроки:

1) зробіть усі ринкові ціни єдиними, включаючи валюту, умови доставки, умови оплати тощо;

2) з набору даних про ціни (який повинен містити щонайменше 3) при необхідності відкиньте незвичайно низькі та незвичайно високі ціни, які об'єктивно виходять за межі діапазону. Незвичайно низька/висока ціна – ціна, яка значно відрізняється (на 50% або більше) від сусідньої ціни;

3) визначте очікувану ціну за одиницю предмету закупівлі як середнє арифметичне отриманого набору даних. Це необхідно зробити за наступною формулою:

$$\langle C_{\text{од}} = (C_1 + \dots + C_K) / K, \text{ де:} \quad (1)$$

- $C_{\text{од}}$ – очікувана ціна за одиницю предмету;
- C_1, C_K – ціни отримані з відкритих джерел інформації;
- K – кількість цін отриманих з відкритих джерел інформації» [7];

4) визначте очікувану вартість товару. Це необхідно зробити за наступною формулою:

$$\langle \text{OB}_{\text{мрц}} = C_{\text{од}} \times V, \text{ де:} \quad (2)$$

- $\text{OB}_{\text{мрц}}$ – очікувана вартість за методом ринкових цін;
- $C_{\text{од}}$ – очікувана ціна за одиницю товару;
- V – кількість (обсяг) товару, що закуповується» [7].

Законодавство ніяк не обмежує вибір джерел цінової інформації [10]. Вся інформація, отримана з Інтернету чи інших джерел має бути роздрукована та надаватись у звітах про розрахунок вартості продукту. Серед основних джерел можна виділити наступні:

- списки із цінами (прайс-листи) виробників та постачальників відповідної продукції;
- комерційні пропозиції виробників та постачальників;
- спеціалізовані торгівельні майданчики – інтернет-ресурси;
- світові, регіональні та галузеві біржі;
- статистичні дані офіційних служб статистики України та країн походження товару;
- дані власних угод і публічна інформація з ProZorro [9];
- ціни врегульовані державою (постанови, накази, тарифи);
- аналітичні та статистичні дані інформаційних видань (агенств).

Перед розробкою системи формування заявок на закупівлю «KNU Orders» Київського національного університету імені Тараса Шевченка, було проаналізовано можливі способи проведення моніторингу цін:

- 1) ручний моніторинг;
- 2) використання програмного забезпечення (агрегатора);

На рисунках нижче зображено процес моніторингу цін для кожного із цих способів.

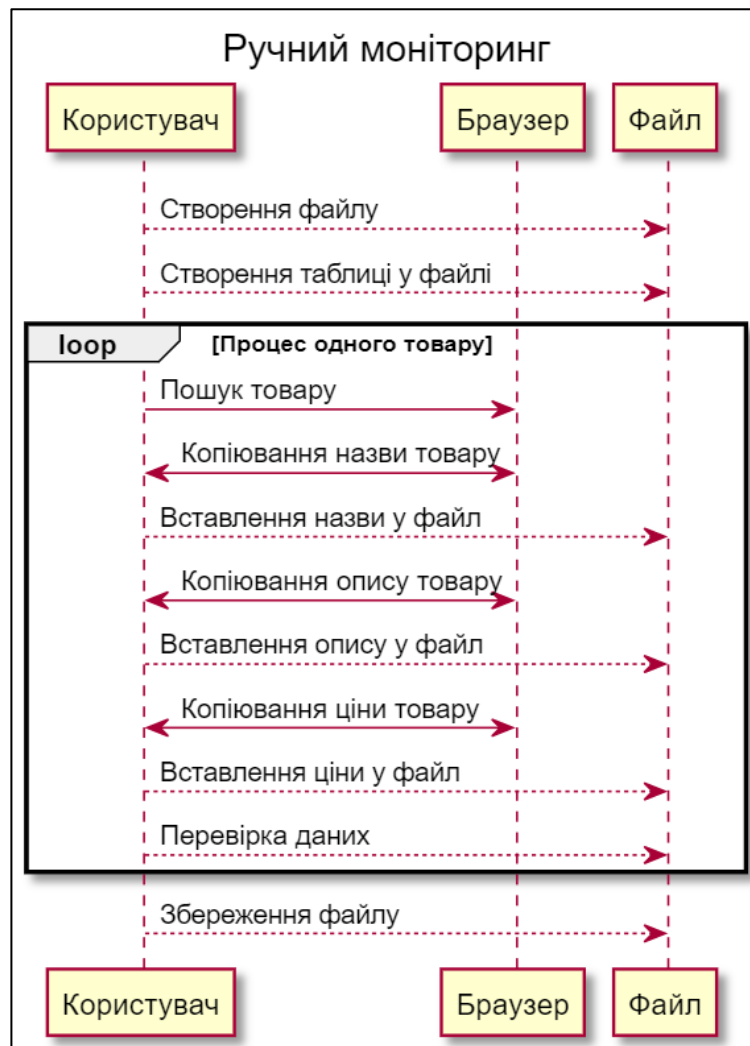


Рис. 2 Процес ручного моніторингу цін

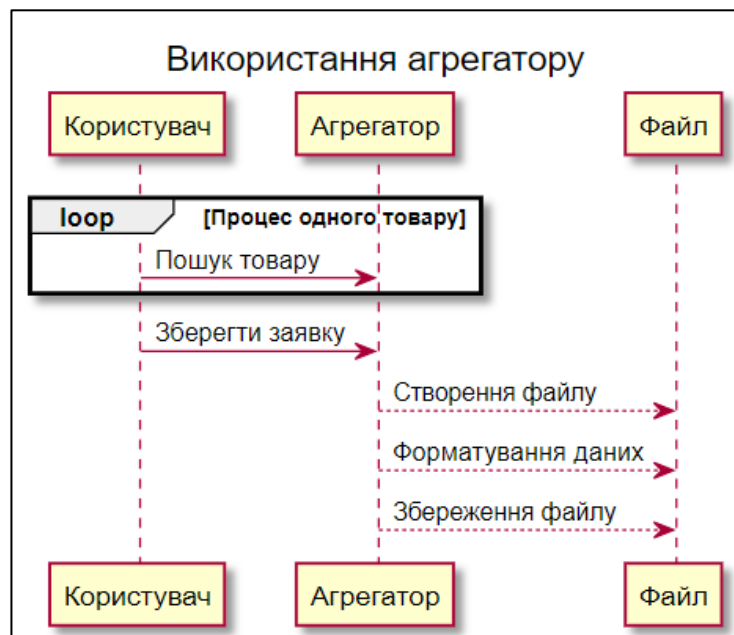


Рис. 3 Процес моніторингу цін із використанням ПЗ-агрегатора

Проаналізувавши процеси моніторингу, дані способи були порівняні в основному по швидкодії.

	Ручний моніторинг	Використання ПЗ-агрегатора
Швидкодія	Повільний, так як вимагає ручного заповнення всієї інформації про товар та перевірку коректності даних	Швидкий, тому що все заповнення та перевірка відбувається автоматично

Таблиця 1. Порівняння способів моніторингу цін.

Отже, проведення моніторингу з використанням ПЗ-агрегатора значно економить час цього процесу, що робить його кращим рішенням. Один із основних способів реалізації цього рішення – це сайт, на якому постачальники самі зможуть реєструвати товари.

Проте створення такого ПЗ на базі університетської інфраструктури вимагає великих ресурсів. Перш за все, це необхідність серверу, на якому буде відбуватись основна робота, а саме підтримка API для отримання запитів від постачальників для створення, редагування чи видалення товарів. Крім цього необхідна база даних, яка зможе зберігати всю цю інформацію. І основне, це наявність людських ресурсів, для майбутньої підтримки цієї системи, її популяризації та зв'язком із постачальниками.

2.2. Технічні вимоги щодо інструментарію

Проаналізувавши вимоги університету щодо заявок на закупівлю (див. підрозділ 1.2), а також можливий моніторинг цін (див. підрозділ 2.1) було вирішено, що найбільш доцільним є розробка програмного забезпечення «KNU Orders» із товстим користувацьким клієнтом, тобто таким, який проводить запитовані користувачем операції незалежно від центрального сервера. Центральний сервер в такому варіанті архітектури може

використовуватися як сховище даних, обробка та надання яких переноситься на робочу машину клієнта. Такий клієнт має певні переваги:

- має ширшу функціональність, ніж тонкий;
- режим багатокористувацької роботи;
- дозволяє працювати навіть при обривах зв'язку з сервером;
- висока швидкодія (залежить від апаратних засобів клієнта)

У таблиці нижче наведено порівняння використання власного ПЗ-агрегатора та додатку до вже існуючого.

	Власний ПЗ-агрегатор	Існуючий ПЗ-агрегатор
Ресурсо-затратність	<ul style="list-style-type: none"> – наявність серверу – наявність БД – розробка API та користувацького інтерфейсу – підтримка системи в майбутньому 	<ul style="list-style-type: none"> – розробка додатку – розробка користувацького інтерфейсу

Таблиця 2. Порівняння використання агрегаторів

Як видно із таблиці головною перевагою використання додатку до існуючого агрегатора перед іншим варіантом є його невелика ресурсозатратність. Тому вирішено створити цей клієнт у вигляді розширення для браузера, яке буде використовувати за основу один із поширених сайтів-агрегаторів України.

Основні характеристики розширення для браузера:

- доступний у будь-якому браузері (необхідно лише додати до розширень браузера)
- можна використовувати на будь-яких пристроях без портування на цільову операційну систему (браузер і його віртуальна машина виступає як цільова універсальна операційна система і комп'ютер).

- швидкість доступу, зручність і зрозумілість використання;
- можливість додати свою функціональність в сторонні продукти, доступу до ядра яких немає;
- можливість об'єднувати свої системи і хмарні сервіси в комплексний корпоративний ландшафт систем;
- простота в обслуговуванні - мають загальну кодову основу (архітектуру) незалежно від пристрою, на якому переглядаються;

Умовна структура розширення представлена на рисунку 3. Стрілками зображено взаємодію між основними компонентами.

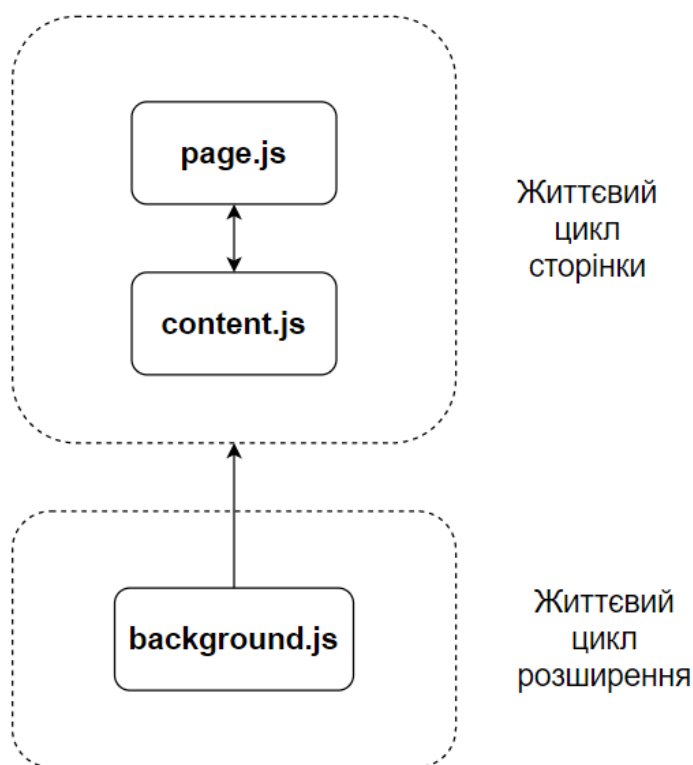


Рис. 4 Умовна структура розширення

Під час роботи додаток для браузера використовує три типи скриптів, наведені на рисунку вище. Вони складають його ядро.

background.js - це невидима фонові сторінка. Вона існує протягом життєвого циклу розширення, тобто ініціалізація викликається під час його встановлення або запуску браузера. Усі вкладки браузера мають однаковий

сценарій `background.js`. Це означає, що він не буде зберігати стан конкретного вмісту сторінки. Він має повний доступ до браузерного API, проте не може співпрацювати із DOM-об'єктами сторінок.

`content.js` – це скрипт, який залежить від окремої вкладки браузерного вікна. Для кожної нової вкладки буде ініціалізовано новий `content.js`. Він також може змінювати елементи розмітки сторінки, і має частковий доступ до браузерного API.

`page.js` – це основний скрипт, з яким працює користувацький інтерфейс розширення для браузера. Він має частковий доступ до браузерного API та лише доступ до DOM-об'єктів вікна самого додатку.

РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ ДЛЯ ФОРМУВАННЯ ЗАЯВОК

Як було зазначено у пункті 2.2 найкращим методом реалізації системи для формування заявок на закупівлю технічного обладнання виступає розробка розширення для браузера. Основою його має виступає вже існуючий в Україні сайт-агрегатор. Було проведено порівняльний аналіз найбільш популярних агрегаторів товарів на сьогодні. Результати наведені в таблиці нижче.

	hotline.ua	price.ua	e-Katalog.ua	magazilla.ua	sравни.ua
Відвідуваність, к-сть раз/місяць	11 млн	4.5 млн	4 млн	940 тис	330 тис
Кількість магазинів	4500	1500	1000	1000	100
Кількість товарів	10 млн	4 млн	2 млн	2 млн	1 млн

Таблиця 3. Порівняння деяких поширених сайтів-агрегаторів товарів в Україні

Отже, в якості базового агрегатора, з яким буде працювати розширення для браузера буде виступати сайт «hotline.ua».

3.1 Визначення стеку технологій для розроблення додатку для браузера

Було вирішено створити розширення у вигляді Single Page Application. Цей підхід передбачає взаємодію з клієнтом шляхом динамічного переписування поточної вебсторінки з новими даними, замість стандартного методу веббраузера, який завантажує цілі нові сторінки. Метою є швидкі переходи, завдяки чому додаток буде більше схожий на настільну програму

Для розробки використовується стандартний стек технологій вебдодатків, тобто HTML/CSS/JavaScript [5].

HTML (HyperText Markup Language - «мова гіпертекстової розмітки») – «стандартизована мова розмітки документів у Всесвітній павутині. Більшість вебсторінок містять опис розмітки на мові HTML (або XHTML)» [3]. HTML визначає вміст сторінки: що користувач побачить у браузері. Вона може містити програми, написані мовами сценаріїв, такими як JavaScript. Вони будуть впливати на поведінку та вміст вебсторінок. Також дозволяє підключати CSS-файли, щоб визначити зовнішній вигляд і макет вмісту.

JavaScript (JS) — «динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript» [4]. Найчастіше ця мова використовується для створення вебсценаріїв, які дозволяють клієнтам (пристроєм кінцевих користувачів) змінювати структуру та зовнішній вигляд вебсторінок, асинхронно спілкуватися з серверами, керувати браузером тощо. Так як JavaScript є дуже популярна протягом багатьох років, до неї з'являється все більше нових бібліотек та фреймворків, які значно спрощують написання сценаріїв вебсторінок.

CSS (Cascading Style Sheets) — «це спеціальна мова стилю сторінок, що використовується для опису їх зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних» [2].

CSS найчастіше використовується для візуалізації сторінок, які написані за допомогою мов HTML і XHTML. При цьому форматування CSS також можна застосувати й до інших типів XML-файлів. Специфікація CSS постійно знаходиться на стадії оновлення.

CSS замінив табличну верстку вебсторінок. Основною її перевагою є поділ вмісту сторінки (даних) та його візуальне представлення. Такий поділ покращує сприйняття та зручність використання вмісту, забезпечує більшу гнучкість і контроль над відображенням вмісту в різних контекстах, робить контент більш структурованим і простішим, усуває дублювання тощо. CSS також дозволяє пристосувати вміст до різних умов відображення (на екрані

монітора, на мобільному пристрої). Один і той самий документ HTML або XML може виглядати по-різному залежно від CSS, що використовується. Стили представлення вебсторінки можна розділити на такі:

- авторські стилі (інформація, що надається розробником сторінки):
- стилі користувача
- стилі браузера

CSS має свій стандарт, що визначає область застосування стилів, тобто порядок і елементи, в яких застосовуються стилі. Тому принцип каскаду використовується тоді, коли вказується лише інформація для елементів про стилі, які змінилися або не визначені більш загальними стилями

Ще деякі технології, які використовуються для розробки браузерних розширень – це API браузера. Вони безпосередньо вбудовані в браузер та дозволяють розробникам виконувати складні операції, не маючи справу зі складним кодом нижнього рівня. Існує ряд API-інтерфейсів браузера для маніпулювання DOM, виконання мережових запитів, керування сховищем на стороні клієнта, отримання медіапотоків пристрою тощо.

3.2 Побудова програмного забезпечення для збереження інформації

3.2.1 Створення файлу маніфесту

Кожне розширення повинне мати файл маніфесту. Manifest.json – це простий файл JSON на ваш вебсайт, який повідомляє браузеру про цей сайт на мобільному пристрої або комп'ютері користувача [1]. Наприклад, браузер Chrome вимагає мати маніфест, щоб відображати значок додатку на панелі закладок браузера.

Коли користувач встановлює або додає вашу вебпрограму в закладки на головний екран або додає її до програми запуску, manifest.json надає браузеру загальну інформацію про розширення. Так файл manifest.json містить відомості про версію маніфесту, назву додатку, значки, які він має використовувати, фоновий сценарій, який має відпрацьовувати під час

відкриття, сайти, для якого розширення має можливість застосовувати сценарій **content.js**, а також права, що має у браузері (табл. 3.2). У правах вказуємо **activeTab**, **tabs** та **storage**. Перше правило означає, що розширення має доступ до вкладки, що наразі відкрита. Друге свідчить про те, що наявний доступ до усіх вкладок, є можливістю для створення, зміни та впорядкування вкладок у браузері. І останнє відповідає за доступ розширення до зберігання, отримання та відстежування зміни в даних користувача. Розширення буде використовувати це право в основному для зберігання списку товарів у локальному сховищі то їх отримання для відображенні інформації.

Файл маніфесту	
Версія	2.0
Назва	KNU Orders
Значок	knu_logo.png
Фоновий сценарій	background.js
Сайти	https://hotline.ua/*
Права	activeTab, tabs, storage

Таблиця 4. Основна інформація маніфесту розширення

Manifest.json розширення «KNU Orders» можна переглянути в Додатку А.

3.2.2 Структура проєкту

Додаток для браузера «KNU Orders» буде складатися із двох основних частин:

- 1) частини, що відповідає за відображення товарів в заявці;
- 2) частини роботи з браузером.

Файлова структура проєкту, що включає перераховані частини наведена на рисунку 5 та схематичне зображення роботи частин розширення продемонстровано на рисунку 6.



Рис.5 Вигляд файлової структури проекту

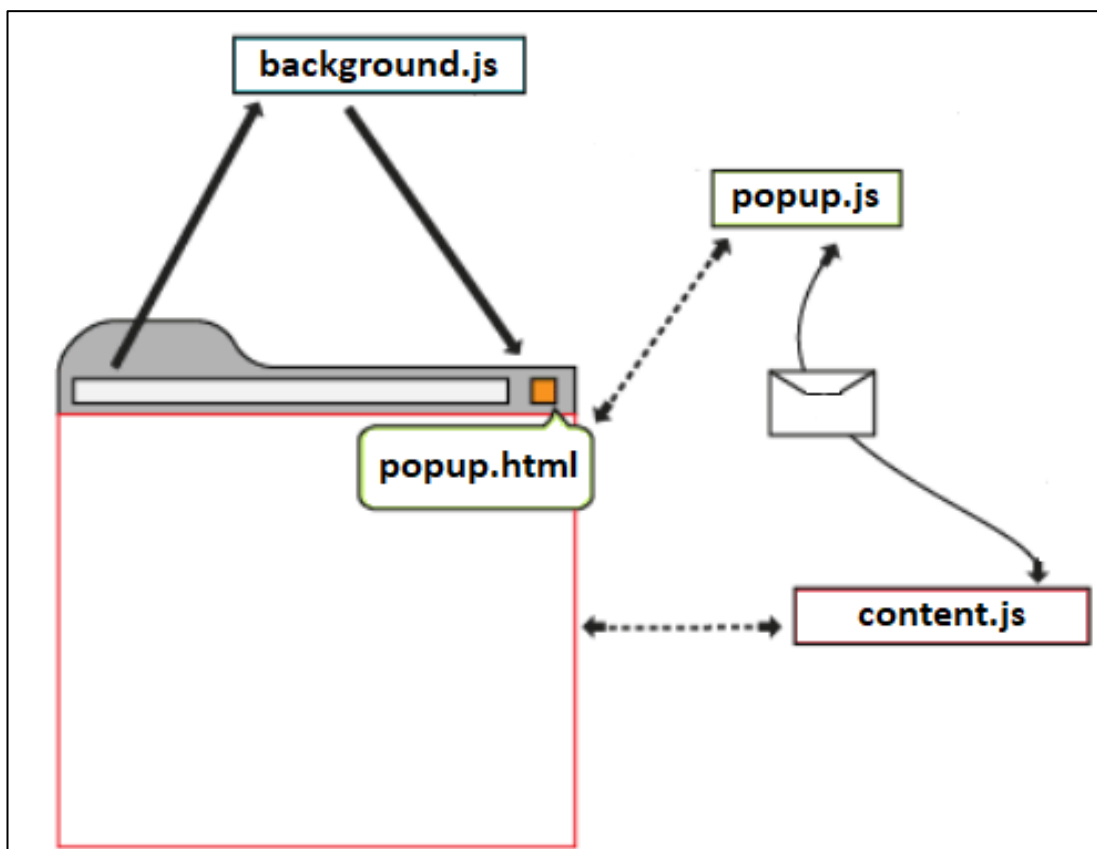


Рис. 6 Схематичне зображення взаємодії частин розширення

3.2.2.1 Частина, що відповідає за відображення товарів в заявці

Ця частина являє собою окреме вікно браузера, яке відкривається при натисканні на значок розширення. Вона складається із трьох основних файлів:

- 1) rorup.html – відповідає за HTML-розмітку даного вікна, його основні компоненти тощо (див. Додаток Б);
- 2) rorup.css – відповідає за стилі елементів даного вікна (див. Додаток В);
- 3) rorup.js – відповідає за основний функціонал: додавання, видалення товарів, ін'єкцію нових DOM-елементів у вікно, експорт заявки, комунікацію з іншими компонентами розширення тощо (див. Додаток Г).

Це вікно буде містити усі додані до заявки товари із сайту «hotline.ua». Для кожного товару буде відображатись наступна інформація:

- назва;
- опис, наявний із сайту hotline.ua;
- діапазон цін, запропонований різними постачальниками;
- кількість (це значення можна змінювати окремо для кожного товару у заявці).

Заявка зберігається в локальному сховищі, тому при закритті розширення усі додані до нього товари не видаляються. При повторному відкритті усі товари автоматично відображаються у вікні, що спливає.

Також на сторінці є кнопка «Сформувати заявку», яка дозволяє експортувати поточну заявку у файл вибраного формату. Отже, можна сформувати заявку у файли CSV (див. Додаток Е), PDF та DOCX-форматах.

	A	B	C	D
1	Name	Description	Price	URL
2	Apple MacBook Air 13" Space Gray Late 2020 (MGN63)	Ноутбук • Класичний • 13,3 • IPS • 2560x1600 • Apple M1 • ОЗП: 8 ГБ • 7 core GPU • SSD: 256 ГБ • 1, 29 кг • ОС: macOS Big Sur • колір: сірий космос	36 927 грн	https://notline.ua/computer-noutbuki-netbuki/apple-macbook-air-13-space-gray-late-2020-mgn63/
3	Apple MacBook Pro 16" Space Gray 2021 (MK183)	Ноутбук • Класичний • 16,2 • mini-LED • 3456x2234 • Apple M1 Pro • ОЗП: 16 ГБ • 16 core GPU • SSD: 512 ГБ • 2, 1 кг • ОС: macOS Monterey • колір: сірий космос	86 694 грн	https://notline.ua/computer-noutbuki-netbuki/apple-macbook-pro-16-space-gray-2021-mk183/

Рис. 7 Вигляд заявки в EXCEL (файл імпортовано в CSV-форматі)

Уся інформація на сторінці відображається у формі «карточок». Реалізацію цього підходу можна переглянути у фрагментах із CSS кодів зазначених нижче.

Реалізація стилю «карточки» (CSS-код):

```
/*Card styles*/
.card{
  flex: 1;
  min-height: 30px;
  height:inherit;
  vertical-align: top;
  box-shadow:0 5px 5px rgba(0,0,0,.15), 0 0 1px
  rgba(0,0,0,.12);
  margin: 5px 0;
  margin-left: 0;
  display: inline-block;
  background: #fff;
  border-radius: 3px;
  padding: 16px;
  border-radius: 8px;
}
```

Основними компонентами «карточки» є її назва (card-title) та основний опис (card-data). Реалізація стилів цих компонентів наведено нижче:

```
/*Card title style*/
.card-title {
  font-size:14px;
  font-weight:bold;
  font-family: 'Roboto'
}

/*Card data style*/
.card-data {
  font-family: 'Roboto';
  font-size: 12px;
  font-weight: bolder;
}
```

```
color: rgb(0,0,0,.5);
margin-top:10px;
}
```

Вікно, що спливає зображено на рисунку нижче.

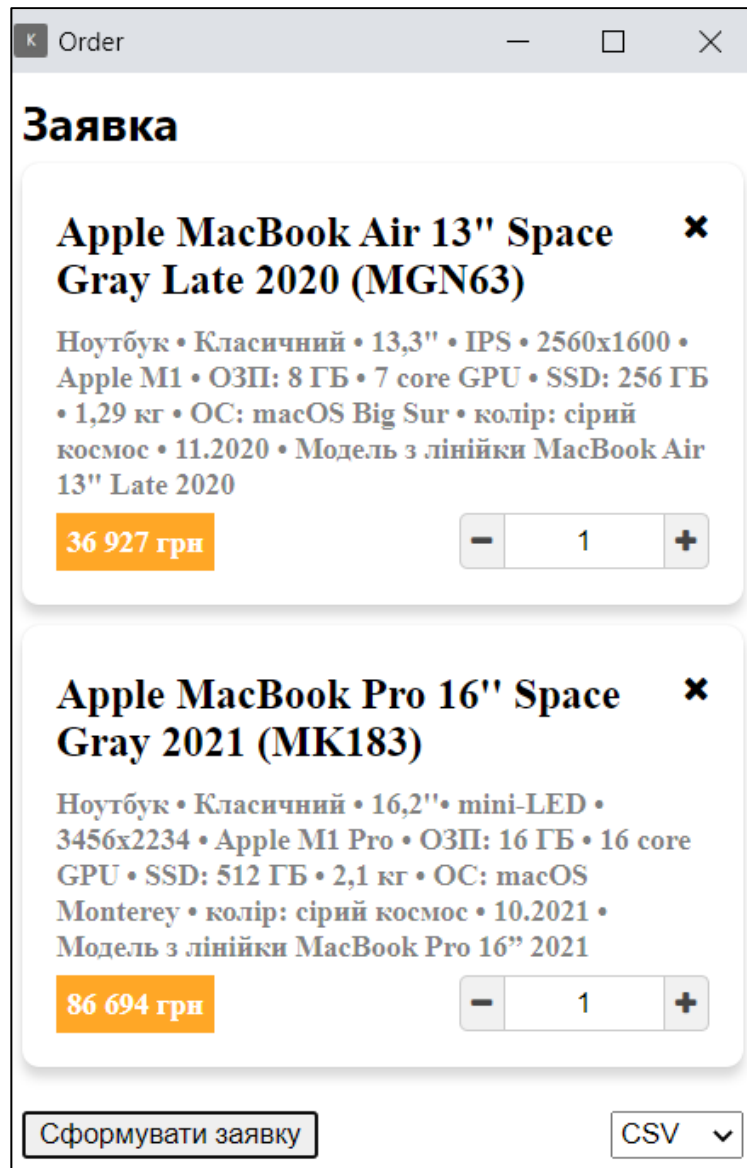


Рис. 8 Приклад відображення товарів в заявці

3.2.2.2 Частина роботи з браузером

Так як перша частина розширення створена лише для відображення інформації товарів у заявці, то вона немає прямого доступу до елементів розмітки сторінки сайту «hotline.ua». Уся робота із сайтом-агрегатором відбувається в другій частині розширення.

Ця частина являє собою набір з двох сценаріїв мовою JavaScript:

- 1) background.js (див. Додаток Г)
- 2) content.js (див. Додаток Д)

Сценарій в першому файлі (background.js) відповідає за відкриття вікна частини, що відповідає за відображення товарів в заявці під час натискання на значок розширення на панель інструментів браузера.

Основна робота виконується у сценарії content.js. Цей файл насамперед відповідає за ін'єкцію додаткових елементів розмітки у структуру сторінки «hotline.ua», отримання інформації із сайту та її відправлення у першу частину застосунку.

Реалізацію отримання інформації із сайту-агрегатора можна переглянути нижче (JS-код):

```
let parent = button.parentElement.parentElement;
let info = parent.getElementsByClassName("item-info")[0];
let main = info.querySelector(".h4 a");
let url = main.href;
let name = main.innerText;
let description = info.querySelector(".text div .text
p").innerText;
let price = decode(parent.querySelector(".item-price .stick-
pull .price-md").innerHTML);

let product = {
    name: name,
    price: price,
    description: description,
    url: url
};
```

3.2.3 Взаємодія між частинами проєкту

Для безперервної роботи між двома частинами розширення налаштовано систему «комунікації» за допомогою браузерного API *chrome*. В рамках проєкту реалізовано декілька основних двосторонніх взаємодій між частинами:

- 1) для додавання та видалення кнопок «Додати до заявки» на сторінку сайту «hotline.ua»;
- 2) для додавання та видалення товару із заявки.

Отже, перша взаємодія відбувається при відкритті та закритті вікна, що впливає відповідно. Наприклад, додавання кнопок відбувається наступним чином. Під час натискання на іконку розширення викликається сценарій `background.js`, який створює нову вкладку розширення. Це викликає відправлення повідомлення «`addButtons`» із сценарію `popup.js` за допомогою команди `sendMessage`. Далі у файлі `content.js` за допомогою команди `onMessage.addListener` відбувається отримання цього повідомлення та додавання кнопки «Додати до заявки» для кожного запропонованого товару на сайті «hotline.ua». На рисунках нижче зображено схему цього процесу та вигляд сайту-агрегатора до та після відкриття розширення.



Рис. 9 Процес додавання кнопок в DOM сайту-агрегатора «hotline.ua»



Рис. 10 Вигляд сайту-агрегатора «hotline.ua» до запуску розширення



Рис. 11 Вигляд сайту-агрегатора «hotline.ua» після запуску розширення

Друга взаємодія відбувається відповідно під час натискання на кнопку «Додати до заявки» додану на сайт процесом описаним вище або видалення товару із вікна, що спливає. Наприклад, додавання товару до заявки відбувається наступним чином. Після отримання інформації із сайту-агрегатора сценарій `content.js` за допомогою команди `sendMessage` відправляє дані на частину, яка відповідає за відображення товарів. У файлі `popup.js` за допомогою команди `onMessage.addListener` відбувається отримання та обробка інформації. На рисунку нижче зображено схему додавання товару до заявки.

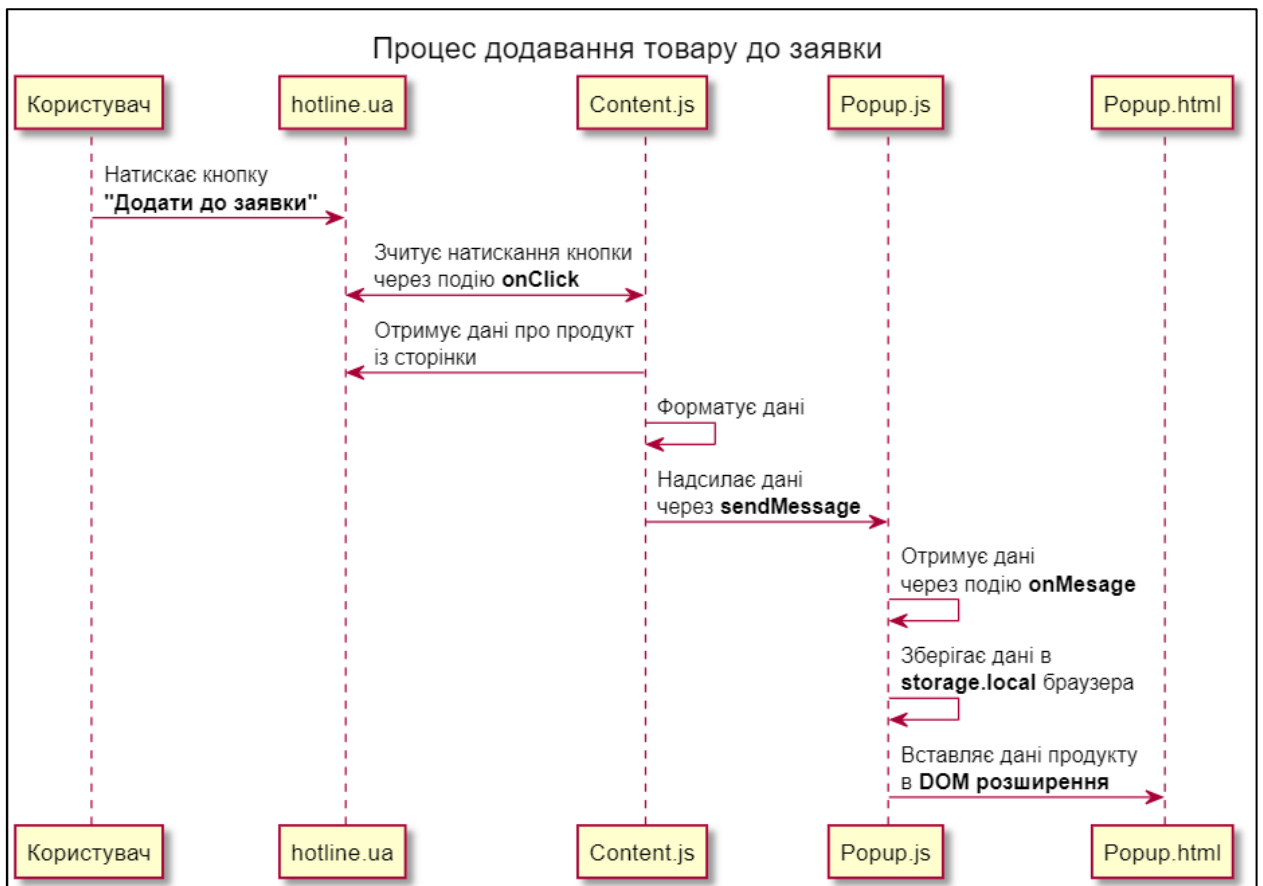


Рис. 12 Процес додавання товару до заявки

Як видно із рисунка від користувача потрібно лише знайти товар на «hotline.ua» та натиснути кнопку «Додати до заявки».

3.3 Додавання розширення на панель інструментів

Для використання розширення його необхідно додати до панелі інструментів браузера, наприклад Google Chrome. Цей процес складається із наступних кроків:

- 1) перейти на панель інструментів за допомогою посилання «<chrome://extensions/>» або налаштування;
- 2) увімкнути режим розробника та завантажити розширення (Рис 13-15).

Надалі заповане розширення можна буде переносити на необхідні пристрої клієнтів. Встановлення відбувається при запуску файлу із форматом **.crx** в браузері. При відкритті необхідно підтвердження, так як розширення

не було перевірено Google у галереї розширень. Після цього розширення з'явиться у панелі інструментів вашого браузера (Рис 16).

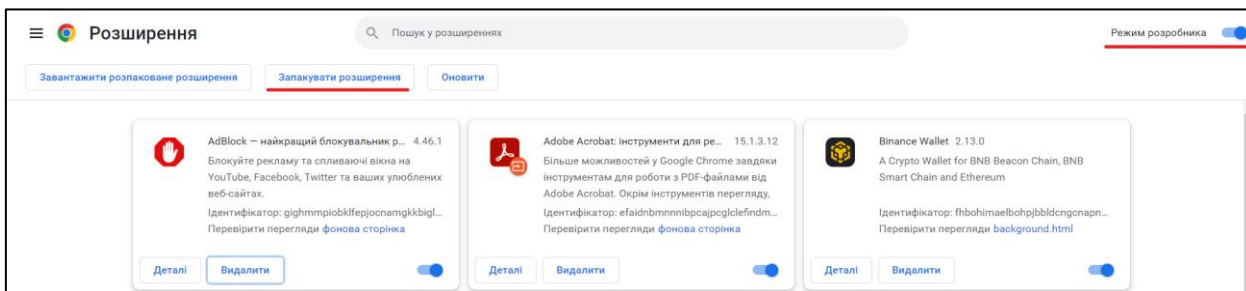


Рис. 13 Панель інструментів із ввімкненим режимом розробника

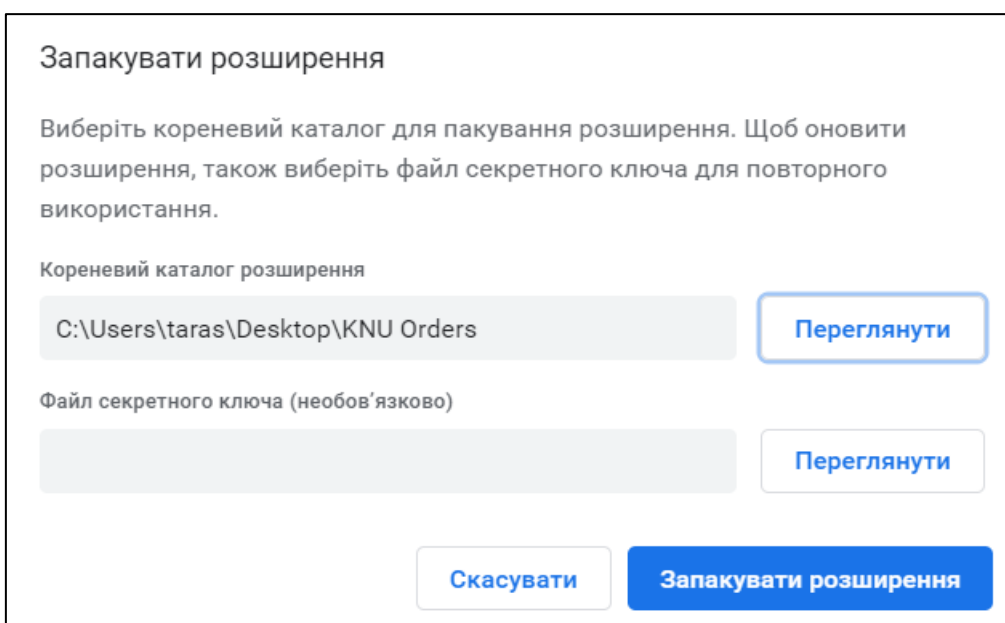


Рис. 14 Процес запакування розширення

📄 KNU Orders.crx	5/3/2022 2:13 AM	CRX File	261 KB
📄 KNU Orders.pem	5/3/2022 2:13 AM	PEM File	2 KB

Рис. 15 Запаковане розширення у файлі формату .crx

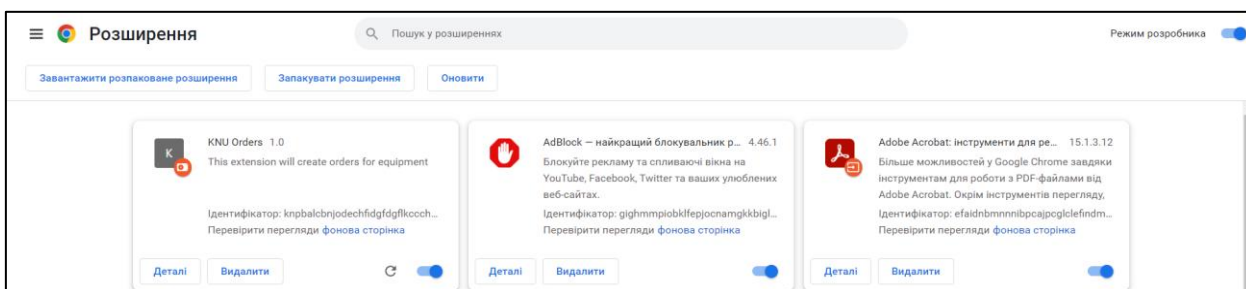


Рис. 16. Встановлене розширення до панелі інструментів

ВИСНОВКИ

В процесі виконання дипломної роботи магістра була розв'язана практична задача автоматизації формування заявок на купівлю обладнання.

У результаті роботи було проведено аналіз рішень моніторингу цін та створення заявок. Він показав, що доцільно при цьому використовувати одну із поширених систем-агрегаторів.

Вирішено реалізувати систему формування заявок у вигляді програми-розширення для браузера (browser extension), оскільки це рішення дозволяє додавати свою функціональність до стороннього сайту, доступу до ядра якого немає.

Обраний стек технологій (HTML/CSS/JavaScript) дозволяє повною мірою забезпечити сумісність програм, які використовуються при розробленні системи.

У результаті дипломної роботи магістра було створено технологічний ланцюжок та розроблено розширення для браузера на основі сайту-агрегатора «hotline.ua», яке задовільняє усім вимогам поставлених перед системою автоматизації формування заявок на придбання обладнання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Chrome Developers [Електронний ресурс] – 2020. – Режим доступу до ресурсу: <https://developer.chrome.com/docs/extensions/mv3/>. Дата звернення 10.03.2022.
2. CSS [Електронний ресурс] – 2021. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/CSS>. Дата звернення 15.03.2022.
3. HTML [Електронний ресурс] – 2021. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/HTML>. Дата звернення 15.03.2022.
4. JavaScript [Електронний ресурс] – 2021. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/JavaScript>. Дата звернення 15.03.2022.
5. MDN Web Docs [Електронний ресурс] – 2021. – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/>. Дата звернення 15.03.2022.
6. Вибір системи електронного документообігу. [Електронний ресурс]. Режим доступу до ресурсу: <https://fosdoc.com/vybor-sed>. Дата звернення 25.10.2021.
7. Визначення очікуваної вартості [Електронний ресурс]. – Режим доступу до ресурсу: <https://cpb.org.ua/viznachennya-ochikuvanoi-vartosti>. Дата звернення 10.11.2021.
8. Електронний документообіг [Електронний ресурс] – 2021. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Електронний_документообіг. Дата звернення 25.10.2021.
9. Київський національний університет імені Тараса Шевченка – ПУБЛІЧНІ ЗАКУПІВЛІ [Електронний ресурс] – Режим доступу до ресурсу: <http://public-purchase.knu.ua/>. Дата звернення 20.11.2021.
10. КОМЕНТАРІ ДО ЗАКОНОДАВСТВА УКРАЇНИ ПРО ПУБЛІЧНІ ЗАКУПІВЛІ [Електронний ресурс] – режим доступу до ресурсу: <https://infobox.prozorro.org/upload/files/main/980/555/zvitkomentar-122016-final-2.pdf>. Дата звернення 20.11.2021.

11. Настанови щодо участі в публічних закупівлях [Електронний ресурс] – режим доступу до ресурсу: <https://infobox.prozorro.org/upload/files/main/662/227/yak-brati-uchast-u-publichnikh-zakupivlyakh.pdf>. Дата звернення 20.11.2021.

12. Про електронні документи та електронний документообіг: Закон України від 22.05.2003 р. № 815-IV. Відомості Верховної Ради України. 2003. № 36, стаття 275;

13. Про публічні закупівлі : Закон України від 25.12.2015 р. № 922-VIII. Відомості Верховної Ради України. 2016. № 9. стор. 5, стаття 89;

14. Щодо порядку визначення предмету закупівлі товарів, робіт та послуг структурними підрозділами університету: Розпорядження від 17.10.2020 № 47 [Електронний ресурс] – режим доступу до ресурсу: <https://science.knu.ua/upload/iblock/4c4/4c40ef25c5defa882e7b8c36d87693cf.PD>
F. Дата звернення 20.11.2021.

ДОДАТКИ

Додаток А. Вміст файлу manifest.json

```
{
  "manifest_version": 2,
  "name": "KNU Orders",
  "description": "This extension will create orders for equipment",
  "version": "1.0",
  "browser_action": {
    "default_icon": "knu_logo.png"
  },
  "permissions": [
    "activeTab",
    "tabs",
    "storage"
  ],
  "background": {
    "scripts": ["/src/browser/background.js"],
    "persistent": false
  },
  "content_scripts": [
    {
      "matches": [
        "https://hotline.ua/*"
      ],
      "js": [
        "/src/browser/content.js"
      ]
    }
  ]
}
```

Додаток Б. Вміст файлу popup.html

```
<!doctype html>
<html>
  <head>
    <title>Order</title>
    <link href="popup.css" rel="stylesheet" type="text/css" />
    <meta charset="UTF-8">
  </head>
  <body>
    <h4>Заявка</h4>
    <div id='myItemList'>
    </div>
    <br/>
    <div id="bottom">
    <button id="createOrder">Сформувати заявку</button>
    <select id="format" name="order_format">
      <option value="1">CSV</option>
      <option value="2">PDF</option>
      <option value="3">DOCX</option>
    </select>
    </div>
    <script src="../models/product.js" type="module"></script>
    <script src="../models/order.js" type="module"></script>
    <script src="../helpers/exporter.js" type="module"></script>
    <script src="popup.js" type="module"></script>
  </body>
</html>
```

Додаток В. Вміст файлу popur.css

```
.card {
  flex: 1;
  min-height: 30px;
  height: inherit;
  vertical-align: top;
  box-shadow: 0 5px 5px rgba(0, 0, 0, .15), 0 0 1px rgba(0, 0, 0, .12);
  margin: 5px 0;
  margin-left: 0;
  display: inline-block;
  background: #fff;
  border-radius: 3px;
  padding: 16px;
  border-radius: 8px;
}

.card-title {
  font-size: 20px;
  font-weight: bold;
  font-family: 'Roboto'
}

.card-data {
  font-family: 'Roboto';
  font-size: 14px;
  font-weight: bolder;
  color: rgb(0, 0, 0, .5);
  margin-top: 10px;
}

.card-price {
  background-color: rgba(255, 167, 38, 255);
  color: #fff;
  display: inline-block;
  padding: 5px;
  margin: 0;
}

.card-row {
  display: flex;
  flex-direction: row;
  justify-content: space-between;
  margin-top: 5px;
}

.card-amount {
  display: flex;
}
```

```

.fa-close {
  font-size: 16px;
  margin-left: 10px;
}

.calc-num {
  outline: none;
  padding: 0 12px;
  width: 50px;
  text-align: center;
  vertical-align: top;
}

.calc-num,
.calc-num-dec,
.calc-num-inc {
  height: 24px;
  user-select: none;
  border: 1px solid #ccc;
}

.calc-num-dec,
.calc-num-inc {
  display: inline-block;
  font-weight: bold;
  line-height: 24px;
  width: 20px;
  background: #f1f1f1;
  cursor: pointer;
  color: #444;
  text-align: center;
}

.calc-num-inc:active {
  background: #ddd;
}

.calc-num-dec {
  border-right: none;
  border-radius: 4px 0 0 4px;
}

.calc-num-inc {
  border-left: none;
  border-radius: 0 4px 4px 0;
}

#bottom {
  display: flex;
  justify-content: space-between;
}

```

```
}  
  
h4 {  
  font-size: 22px;  
  margin: 0;  
}  
  
i {  
  cursor: pointer;  
}
```

Додаток Г. Вміст файлу popup.js

```
import { Order } from "../models/order.js";
import { Product } from "../models/product.js";
import { Exporter } from "../helpers/exporter.js";

var order;

chrome.runtime.onMessage.addListener(

  function (msg, sender, sendResponse) {
    if (msg.action === "addProduct") {

      var product = Product.fromString(msg.product);
      order.add(product);
      addListItem(product, order.products.length);

      chrome.storage.local.get('windowId', function (result) {
        if (result.windowId) {
          chrome.windows.update(result.windowId, { focused: true });
        }
      });

      sendResponse({ status: "added" });
    }
  }

);

window.addEventListener('beforeunload', function () {
  chrome.storage.local.set({ 'opened': false });
  message({ action: 'removeButtons' });
});

function removeFromOrder(li) {
  let ul = document.getElementById('myItemList');
  ul.removeChild(li);
  order.remove(li.id);
};

function addListItem(product, index) {
  let list = document.getElementById('myItemList');
  let card = document.createElement("div", { "id": index });
  card.classList.add('card');

  let titleRow = document.createElement("div");
  titleRow.classList.add('card-row');
```

```

let title = document.createElement("div");
title.classList.add('card-title');
title.innerHTML = product.name;

let closeIcon = document.createElement('i');
closeIcon.classList.add('fa', 'fa-close');
closeIcon.onclick = () => removeFromOrder(card);

titleRow.appendChild(title);
titleRow.appendChild(closeIcon);

let data = document.createElement("div");

let description = document.createElement("div");
description.classList.add('card-data');
description.innerHTML = product.description;

let price = document.createElement("div");
price.classList.add('card-data', 'card-price');
price.innerHTML = product.price;

let amount = document.createElement("div");
amount.classList.add('card-amount');
amount.innerHTML =
  `

```

```

        order = Order.fromString(result.order);
        console.log(order);
        Array.from(order.products).forEach((product, index) => {
            addListItem(product, index);
        });
    }
    else {
        order = new Order('test');
    }
});

chrome.storage.local.set({ 'opened': true });

message({ action: 'addButtons' });

var createOrder = document.getElementById('createOrder');
createOrder.onclick = () => exportData();
}

function message(message) {
    chrome.tabs.query({}, function (tabs) {
        for (var i = 0; i < tabs.length; i++) {
            if (tabs[i].url.startsWith("https://hotline.ua/")) {
                chrome.tabs.sendMessage(tabs[i].id, message, function (response)
{
                    console.log(response.status);
                });
            }
        }
    });
}

function appendCalc(card) {

    var el = card.getElementsByClassName("input-number")[0];

    var min = el.getAttribute('min') || false;
    var max = el.getAttribute('max') || false;

    init(el);

    function init(el) {

        var els = {};

        els.dec = el.previousSibling;
        els.inc = el.nextSibling;

        els.dec.onclick = () => decrement();
        els.inc.onclick = () => increment();
    }
}

```

```

function decrement() {
    var value = el.value;
    value--;
    if (!min || value >= min) {
        el.value = value;
    }
}

function increment() {
    var value = el.value;
    value++;
    if (!max || value <= max) {
        el.value = value++;
    }
}
}
};

function exportData() {
    var format = document.getElementById("format").value;

    switch (format) {
        case 1:
            Exporter.exportCSVData(order);
            break;
        case 2:
            Exporter.exportPDFData(order);
            break;
        case 3:
            Exporter.exportDOCXData(order);
            break;
        default:
            alert("Need to implement\nPlease use CSV format")
    }
}

ready();

```

Додаток Г. Вміст файлу background.js

```
console.log("background running");

var myid = chrome.runtime.id;

chrome.browserAction.onClicked.addListener(
  function () {
    chrome.tabs.query({}, function (tabs) {
      var doFlag = true;
      for (var i = 0; i < tabs.length; i++) {
        if (tabs[i].url === `chrome-
extension://${myid}src/user/popup.html`) {
          doFlag = false;

          chrome.storage.local.get('windowId', function (result) {
            if (result.windowId) {
              chrome.windows.update(result.windowId, { focused:
true });
            }
          });
          break;
        }
      }
      if (doFlag) {
        chrome.windows.create({
          url: chrome.runtime.getURL("src/user/popup.html"),
          type: "popup",
          height: 500,
          width: 350,
          left: screen.width - 350,
          top: 100
        }, function (win) {
          console.log(win)
          chrome.storage.local.set({ 'windowId': win.id });
        });
      }
    });
  }
);
```

Додаток Д. Вміст файлу content.js

```
chrome.runtime.onMessage.addListener(  
  
  function (msg, sender, sendResponse) {  
    if (msg.action === "addButtons") {  
      sendResponse({ status: "added buttons" });  
      checkButtons();  
    }  
    else if (msg.action === "removeButtons") {  
      sendResponse({ status: "removed buttons" });  
      removeOrderButtons();  
      buttons = false;  
    }  
  }  
);  
  
function addOrderButtons() {  
  let items = document.getElementsByClassName("product-item");  
  
  Array.from(items).forEach((item) => {  
    let div = document.createElement("div");  
    div.classList.add('item-order');  
    div.appendChild(addListItem());  
  
    item.appendChild(div);  
  });  
}  
  
function removeOrderButtons() {  
  let items = document.getElementsByClassName("product-item");  
  
  Array.from(items).forEach((item) => {  
    let div = item.getElementsByClassName("item-order")[0];  
    item.removeChild(div);  
  });  
}  
  
function addListItem() {  
  let span = document.createElement("span", {"title": "Додати товар до  
накладної"});  
  span.classList.add('inline');  
  span.onclick = () => addToOrder(span);  
  
  let innerSpan = document.createElement("span");  
  innerSpan.setAttribute("class", "crutch-data");  
  innerSpan.innerHTML = "Додати до заявки";  
  innerSpan.style.cursor = "pointer";
```

```

span.appendChild(document.createElement("br"));
span.appendChild(innerSpan);

return span;
}

function addToOrder(button) {
  let parent = button.parentElement.parentElement;
  let info = parent.getElementsByClassName("item-info")[0];
  let main = info.querySelector(".h4 a");
  let url = main.href;
  let name = main.innerText;
  let description = info.querySelector(".text div .text p").innerText;
  let price = decode(parent.querySelector(".item-price .stick-pull .price-
md").innerHTML);

  let product = {
    name: name,
    price: price,
    description: description,
    url: url
  };

  chrome.runtime.sendMessage({ action: "addProduct", product:
JSON.stringify(product) }, function (response) {
    console.log(response.status);
  });
}

function decode(htmlInfo) {
  var info = document.createElement("textarea");
  info.innerHTML = htmlInfo;
  return info.value;
}

var buttons = false;

function checkButtons() {
  if (!buttons) {
    chrome.storage.local.get('opened', function (result) {
      if (result.opened) {
        addOrderButtons()
        buttons = true;
      }
    })
  }
}

checkButtons();

```

Додаток Е. Програмний код експорту заявки в CSV-файл

```
static exportCSVData(order){
  const csvString = [
    [
      "Name",
      "Description",
      "Price",
      "URL"
    ],
    ...(order.products).map(item => [
      item.name,
      item.description.includes(',') || item.description.includes('\n')
        ? `\"${item.description}\"`
        : item.description,
      item.price,
      item.url.includes(',')
        ? `\"${item.url}\"`
        : item.url
    ])
  ]
  .map(e => e.join(","))
  .join("\n");

  console.log(csvString);

  var universalBOM = "\uFEFF";
  var link = document.createElement("a");
  link.setAttribute("href", 'data:text/csv; charset=utf-8,' +
  encodeURIComponent(universalBOM+csvString));
  link.setAttribute("download", `Order_Report_${new Date()}.csv`);
  document.body.appendChild(link);

  link.click();
}
```