

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

**Пояснювальна записка**

до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ  
ЗАСОБІВ ЗБОРУ ІНФОРМАЦІЇ ПРО ТОВАРИ  
RESEARCH AND SOFTWARE IMPLEMENTATION OF  
TOOLS FOR COLLECTING INFORMATION ABOUT GOODS

Виконав(ла): студент(ка) 2 курсу, групи КНТ-113М  
Спеціальності 121 Інженерія програмного  
(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ФАЛАЛЄСВ О.О.

(ПРИЗВИЩЕ та ініціали)

Керівник ОЛІЙНИК А.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент ІЛ'ЯШЕНКО М.Б.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
**Національний університет «Запорізька політехніка»**

Факультет КНТ  
Кафедра програмних засобів  
Ступінь вищої освіти магістр  
Спеціальність 121 Інженерія програмного забезпечення  
(код і найменування)

Освітня програма (спеціалізація) Інженерія програмного забезпечення  
(назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ПЗ, д.т.н, проф.  
Сергій СУББОТІН  
“ ” 2024 року

**З А В Д А Н Н Я**

**НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)**

ФАЛАЛЄЄВА Олега Олексійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та програмна реалізація засобів збору інформації про товари. Research and Software Implementation of Tools for Collecting Information about Goods

керівник проєкту (роботи) д.т.н., професор, ОЛІЙНИК Андрій Олександрович,  
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “07” жовтня 2024 року № 398

2. Строк подання студентом проєкту (роботи) 02 грудня 2024 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

## 6. Консультанти розділів проєкту (роботи)

| Розділ              | ПРІЗВИЩЕ, ініціали та посада консультанта | Підпис, дата   |                           |
|---------------------|-------------------------------------------|----------------|---------------------------|
|                     |                                           | завдання видав | прийняв виконане завдання |
| 1-4 Основна частина | ОЛІЙНИК А.О., професор                    |                |                           |
| Нормоконтроль       | КАЛІНІНА М.В., асистент                   |                |                           |

7. Дата видачі завдання “9” вересня 2024 року.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проєкту (роботи)                             | Строк виконання етапів проєкту (роботи) | Примітка     |
|-------|----------------------------------------------------------------------|-----------------------------------------|--------------|
| 1     | Постановка завдання роботи.                                          | 1 тиждень                               | Завдання, ТЗ |
| 2     | Аналіз предметної області.                                           | 1 тиждень                               | Розділ 1     |
| 3     | Вибір мови програмування та інших технологій розробки.               | 2 тиждень                               | Розділ 2     |
| 4     | Розробка структури програми.                                         | 3 тиждень                               | Розділ 3     |
| 5     | Розробка програми.                                                   | 4-7 тижні                               | Розділи 3,4  |
| 6     | Тестування та експериментальне дослідження програмного забезпечення. | 8 тиждень                               | Розділ 4     |
| 7     | Оформлення пояснювальної записки та документів до неї.               | 9-10 тижні                              | Додатки      |
| 8     | Нормоконтроль та рецензування.                                       | 11 тиждень                              |              |
| 9     | Захист роботи.                                                       | 12 тиждень                              |              |

Студент(ка)

\_\_\_\_\_  
( підпис )      Олег ФАЛАЛЄЄВ  
(Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

\_\_\_\_\_  
( підпис )      Андрій ОЛІЙНИК  
(Ім'я ПРІЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:  
89 с., 3 табл., 34 рис., 3 дод., 19 джерел.

C#, MICROSOFT VISUAL STUDIO, ЗБИРАННЯ ІНФОРМАЦІЇ, МОВА ПРОГРАМУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ТОВАР, ЦІНА.

Об'єкт дослідження – процес розробки програмного забезпечення для збору інформації про товари.

Предмет дослідження – програмні засоби для підтримки збору інформації про товари.

Мета роботи – дослідження та розробка програмного забезпечення для збору інформації про товари.

Матеріали, методи та технічні засоби: мова програмування C#, середовище розробки Microsoft Visual Studio.

Результати. Розроблено програмне забезпечення для збору інформації про товари за допомогою мови програмування C# та середовища розробки Microsoft Visual Studio.

Практична цінність роботи полягає у розробці програмного забезпечення, що забезпечує для збирання інформації про товари.

Висновки. Виконано проєктування програмного забезпечення для збору інформації про товари. Розроблено програмне забезпечення для збору інформації про товари. Здійснено тестування розробленого програмного забезпечення для збору інформації про товари.

Галузь використання – агрегатори цін про товари.

## **ABSTRACT**

Explanatory note to the diploma qualifying work of the master: 89 pages, 3 tables, 34 figures, 3 appendixes, 19 sources.

C#, MICROSOFT VISUAL STUDIO, INFORMATION COLLECTION, PROGRAMMING LANGUAGE, SOFTWARE, PRODUCT, PRICE.

The object of research is the process of developing software for collecting information about goods.

The subject of the research is software for collecting information about goods.

The purpose of this work is to develop the software for collecting information about goods.

Materials, methods and technical tools: C# programming language, Microsoft Visual Studio development environment.

Results. The software tools for collecting product information using the C# programming language and the Microsoft Visual Studio development environment have been developed.

The practical value of the work lies in the development of software for collecting product information.

Conclusions. The software for collecting product information has been designed. The software for collecting product information has been developed. The developed software for collecting product information has been tested.

Scope of use – aggregators of prices for goods.

## ЗМІСТ

|                                                                                                              |    |
|--------------------------------------------------------------------------------------------------------------|----|
|                                                                                                              | С. |
| Перелік скорочень та умовних позначок .....                                                                  | 8  |
| Вступ.....                                                                                                   | 9  |
| 1 Аналіз предметної області.....                                                                             | 11 |
| 1.1 Програмне забезпечення для збору інформації про товари .....                                             | 11 |
| 1.2 Аналіз програмного забезпечення для збору інформації про товари .....                                    | 20 |
| 1.3 Висновки за розділом 1 .....                                                                             | 32 |
| 2 Матеріали і методи .....                                                                                   | 34 |
| 2.1 Вибір мови програмування .....                                                                           | 34 |
| 2.2 Вибір середовища розробки для створення програмного забезпечення<br>для збору інформації про товари..... | 37 |
| 2.3 Розробка моделі взаємодії користувача з програмним забезпеченням<br>для збору інформації про товари..... | 40 |
| 2.4 Висновки за розділом 2 .....                                                                             | 42 |
| 3 Опис програми .....                                                                                        | 43 |
| 3.1 Структура програмного забезпечення для збору інформації про товари.....                                  | 43 |
| 3.2 Функціонування програмного забезпечення для збору інформації про<br>товари .....                         | 45 |
| 3.3 Розробка бази даних програмного забезпечення для збору інформації<br>про товари.....                     | 47 |
| 3.4 Проєктування інтерфейсу програмного забезпечення для збору<br>інформації про товари .....                | 49 |
| 3.5 Висновки за розділом 3 .....                                                                             | 53 |
| 4 Експлуатація, тестування та експериментальне дослідження програми.....                                     | 54 |
| 4.1 Призначення й умови застосування програми .....                                                          | 54 |
| 4.2 Характеристики програми для збору інформації про товари .....                                            | 55 |
| 4.3 Інструкція по експлуатації програми.....                                                                 | 56 |
| 4.3.1 Звернення до програми .....                                                                            | 56 |
| 4.3.2 Вхідні й вихідні дані.....                                                                             | 56 |

|                                                                                 |    |
|---------------------------------------------------------------------------------|----|
| 4.3.3 Повідомлення .....                                                        | 56 |
| 4.4 Виконання програмного забезпечення для збору інформації про<br>товари.....  | 57 |
| 4.5 Тестування програмного забезпечення для збору інформації про<br>товари..... | 61 |
| 4.6 Висновки за розділом 4 .....                                                | 61 |
| Висновки .....                                                                  | 62 |
| Перелік джерел посилань .....                                                   | 65 |
| Додаток А Технічне завдання .....                                               | 67 |
| Додаток Б Фрагмент тексту програми .....                                        | 71 |
| Додаток В Слайди презентації .....                                              | 82 |

## **ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК**

API – Application Programming Interface;

MVC – Model-View-Controller;

БД – база даних;

ЗІ – збирання інформації;

ПЗ – програмне забезпечення.

## ВСТУП

Актуальність програмного забезпечення для збору інформації про товари зростає у зв'язку з кількома ключовими тенденціями. З розвитком Інтернет торгівлі та зростанням конкуренції серед онлайн майданчиків програмне забезпечення для збору та аналізу даних про товари стало необхідністю для відстеження цін, наявності, характеристик та відгуків на товари [1]-[6].

Програмні засоби для збору інформації дозволяють компаніям значно зекономити час і ресурси, отримуючи дані в реальному часі та швидко реагуючи на зміни ринку. Таке програмне забезпечення у деяких випадках допомагає компаніям прогнозувати тренди, розуміти вподобання споживачів та адаптувати свою продукцію або стратегію продажів [1]-[6].

Програмні інструменти для збору даних дозволяють підприємствам відстежувати дії конкурентів, їх ціни, новинки товарів та рекламні кампанії, що допомагає покращувати власні продукти та маркетингові стратегії [1]-[6].

Проте використання у програмних засобах збору інформації про товари ненадійних джерел інформації, що надають неактуальні або неточні дані про товари, може призвести до хибних висновків або неправильних рішень. Крім того, впровадження та налаштування під конкретні вимоги торгівельної організації деякого існуючого програмного забезпечення для збору даних про товари може вимагати значних фінансових інвестицій, особливо якщо йдеться про складні програмні системи з інтеграцією штучного інтелекту або машинного навчання. В деяких випадках інтеграція з різними програмними платформами та системами може бути складною та вимагати спеціальних технічних знань, також можливі проблеми сумісності з різними джерелами даних або іншими бізнес-системами [1]-[6].

Тому актуальною є розробка програмного забезпечення для збору інформації про товари. У дипломній кваліфікаційній роботі магістра

розв'язується актуальне завдання дослідження та розробки програмного забезпечення для збору інформації про товари.

Для досягнення поставленої мети у кваліфікаційній роботі магістра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для збору інформації про товари;
- розробити модель взаємодії користувача з програмним забезпеченням для збору інформації про товари;
- здійснити проєктування програмного забезпечення для збору інформації про товари;
- створити програмне забезпечення для збору інформації про товари;
- виконати тестування розробленого програмного забезпечення для збору інформації про товари.

## **1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

### **1.1 Програмне забезпечення для збору інформації про товари**

Збір даних з вебсайтів полягає у видобуванні інформації з відкритих онлайн джерел та збереженні її в базах даних. Цей процес зазвичай здійснюється за допомогою скриптів або програм, які можуть обходити захисні механізми вебсайтів. Однак навіть якщо вебсайт змінить свій код чи захист, це не робить збір даних незаконним, оскільки публічна інформація, доступна в Інтернеті, може бути використана. Проте отримання закритої або конфіденційної інформації може бути проблемою, оскільки вона може бути захищена правилами доступу або іншими юридичними обмеженнями. У цьому контексті ми досліджуємо лише збір публічних даних [1]-[7].

Існують різні методи збору даних, зокрема форми зворотного зв'язку, опитування в соціальних мережах або онлайн дослідження. Такі інструменти дозволяють компаніям залучати аудиторію та отримувати зворотний зв'язок від клієнтів або співробітників. Зібрані дані можуть бути як структурованими, так і неструктурованими, що ускладнює їх обробку, особливо якщо вони вимагають більш якісного аналізу [1]-[7].

Програмні засоби, зокрема бази даних, є основним вибором для зберігання великих обсягів даних, які щодня генеруються в Інтернеті. Інструменти машинного навчання часто використовуються для автоматизації цього процесу. Однак збір даних потребує постійного оновлення, щоб інформація залишалася актуальною, що займає багато часу. У цьому полягає складність і трудомісткість процесу для компаній, але деякі компанії спеціалізуються на управлінні даними, дозволяючи клієнтам зосередитися на бізнес-цілях, а не на технічних проблемах [1]-[7].

Відстеження даних онлайн є відмінним методом збору інформації від відвідувачів вебсайтів. Коли користувач заходить на вебсайт, він може залишити багато даних, особливо якщо цікавиться вашими продуктами або послугами, або підписується на розсилку. Отримана інформація може

включати як демографічні дані, так і контактні деталі, такі як електронна пошта або номер телефону [1]-[7].

Цей процес також допомагає поліпшити роботу сайту, оскільки можна аналізувати клієнтські потоки, оцінювати ефективність різних розділів сайту, дізнаватися, з яких пристроїв (мобільних чи комп'ютерів) приходить більше користувачів та відслідковувати важливі метрики [1]-[7].

Щодо програмних інструментів збору даних, вони значно швидші й ефективніші, ніж ручний збір інформації. Проте, в окремих випадках, коли потрібні лише певні обмежені дані, наприклад, кількість технологічних компаній у конкретному місті, які не потребують постійного оновлення, можна зібрати інформацію самотійно, не вдаючись до використання складних програмних рішень. Однак, якщо потрібні регулярні оновлення та доступ до більш масштабних даних, вигідніше використовувати послуги програм-постачальників даних або центрів обробки даних [1]-[7].

Основні принципи збору даних онлайн включають кілька важливих аспектів. Перш за все, необхідно чітко визначити мету збору даних, оскільки це допоможе налаштувати алгоритми так, щоб вони отримували тільки релевантну інформацію, яка відповідає бізнес-цілям. Важливо також вирішити, чи потрібні вам якісні або кількісні дані, що дозволить глибше аналізувати і вносити потрібні зміни. Конвеєр даних забезпечує перенесення нової інформації від джерела до місця призначення, тому варто заздалегідь обрати базу даних, де ці дані будуть зберігатися. При зборі даних можна використовувати як внутрішні бізнес-дані про клієнтів, так і загальнодоступні джерела, наприклад, соціальні мережі чи бізнес-сайти. Після збору даних потрібно оцінити їх якість, легітимність і відповідність алгоритмам штучного інтелекту чи машинного навчання. Це включає аналіз моделей помилок, перевірку наявності технічних проблем, дублікатів, а також збалансованість алгоритму. Збір даних має бути постійним процесом, адже важливо не лише зібрати інформацію, але й регулярно її оновлювати. Тому, щоб досягти бізнес-

цілей, необхідно правильно визначити потреби, отримати відповідні дані й використовувати їх для прийняття ефективних рішень [1]-[7].

Інтернет став потужним джерелом інформації, де доступні величезні обсяги даних, включаючи інформацію з соціальних мереж, електронної комерції, нерухомості та сайтів новин. Багато компаній активно присутні онлайн, і збір даних із вебресурсів вручну є неефективним та трудомістким. При наявності величезної кількості інформації автоматизація процесу збору є необхідною. Наприклад, багато онлайн продавців використовують динамічне ціноутворення, відстежуючи запаси та ціни конкурентів, тоді як компанії з нерухомості і маркетингові агентства отримують цінні дані з соціальних мереж та інших ресурсів. Для великомасштабного збору даних найкращим рішенням є написання спеціального коду для збирання даних, що дозволяє ефективно збирати дані з кількох сайтів одночасно. Наприклад, якщо компанія займається ціноутворенням, їй потрібно відстежувати інформацію з різних сайтів електронної комерції, використовуючи спеціальні агенти та сервери для уникнення блокувань. У деяких випадках, залежно від адреси, сайти показують різну інформацію, тому важливо використовувати відповідні методи для збору даних з різних регіонів [1]-[7].

Веббраузери стають дедалі важливішим інструментом для сучасних програм. Багато сайтів зараз розробляються з використанням передових фреймворків, таких як Vue.js, Angular.js та React.js. Ці фреймворки використовують Javascript для динамічного завантаження та відображення даних через об'єктну модель документа на стороні клієнта. Ще одним викликом при зборі даних є те, що багато сайтів використовують спеціальні засоби, щоб відрізнити автоматизовані клієнтські запити від реальних користувачів. Проте деяке програмне забезпечення дозволяє обійти такі тести та отримати необхідний вміст [1]-[7].

Для вилучення даних із HTML-елементів використовуються бізнес-правила, які визначають, як вибрати потрібні елементи. Найбільш популярні

методи вибору елементів полягають використанні відповідних параметрів та засобів [1]-[7].

Основною причиною збору інформації з Інтернет ресурсів є те, що співробітники витрачають на це багато часу. Вебсайти, особливо ті, що недавно створені, часто оновлюють свою HTML-структуру, тому потрібно регулярно оновлювати параметри збору даних з тих сайтів. Планування у процесі збору даних також має важливе значення, наприклад, можна налаштувати збір даних на щоденній або щотижневій основі. Перевага планування завдань полягає в тому, що можна відстежити невдалі завдання. Обробка помилок у процесі збору даних має велике значення, оскільки не всі помилки легко контролюються [1]-[7].

Серед можливих проблем при зборі даних можна відзначити такі: HTML-структура сторінки змінилася і більше не відповідає критеріям збору даних, цільова область вебсайту може стати недоступною, проксі-сервер може працювати повільно або нестабільно [1]-[7].

Ці проблеми можна мінімізувати за допомогою правильного планування та інструментів для обробки помилок. Для цього можна використовувати командні інструменти або бібліотеки для планування завдань [1]-[7].

Щодо зберігання даних: після збору даних з вебсайтів, їх зазвичай необхідно зберігати у відповідному форматі. Зібрані дані найчастіше зберігаються в таких стандартних форматах. Моніторинг процесу збору даних є надзвичайно важливим. Як згадувалося раніше, під час збору інформації можуть виникнути різні проблеми. Важливо переконатися, що бази даних заповнені, а програмні агенти працюють належним чином [1]-[7].

Існують рішення для збору даних, які не потребують програмування або використовують мінімальний код через застосування засобів прикладного програмного інтерфейсу (Application Programming Interface, API). Це ефективний підхід, коли необхідно отримати дані одноразово, а не регулярно. Якщо потрібен великий обсяг даних з Інтернету, важливо спочатку

перевірити, чи не існує вже готової бази даних. Наприклад, якщо потрібен перелік усіх вебсайтів, які розв'язують специфічну задачу, це вимагатиме значних ресурсів для збору таких даних через сканування Інтернету чи каталоги. Проте, якщо така база даних вже існує, наприклад, як послуга, її можна знайти та придбати через наявні джерела [1]-[7].

Що стосується збору даних з конкретного вебсайту, то використання API може бути значно ефективнішим рішенням. Наприклад, API дозволяє автоматизувати процес збору даних, знижуючи необхідність у підтримці, особливо якщо вебсайт оновлює свій HTML-код. Це виключає потребу постійно перевіряти зміни на сайті чи оновлювати свої програмні інструменти збору даних [1]-[7].

Ще один важливий аспект полягає у тому, що перш ніж починати збір даних, варто перевірити, чи не надає сам сайт публічний або приватний API. Це значно полегшить задачу, оскільки існує можливість отримати потрібну інформацію легально і швидше, без необхідності використовувати спеціальні програми або складні методи збору даних [1]-[7].

Цей підхід з API також може бути актуальним для проєктів, які пов'язані з великими даними та аналізом вебконтенту за допомогою засобів інтелектуального аналізу даних і їх попередньої обробки [1]-[7].

Вбудований браузер може стати зручним інструментом для збору інформації з вебсайтів. Це особливо корисно для отримання структурованих даних, таких як таблиці або списки з локальних сторінок. Деякі розширення пропонують готові рішення для збору даних з популярних ресурсів [1]-[7].

Інструменти для збору даних з вебсайтів чудово підходять для видобування інформації. Вони можуть бути корисними для різних цілей, таких як пошукова оптимізація, дослідження ключових слів або пошук реальних посилань [1]-[7].

Також існують зручні програми для тих, хто не володіє навичками програмування. Це настільні програми, які дозволяють легко витягувати дані з

Інтернету. Користувач може просто вибирати потрібні елементи та переглядати результати прямо в програмі [1]-[7].

Однак ці методи можуть мати свої обмеження та потребують певного часу на освоєння [1]-[7].

Деякі компанії та незалежні постачальники послуг займаються видобуванням даних з Інтернет ресурсів і можуть допомогти вирішити певні завдання з видобування даних. Використання таких сервісів може стати хорошим варіантом, якщо інші рішення не підходять для розв'язання певної задачі [1]-[7].

Програмне забезпечення для збору даних перетворює текст та дані, представлені у форматі HTML, у придатні для використання формати, такі як таблиці або бази даних. Це дозволяє аналізувати дані для подальшого використання в бізнес-додатках. Використання API або автоматизованих інструментів для збору контенту та даних може значно скоротити час, необхідний для ручної роботи, завдяки розумній автоматизації [1]-[7].

Щоб зібрати інформацію, потрібні два компоненти: кроулер і скрепер. Кроулер є програмним інструментом на базі штучного інтелекту, що шукає в Інтернеті специфічну інформацію. Він може використовувати HTTP або браузер, такі як Chrome, для збору даних з вебсторінок. Скрепер — це інструмент, що безпосередньо витягує ці дані з вебсайтів [1]-[7].

Законодавство різних країн накладає певні обмеження на те, що можна збирати, але компанії часто використовують такі дані з вебсайтів: тексти, зображення, візуальний контент, інформація про продукти, дані користувачів та податкову інформацію, відгуки з соціальних мереж, ціни для порівняння товарів у різних магазинах [1]-[7].

Залежно від рівня складності використання та технологій, які застосовуються, програмне забезпечення для збирання даних можна розділити на п'ять основних типів [1]-[7]. Браузери з вбудованими програмними модулями потребують високих навичок програмування, але надають більше можливостей для налаштування. Готові рішення для збору даних створені для

швидкого використання, що дозволяє легко розпочати роботу. Розширення для браузерів пропонують зручні інструменти, часто з унікальними функціями для збору даних [1]-[7]. Хмарні браузери використовують віддалені сервери, що дозволяє зберегти ресурси вашого комп'ютера та працювати над іншими завданнями. Локальні браузери використовують апаратні ресурси комп'ютера, такі як процесор та оперативна пам'ять, для виконання завдань [1]-[7].

Проаналізуємо методи збору даних з вебсайтів. Ручне копіювання та вставка – процес, коли людина вручну копіює дані з вебсайтів у текстовий документ або інші формати. Використання регулярних виразів – автоматизація збору даних за допомогою програмних інструментів для роботи з регулярними виразами. Використовують інші мови програмування, як-от Python, Ruby, Java, C++ та JavaScript. Вбудований у браузери механізм для аналізу динамічних даних, які генеруються клієнтськими документами. Розпізнавання семантичних анотацій – пошук та збір даних на основі метаданих або анотацій. Комп'ютерний зір – метод, що використовує машинне навчання та технології комп'ютерного зору для збору інформації з вебсторінок [1]-[7].

Мова розмітки визначає структуру вмісту вебсайту. Оскільки існують загальні стандарти для мовних компонентів і кодів, аналітикам набагато легше збирати необхідну інформацію. У процесі збору та аналізу даних синтаксичний аналіз за допомогою HTML є досить важливим, потім нейронна мережа виявляє та зберігає необхідну інформацію. Також використовуються засоби API, які дозволяють двом програмам працювати разом для збору даних [1]-[7].

Процес збору даних починається зі створення HTTP-запиту. Для цього використовується, наприклад, браузер, який може працювати на сервері з даними. Після цього важливо витягнути та перевірити код сайту, щоб визначити структуру його вмісту. Далі програмне забезпечення для збору даних аналізує код, перевіряючи його на наявність попередньо визначених об'єктів, таких як таблиці, графіки, класи, теги або ідентифікатори, і видаляє їх при необхідності. Після завершення аналізу дані зберігаються локально у

форматі XML або HTML. Ці етапи складають основи аналізу роботи, але важливо пам'ятати, що надмірна кількість HTTP-запитів може призвести до збою сайту. Тому програмні боти для збору даних повинні дотримуватися різних правил щодо очікування [1]-[7].

На початковому етапі збору даних з вебсайту в ручному режимі відбувається відкриття сторінки, клацнувши правою кнопкою миші та вибравши опцію перевірки елемента. Після цього знаходиться необхідна інформація, звернувши увагу на теги div або інші елементи, що містять дані. Далі налаштовується програмне забезпечення для пошуку цих елементів і вилучення інформації, використовуючи програмні бібліотеки. Після цього програмне забезпечення для отримання та аналізу даних запускається, дані збираються, а потім вони зберігаються у різних форматах [1]-[7].

Проаналізуємо застосування програмного забезпечення для збору та аналізу даних [1]-[7].

У цифровому маркетингу програмні засоби можуть збирати контактні дані, як-от номери телефонів та електронні адреси, що знаходяться в загальному доступі в Інтернеті. Хоча такі дані не завжди можуть бути на точними, цей метод є досить ефективним для швидкого отримання результатів [1]-[7].

Порівняння цін на різні товари є важливим етапом для тих, хто прагне знайти вигідні пропозиції. Якщо ціни на товари або послуги встановлюються в Інтернеті, існують інструменти, що дозволяють отримувати порівняння цін і знижок у реальному часі [1]-[7].

Одним із найефективніших застосувань засобів збору даних є оптимізація для пошукових систем. Пошукові системи мають різні правила для індексації результатів за ключовими словами, деякі з яких є більш критичними, ніж інші. Спеціалізоване програмне забезпечення сканує Інтернет та аналізує вміст, щоб визначити, як домінувати в пошукових запитах. Використання цих маркетингових підходів у щоденній стратегії є корисним [1]-[7].

Також варто зазначити, що більш розвинені сервіси збирання даних здатні проводити аналіз даних. Компанії використовують соціальні мережі для збору та публікації інформації в реальному часі з платформ. Ці дані можна використовувати для аналізу кількісних показників, таких як коментарі і згадування, а також для якісного аналізу, включаючи настрої бренду та порівняння контенту [1]-[7].

Програмне забезпечення для пошукових систем здійснює сканування Інтернету, аналізує та порівнює вміст, щоб визначити, які сайти мають перевагу в результатах пошуку. Ці маркетингові стратегії можуть бути корисними для впровадження у щоденну діяльність [1]-[7].

Програмні засоби для збору даних мають свої недоліки, і користувачі Інтернету можуть стикатися з кількома обмеженнями. По-перше, хоча інструменти аналітики спрощують процес збору даних, необхідно витратити час на навчання, щоб повністю освоїти їх використання [1]-[7].

Крім того, структура та склад даних можуть варіюватися, оскільки розробники постійно вдосконалюють свої сайти для покращення користувацького досвіду. Навіть незначні зміни в дизайні можуть вплинути на інформацію, яку вдається зібрати [1]-[7].

Складні вебсторінки часто потребують спеціалізованих навичок для доступу до динамічного вмісту. Крім технічних обмежень, деякі сайти впроваджують політики безпеки, використовуючи власні алгоритми та шифрування для захисту свого контенту [1]-[7].

Щодо програмного забезпечення для аналізу даних, системи інтелектуального аналізу допомагають виявляти як неструктуровані, так і структуровані дані для бізнес-аналітики. Ці інструменти використовують якісні медіа та засоби обробки даних, щоб упорядкувати та очистити інформацію. Крім того, компанії використовують бази даних для об'єднання різних типів даних і джерел в одному місці [1]-[7].

Визначено, що збір даних з вебсайтів полягає у видобуванні інформації з відкритих онлайн джерел та збереженні її в базах даних. Цей процес зазвичай

здійснюється за допомогою скриптів або програм, які можуть обходити захисні механізми вебсайтів. Проаналізовано різні методи збору даних з вебсайтів. Проаналізовано застосування програмного забезпечення для збору та аналізу даних.

## 1.2 Аналіз програмного забезпечення для збору інформації про товари

Проаналізуємо програмне забезпечення для збору інформації про товари [8]-[14].

Програмне забезпечення Diffbot (рис. 1.1) з його потужними алгоритмами здатне виконувати майже будь-який онлайн запит для збору даних. Використовуючи цю платформу, можна розробляти ефективні методи отримання інформації з різних джерел, таких як блоги, прес-релізи, веб-сторінки, сторінки товарів, форуми та відгуки користувачів [8], [10], [12].

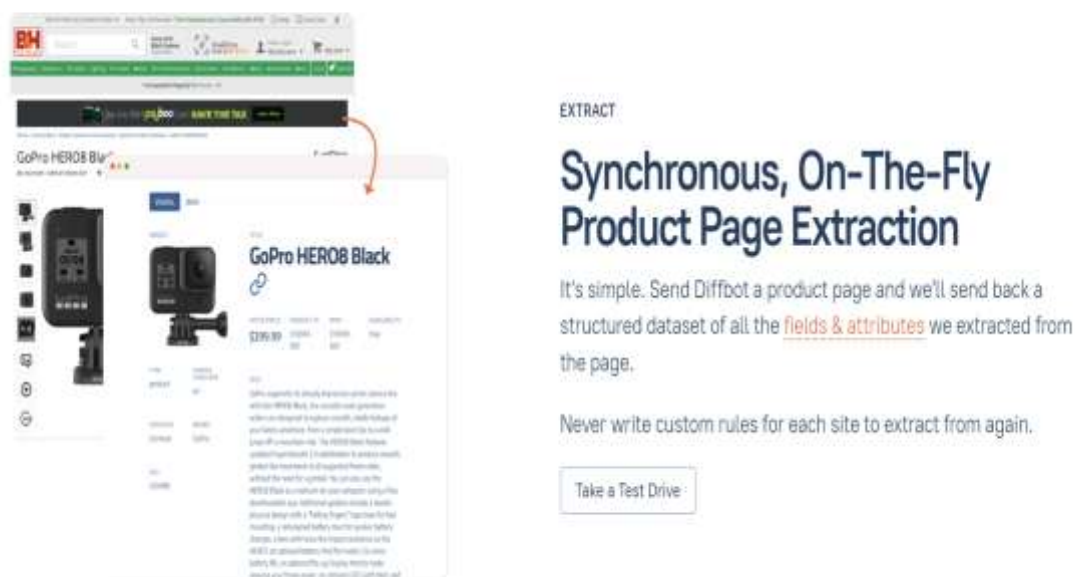


Рисунок 1.1 – Програмне забезпечення Diffbot [12]

Програмне забезпечення Diffbot надає можливість повністю контролювати вибір полів даних при зборі. Наприклад, при аналізі конкурентів можна легко отримати дані про зображення товарів, їх ціни, характеристики, кольори, кількість та інші параметри. Так само під час аналізу компаній можна збирати інформацію про їхні доходи, географічне розташування та інвестиції [8], [10], [12].

Завдяки своїм інтелектуальним мовним моделям Diffbot не тільки збирає дані, а й аналізує контекст, що дозволяє отримувати більш глибоку та корисну інформацію для бізнесу. Однією з ключових переваг цього інструменту є його стійкість до змін у дизайні веб-сайтів, що робить його більш надійним у порівнянні з іншими рішеннями [8], [10], [12].

Diffbot є одним із найпотужніших інструментів для автоматизації збирання даних з вебсайтів завдяки своїй здатності адаптуватися до різноманітних структур вебсторінок. Використовуючи технології машинного навчання та обробки природної мови, Diffbot автоматично розпізнає та структурує інформацію з вебсайтів без необхідності в ручному налаштуванні. Це робить платформу особливо корисною для компаній, яким потрібно збирати дані з різних джерел і форматів [8], [10], [12].

Ще однією перевагою Diffbot є його універсальність. Він може бути інтегрований у різні бізнес-системи та платформи, надаючи можливість автоматично оновлювати бази даних та виконувати складні аналітичні задачі. Замість того, щоб вручну налаштовувати збирання даних для кожного окремого сайту, Diffbot автоматично адаптується до змін на сторінках, включаючи зміни у верстці, форматуванні чи структурі [8], [10], [12].

Diffbot також забезпечує високу точність при роботі з даними, що робить його ідеальним рішенням для бізнесу, де необхідна достовірність інформації. Він може використовуватися в різних сферах, таких як електронна комерція, фінансовий аналіз, маркетингові дослідження та навіть у наукових дослідженнях. Завдяки масштабованій архітектурі, Diffbot здатен обробляти

великі обсяги даних у короткі терміни, що робить його потужним інструментом для великих підприємств [8], [10], [12].

Основними характеристиками програмного забезпечення Diffbot є такі [8], [10], [12]:

- використання засобів штучного інтелекту та машинного навчання: програмне забезпечення Diffbot використовує алгоритми машинного навчання для автоматичного розпізнавання різних типів вмісту на вебсторінках. Завдяки цьому програмне забезпечення може коректно розпізнавати текст, зображення, відео, таблиці та інші елементи сторінок без необхідності налаштування під кожен сайт [8], [10], [12];

- можливість обробки природної мови: завдяки використанню засобів обробки природної мови програмне забезпечення Diffbot здатне аналізувати текст на вебсторінках, розуміючи контекст і зміст інформації. Це дає змогу не лише збирати дані, а й отримувати більш глибоке розуміння змісту, що підходить для аналітичних досліджень і бізнес-аналітики [8], [10], [12];

- масштабованість: програмне забезпечення Diffbot може обробляти великі обсяги даних одночасно. Це дозволяє збирати дані з тисяч вебсторінок за короткий час, що є особливо корисним для великих компаній, які працюють з масштабними даними [8], [10], [12];

- гнучкість у налаштуванні: програмне забезпечення Diffbot надає користувачам повний контроль над тим, які поля та типи даних збирати. Це дозволяє адаптувати його під конкретні завдання, наприклад, витягати ціни, зображення продуктів, описи або інші характеристики [8], [10], [12];

- засоби автоматизації: програмне забезпечення Diffbot забезпечує повністю автоматизоване збирання вебданих, завдяки чому відпадає необхідність у постійному ручному налаштуванні або коригуванні під змінені структури сайтів [8], [10], [12];

- стійкість до змін у верстці вебсторінок: на відміну від багатьох інших інструментів для збору даних, Diffbot не сильно залежить від змін у дизайні

або структурі сайтів. Він продовжує працювати ефективно навіть тоді, коли веб-сторінки оновлюються або змінюються [8], [10], [12];

- API для інтеграції: Diffbot надає потужний API, який дозволяє легко інтегрувати його з іншими програмними системами, такими як маркетингові платформи або аналітичні інструменти. Це дозволяє автоматично імпортувати зібрані дані у внутрішні системи компаній [8], [10], [12];

- структурований вивід даних: зібрані за допомогою програмного забезпечення Diffbot дані можуть бути експортовані в популярні формати, такі як JSON, CSV, або Excel, що дозволяє легко використовувати їх у подальшому аналізі [8], [10], [12].

Серед недоліків програмного забезпечення Diffbot можна відзначити обмежені можливості для роботи з деякими динамічними сайтами: попри високий рівень автоматизації, Diffbot іноді стикається з труднощами при зборі даних із динамічних сайтів. У таких випадках може знадобитися додаткове налаштування або використання інших програмних інструментів. Крім того, хоча інтерфейс Diffbot є досить зрозумілим, для максимально ефективного використання програмного забезпечення потрібні базові знання програмування та розуміння API. Це може стати викликом для користувачів без технічної підготовки [8], [10], [12].

Програмне забезпечення Webscraper.io (рис. 1.2) представляє собою потужний програмний інструмент для збору даних, що пропонує два основні варіанти: розширення для Chrome і хмарне рішення. Використовуючи розширення Webscraper.io для Chrome, можна створити схему сайту для навігації по веб-сторінках і вказати, які саме дані слід зібрати. Хмарне розширення Webscraper.io дозволяє обробляти великі обсяги даних і виконувати декілька процесів сканування одночасно [8]-[10], [13].

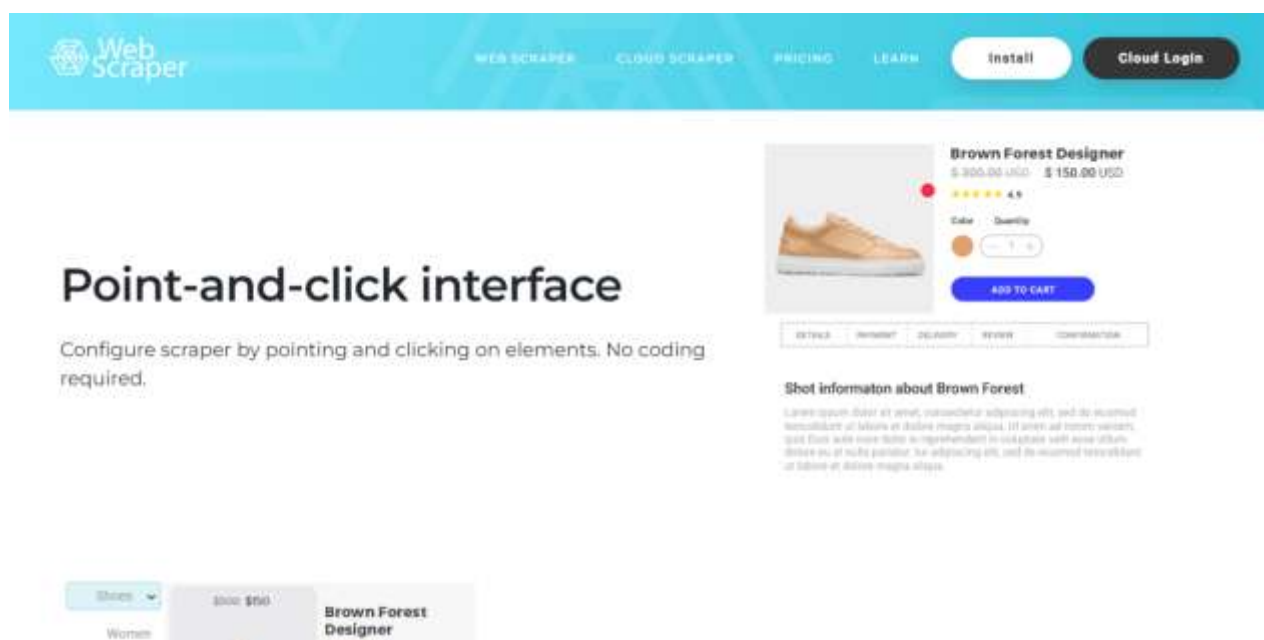


Рисунок 1.2 – Програмне забезпечення Webscraper.io [13]

Цей програмний інструмент є одним із найпоширеніших розширень для браузерів Chrome і Firefox для збору даних без потреби у програмуванні. Його можливості дозволяють динамічно витягувати дані у популярних форматах. Крім того, для тих, хто потребує розширеного функціоналу, хмарна версія програмного забезпечення пропонує сервіси, можливості для аналізу та планування збору даних [8]-[10], [13].

Webscraper.io відзначається простотою використання, пропонуючи інтуїтивно зрозумілий інтерфейс, де за допомогою миші можна налаштувати процес збору даних без жодних знань програмування. Цей інструмент добре підходить як для простих, так і для складних завдань, що можуть включати обробку JavaScript, глибоку навігацію по сайтах та інші вимоги. Додатково користувачі можуть обрати платні підписки для отримання більш широкого функціоналу [8]-[10], [13].

Програмне забезпечення Webscraper.io широко використовується як окремими користувачами, так і компаніями для вилучення великих обсягів даних. Він здатен обробляти різноманітну інформацію, включаючи текст, зображення та таблиці. Хоча цей інструмент здобув як позитивні, так і

негативні відгуки, його надійність і ефективність варто оцінити перед впровадженням у конкретний проект [8]-[10], [13].

Окрім цього, Webscraper.io підтримує використання серверів, за замовчуванням застосовуючи IP-адреси з провідної країни. При необхідності можна зв'язатися зі службою підтримки для отримання IP-адрес з інших країн, або ж підключити сторонні сервіси [8]-[10], [13].

Для зручності користувачів програмне забезпечення Webscraper.io надає різноманітні навчальні матеріали, включаючи блоги, документацію, відео уроки та форум. Також є можливість звернутися до служби підтримки через електронну пошту [8]-[10], [13].

Інтеграція з іншими сервісами, такими як Dropbox, Google Sheets або Amazon S3, забезпечує додаткові можливості для керування даними через API. Проте, користувачі часто повідомляють про певні проблеми зі швидкістю роботи та стабільністю під час обробки великих обсягів даних або кількох вебсторінок [8]-[10], [13].

Розширені функції програмного забезпечення Webscraper.io включають планувальник, синтаксичний аналізатор, підтримку JavaScript, доступ до API та можливість налаштовувати карти сайту для ефективнішого збору необхідних даних [8]-[10], [13].

Програмне забезпечення Webscraper.io має такі особливості [8]-[10], [13]:

- простота використання: програмне забезпечення Webscraper.io має інтуїтивний інтерфейс, що дозволяє навіть новачкам швидко налаштувати процес збору даних без потреби в написанні коду [8]-[10], [13];

- хмарне рішення: програмне забезпечення Webscraper.io пропонує хмарне розширення, яке здатне збирати великі обсяги даних та виконувати паралельно кілька завдань сканування [8]-[10], [13];

- підтримка різних форматів даних: програмне забезпечення Webscraper.io дозволяє збирати дані у різних форматах, таких як CSV, Excel, JSON, що робить його гнучким для різних аналітичних потреб [8]-[10], [13];

– масштабованість: програмне забезпечення Webscraper.io підходить як для простих завдань, так і для більш складних завдань, що потребують виконання JavaScript або багатоетапної навігації по вебсайтах. Також є можливість переходу з безкоштовної версії на платні плани для доступу до додаткових функцій [8]-[10], [13];

– підтримка проксі: програмне забезпечення підтримує використання проксі для обходу обмежень та збору даних з різних географічних місць, що корисно для збору інформації зі світових ринків [8]-[10], [13];

– інтеграція з іншими сервісами: програмне забезпечення Webscraper.io легко інтегрується з популярними хмарними сховищами, такими як Dropbox, Google Sheets і Amazon S3, а також надає засоби API для доступу до зібраних даних [8]-[10], [13];

– швидкість та продуктивність: незважаючи на свою зручність, програмне забезпечення Webscraper.io має деякі обмеження в продуктивності, особливо при зборі великих обсягів даних. Користувачі можуть стикатися з проблемами швидкості та випадковими збоями при скануванні великих веб-сторінок [8]-[10], [13];

– розширені функції: програмне забезпечення включає в себе такі функції, як планувальник завдань, синтаксичний аналізатор, можливість налаштування карт сайту, підтримка JavaScript та вебхуки для автоматизації збору даних [8]-[10], [13];

– підтримка та спільнота: Webscraper.io пропонує доступ до блогу, форуму, відеоуроків і документації, що допомагає користувачам навчатися і вирішувати проблеми [8]-[10], [13].

Серед недоліків програмного забезпечення Webscraper.io можна відзначити обмежені можливості для роботи з деякими динамічними сайтами: попри високий рівень автоматизації, Webscraper.io іноді стикається з труднощами при зборі даних із динамічних сайтів. У таких випадках може знадобитися додаткове налаштування або використання інших програмних інструментів. Крім того, хоча інтерфейс Webscraper.io є досить зрозумілим,

для максимально ефективного використання програмного забезпечення потрібні базові знання програмування та розуміння API. Це може стати викликом для користувачів без технічної підготовки [8]-[10], [13].

Програмне забезпечення Apify (рис. 1.3) є хмарною платформою для збору даних, яка надає готові інструменти для вилучення інформації з вебсторінок, а також пропонує можливість створювати індивідуальні рішення для збору даних. Ця платформа дозволяє легко створювати вебсканери без потреби програмування, автоматизувати процес збору даних і керувати ними через інтуїтивний інтерфейс [8]-[10], [14].

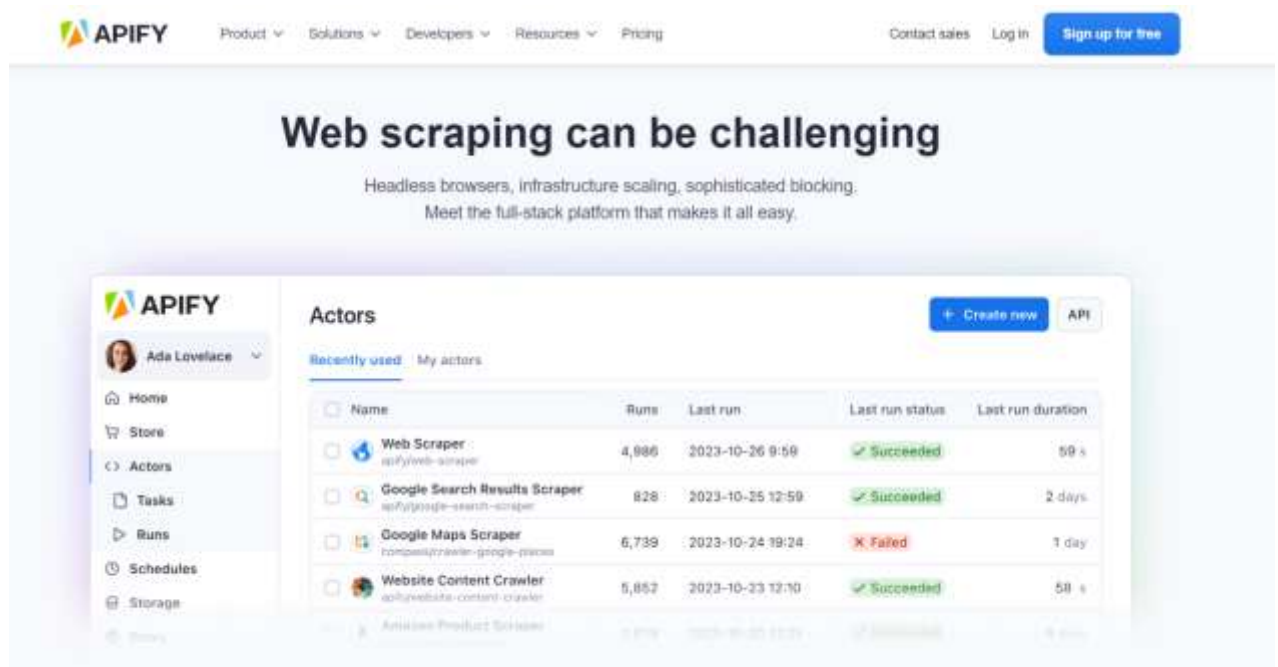


Рисунок 1.3 – Програмне забезпечення Apify [14]

Apify здобула популярність завдяки функції Apify Store, яка пропонує понад велику кількість попередньо створених API. Ці API охоплюють популярні сервіси, як-от Instagram, Twitter, Google Maps, що значно спрощує роботу з цими платформами. Крім того, користувачі можуть створювати власні API, якщо стандартні рішення не відповідають їхнім потребам. Платформа також підтримує змінні сервери, надає бібліотеку для видобування

даних з відкритим кодом і командний рядок для управління завданнями щодо збирання даних [8]-[10], [14].

Arify є середовищем для створення, запуску та моніторингу інструментів для збору даних і автоматизації роботи з браузерами. Платформа забезпечує комплексне рішення для вилучення даних, дозволяючи користувачам отримувати майже будь-яку загальнодоступну інформацію з Інтернету [8]-[10], [14].

Arify також надає можливість створювати складні скребки для підтримки різних онлайн підприємств. Крім того, на платформі є можливість придбання інструментів, створених іншими користувачами, де можна знайти скребки для аналізу результатів Google, продуктів Amazon, інструменти для аналізу конкурентів за допомогою штучного інтелекту та багато інших корисних рішень [8]-[10], [14].

Для повноцінного використання цього програмного забезпечення необхідно мати певні навички програмування. Платформа підтримує роботу з популярними мовами програмування, такими як Python і JavaScript, що дає можливість створювати унікальні рішення для бізнесу. Також існує можливість застосовувати ротацію серверів для обходу захисних механізмів вебсайтів [8]-[10], [14].

З точки зору зручності використання, клієнти часто відзначають простоту Arify. Інструмент має гнучку, але стабільну функціональність, вбудовані готові до використання скребки та безліч варіантів інтеграції. Arify також відомий своїми можливостями масштабування. Програмна платформа Arify здатна задовольнити зростаючі потреби користувачів, надаючи додаткові ресурси та можливості для виконання більш складних завдань зі збору даних [8]-[10], [14].

З допомогою Arify можна збирати майже будь-які типи даних з вебсторінок по всьому світу. Це корисно для створення баз даних клієнтів, дослідження ринку, порівняння цін, аналізу настроїв тощо. Надійність Arify підтверджується позитивними відгуками користувачів. Користувачі

відзначають простоту у використанні та можливість сканування сайтів у необхідному масштабі [8]-[10], [14].

Платформа пропонує велику базу знань із детальною інформацією щодо початку роботи, інтеграцій та іншого. Підтримка клієнтів доступна через електронну пошту. Arify має широкі можливості інтеграції та сумісності, дозволяючи підключати його до практично будь-якої хмарної служби або програми [8]-[10], [14].

Продуктивність Arify зазвичай на високому рівні, хоча іноді можуть виникати затримки через збої, що характерно для багатьох безкоштовних інструментів збору даних [8]-[10], [14].

До розширених функцій платформи належать автоматичне сканування сайтів із переходом на наступні сторінки, видалення таблиць і списків, автоматизований пошук електронних адрес, а також завантаження вебсторінок з візуальним контентом. Дані можна експортувати у форматах JSON, CSV і Excel [8]-[10], [14].

Програмне забезпечення Arify має такі особливості [8]-[10], [14]:

- хмарна платформа: Arify працює у хмарному середовищі, що дозволяє користувачам автоматизувати процес збирання вебданих без потреби в управлінні фізичною інфраструктурою [8]-[10], [14];

- підтримка відомих мов програмування: платформа підтримує Python і JavaScript, що дає можливість розробляти спеціалізовані рішення для бізнесу та використовувати гнучкі алгоритми збору даних [8]-[10], [14];

- Arify Store: програмне забезпечення Arify надає магазин готових рішень, де зібрано велику кількість API для популярних сайтів, таких як Instagram, Twitter, Google Maps. Це дозволяє значно скоротити час на налаштування власних скребків [8]-[10], [14];

- масштабованість: платформа Arify здатна виконувати завдання різного рівня складності – від невеликих проєктів до великих засобів збору даних з багаторівневою логікою [8]-[10], [14];

- змінні сервери: програмне забезпечення Arify підтримує змінні сервери, що дозволяє обходити захисні системи вебсайтів, які можуть блокувати запити зі статичних IP-адрес [8]-[10], [14];

- засоби автоматизації: програмне забезпечення Arify дозволяє автоматизувати збір даних з динамічних сайтів, використовуючи браузер для обробки складних вебсторінок, що працюють з JavaScript [8]-[10], [14];

- відкритий код: Arify має бібліотеку з відкритим вихідним кодом, що дозволяє користувачам розширювати функціональність скребків або використовувати готові рішення [8]-[10], [14];

- планування завдань: програмне забезпечення Arify дозволяє планувати сканування на регулярній основі, що є зручним для отримання оновлених даних з різних вебсайтів [8]-[10], [14];

- можливості інтеграції: програмне забезпечення Arify легко інтегрується з іншими сервісами, такими як Dropbox, Google Sheets та Amazon S3, що дозволяє спрощено керувати отриманими даними [8]-[10], [14];

- експорт даних: підтримуються різні формати експорту даних, зокрема JSON, CSV та Excel [8]-[10], [14];

- гнучкість і надійність: програмне забезпечення Arify може працювати як з простими, так і зі складними веб-ресурсами, надаючи стабільні результати. Платформа Arify отримує високі оцінки за надійність від користувачів [8]-[10], [14].

Серед недоліків програмного забезпечення Arify можна відзначити обмежені можливості для роботи з деякими динамічними сайтами: попри високий рівень автоматизації, Arify іноді стикається з труднощами при зборі даних із динамічних сайтів. У таких випадках може знадобитися додаткове налаштування або використання інших програмних інструментів. Крім того, хоча інтерфейс Arify є досить зрозумілим, для максимально ефективного використання програмного забезпечення потрібні базові знання програмування та розуміння API. Це може стати викликом для користувачів без технічної підготовки [8]-[10], [14].

Порівняльну характеристику програмного забезпечення для збору інформації про товари наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння програмного забезпечення для збору інформації про товари

| Критерій порівняння                                           | Diffbot [12] | Webscraper.io [13] | Apify [14] |
|---------------------------------------------------------------|--------------|--------------------|------------|
| Використання засобів штучного інтелекту та машинного навчання | +            | —                  | +—         |
| Масштабованість                                               | +—           | +                  | +          |
| Розширення для веббраузера                                    | —            | +                  | —          |
| Гнучкість у налаштуванні                                      | +—           | +                  | +          |
| Стійкість до змін у верстці вебсторінок                       | +—           | +                  | +—         |
| API для інтеграції                                            | +            | +—                 | +          |

За результатами проведеного аналізу можна зробити висновок, що у наш час існує досить багато програмних засобів для збору інформації про товари. Проте використання у програмних засобах збору інформації про товари ненадійних джерел інформації, що надають неактуальні або неточні дані про товари, може призвести до хибних висновків або неправильних рішень. Крім того, впровадження та налаштування під конкретні вимоги торгівельної організації деякого існуючого програмного забезпечення для збору даних про товари може вимагати значних фінансових інвестицій, особливо якщо йдеться про складні програмні системи з інтеграцією штучного інтелекту або машинного навчання. В деяких випадках інтеграція з різними програмними платформами та системами може бути складною та вимагати спеціальних технічних знань, також можливі проблеми сумісності з різними

джерелами даних або іншими бізнес-системами. Тому актуальною є розробка програмного забезпечення для збору інформації про товари.

При розробці програмного забезпечення для збору інформації про товари необхідно забезпечити такі функціональні вимоги:

- відображення каталогу товарів;
- можливість реєстрації та авторизації користувачів програмного забезпечення для збору інформації про товари;
- можливість відображення діапазону цін на товари на основі зібраної інформації;
- підтримка можливості подання користувачу детальної інформації про обраний товари;
- підтримка можливості інформування користувача щодо цін на обраний товар у різних магазинах;
- можливість замовлення обраного товару;
- можливість додавання та видалення товарів у каталог (функція адміністратора).

### **1.3 Висновки за розділом 1**

Визначено, що збір даних з вебсайтів полягає у видобуванні інформації з відкритих онлайн джерел та збереженні її в базах даних. Цей процес зазвичай здійснюється за допомогою скриптів або програм, які можуть обходити захисні механізми вебсайтів. Проаналізовано різні методи збору даних з вебсайтів. Проаналізовано застосування програмного забезпечення для збору та аналізу даних.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для збору інформації про товари. Проте використання у програмних засобах збору інформації про товари ненадійних джерел інформації, що надають неактуальні або неточні дані про товари, може призвести до хибних висновків або неправильних рішень. Крім

того, впровадження та налаштування під конкретні вимоги торгівельної організації деякого існуючого програмного забезпечення для збору даних про товари може вимагати значних фінансових інвестицій, особливо якщо йдеться про складні програмні системи з інтеграцією штучного інтелекту або машинного навчання. В деяких випадках інтеграція з різними програмними платформами та системами може бути складною та вимагати спеціальних технічних знань, також можливі проблеми сумісності з різними джерелами даних або іншими бізнес-системами. Тому актуальною є розробка програмного забезпечення для збору інформації про товари.

Сформульовано функціональні вимоги до програмного забезпечення для збору інформації про товари.

## 2 МАТЕРІАЛИ І МЕТОДИ

### 2.1 Вибір мови програмування

При розробці програмного забезпечення для збору інформації про товари було використано мову програмування C# [15], [16].

C# є мовою програмування, що відноситься до технології Microsoft .NET, та служить базою для створення програмного забезпечення та вебзастосунків. Однією з ключових ідей .NET є забезпечення сумісності програм, написаних на різних мовах програмування. Платформа .NET також легко інтегрується з хмарними сервісами, як-от Microsoft Azure, що дає можливість ефективно працювати з великими обсягами даних, аналізуючи їх у режимі реального часу. Це особливо важливо для компаній, що керують великим масивом даних про товари [15], [16].

Мова програмування C# добре поєднується з різними інтерфейсами прикладних програм (Application Programming Interface, API), дозволяючи отримувати інформацію з зовнішніх сервісів, таких як API торгових платформ або платформ товарних пропозицій. Така інтеграція особливо корисна для програмних засобів, яким потрібен доступ до оновлених даних з кількох джерел [15], [16].

Мова C# має повну підтримку від Microsoft і велику спільноту розробників, що забезпечує доступ до безлічі ресурсів, форумів і навчальних матеріалів. Це полегшує вирішення технічних питань і дає змогу використовувати найефективніші методи розробки [15], [16].

Розробка програм у .NET має ряд переваг. Вона дозволяє значно знизити витрати на створення інструментів для стимулювання працівників, а також сприяє повторному використанню коду, що прискорює процес розробки. Клієнтські програми, створені на .NET (так звані Smart Clients), можуть бути адаптовані до різних платформ із мінімальними витратами. Крім того, будь-яку програму, написану на .NET, можна легко перетворити на вебслужбу для роботи з XML [15], [16].

.NET Framework є об'єктно-орієнтованим, що робить його зручним для розробки складних систем. Проектна бібліотека в ньому має інтуїтивно зрозумілий інтерфейс, тому навіть користувачі з мінімальними знаннями програмування можуть створювати привабливі додатки. Платформа також забезпечує незалежність мов програмування, що дозволяє мовам, таким як C#, F# і C, взаємодіяти та компілюватися в єдину проміжну мову [15], [16].

.NET підтримує динамічні вебсторінки через технологію ASP.NET, що дозволяє створювати сучасні вебзастосунки. Спільне використання коду стало простішим завдяки введенню концепції агрегації, що замінила класичні бібліотеки. Важливою перевагою цієї платформи є також безпека, оскільки .NET забезпечує захист даних, визначаючи рівні доступу для різних груп користувачів [15], [16].

.NET спрощує процес розробки текстових програм, дозволяючи створювати їх у межах однієї програмної системи. Платформа підтримує C#, яка є об'єктно-орієнтованою мовою програмування, спеціально розробленою для роботи з .NET [15], [16].

Завдяки .NET Core програми, створені на C#, можуть функціонувати не тільки в Windows, а й на таких системах, як Linux і macOS. Це дозволяє використовувати програмне забезпечення для збору даних із різних вебресурсів. Мова C# також має великий набір бібліотек і фреймворків, які полегшують створення баз даних та інструментів для аналізу даних, таких як, робота з API, обробка JSON і XML, що робить її зручною для широкомасштабної обробки та аналітики [15], [16].

C# є основною мовою для .NET Framework, що забезпечує створення потужних вебзастосунків. Це особливо корисно для збору даних із онлайн магазинів, API та інших джерел, оскільки C# та .NET надають ефективні інструменти для інтеграції з вебтехнологіями, такими як HTTP-запити та аналіз даних [15], [16].

C# добре інтегрується з різними інтерфейсами прикладних програм, що дозволяє збирати дані з інтерфейсів сторонніх сервісів, як-от API торгових

платформ або інших програмних засобів. Це є досить важливим для програмного забезпечення, що потребує доступу до актуальних даних про товари з багатьох джерел [15], [16].

Обґрунтування вибору мови програмування для розробки програмного забезпечення для збору інформації про товари наведено у таблиці 2.1.

Таблиця 2.1 – Обґрунтування вибору мови програмування для розробки програмного забезпечення для збору інформації про товари

| Критерій порівняння мов програмування                      | Мова програмування |        |      |
|------------------------------------------------------------|--------------------|--------|------|
|                                                            | C#                 | Python | Ruby |
| Засоби інтеграції з різними API для можливості збору даних | +                  | +-     | +-   |
| Бібліотеки для збору даних                                 | +                  | +      | +-   |
| Багатоплатформність                                        | +                  | +      | +-   |
| Зручність для створення ПЗ для збору інформації про товари | +                  | -      | -    |
| Активна підтримка спільноти                                | +                  | +      | +-   |

Отже, для реалізації програмного забезпечення для збору інформації про товари обрано мову програмування C#, яка має велику кількість готових бібліотек і фреймворків, що можуть допомогти при створенні програм для збору даних. Крім того, мова C# добре інтегрується з різними API, що дозволяє збирати дані з інтерфейсів сторонніх сервісів, зокрема, API торгових платформ або агрегаторів товарів. Іншою перевагою C# є кросплатформність через .NET Core, оскільки завдяки .NET Core, програми, написані на C#, можуть працювати не тільки на Windows, але і на інших операційних системах, таких як Linux і macOS. Це розширює можливості використання програмного забезпечення для збору інформації на різних платформах.

## **2.2 Вибір середовища розробки для створення програмного забезпечення для збору інформації про товари**

В якості середовища розробки для створення програмного забезпечення для збору інформації про товари було обрано середовище Microsoft Visual Studio [17], [18].

Середовище розробки Microsoft Visual Studio є комплексом програмних продуктів від Microsoft, що включає середовище розробки та різноманітні інструменти. Visual Studio має тісну інтеграцію з платформами .NET Framework і .NET Core, що значно полегшує розробку додатків на C# та інших мовах, які підтримуються цими платформами. Це дозволяє створювати високопродуктивні додатки, здатні збирати інформацію з різних джерел, таких як веб-сайти, API та бази даних [17], [18].

Програма безперешкодно інтегрується з Azure DevOps та іншими інструментами, що надає можливість автоматизувати процеси аналізу даних і розгортання додатків на серверах або в хмарі. Завдяки підтримці .NET Core, розроблені у Visual Studio додатки можуть працювати не тільки на Windows, але й на інших операційних системах, зокрема Linux і macOS. Це розширює можливості використання програмного забезпечення для збору даних з різних вебресурсів [17], [18].

Також Visual Studio підтримує розробку програмних засобів з графічними інтерфейсами для платформ, таких як Windows Forms, а також вебсайти, які можуть бути написані як у вбудованому, так і в керованому коді. Крім операційної системи Windows, програми можуть бути розгорнуті для платформ Windows Mobile, Windows Phone, Windows CE, .NET Compact Framework і Silverlight [17], [18].

Microsoft Visual Studio Community Edition — це безкоштовна версія програми, яка не включає функції для інтеграції з додатковими технологіями чи спільної роботи, однак містить усе необхідне для завершення проєкту. Visual Studio має вбудовану підтримку Git та інших систем контролю версій,

що дозволяє розробникам краще координувати роботу, керувати змінами в коді та підтримувати його якість [17], [18].

Ця IDE підтримує численні розширення та плагіни, які полегшують роботу з різними технологіями, такими як аналітика, бази даних, інтеграція даних і штучний інтелект, даючи можливість налаштувати середовище під конкретні потреби бізнесу. Visual Studio також надає зручні інструменти для роботи з вебресурсами, використання HTTP-запитів і інтеграції API, що сприяє ефективному збору даних з різних джерел. Наприклад, бібліотека HttpClient інтегрована безпосередньо в середовище, спрощуючи роботу з вебданими [17], [18].

ASP.NET — це платформа Microsoft для створення програм у вигляді вебзастосунків, яка є частиною .NET і сучасним розвитком старішої технології ASP. Хоча вона має схожість у назві з оригінальною ASP, ASP.NET суттєво відрізняється, оскільки побудована на основі Common Language Runtime. ASP.NET компілює код перед його виконанням, що забезпечує більшу продуктивність і оптимізацію під час роботи [17], [18].

Visual Studio пропонує спеціалізований редактор коду, який підтримує функції авто доповнення, діалогові вікна та налагодження, що дозволяє виявляти й виправляти помилки в режимі реального часу. Це особливо корисно для розробки складних проєктів, оскільки допомагає швидко виправляти помилки [17], [18].

Завдяки інтеграції з хмарними сервісами Azure, Visual Studio дозволяє створювати програми, які використовують хмарні ресурси для збору, обробки та зберігання великих обсягів даних. Це є особливо важливим для інформаційних систем, які потребують обробки та аналізу даних у реальному часі, використовуючи хмарні можливості [17], [18].

Обґрунтування вибору середовища розробки для створення програмного забезпечення для збору інформації про товари наведено у таблиці 2.2.

Таблиця 2.2 – Обґрунтування вибору середовища розробки для створення програмного забезпечення для збору інформації про товари

| Критерій порівняння середовищ розробки | Середовища розробки     |         |               |
|----------------------------------------|-------------------------|---------|---------------|
|                                        | Microsoft Visual Studio | Eclipse | IntelliJ IDEA |
| Підтримка багатоплатформової розробки  | +                       | +       | +—            |
| Потужний редактор коду                 | +                       | +—      | —             |
| Інструменти для налагодження           | +                       | +—      | +             |
| Можливість роботи з хмарними сервісами | +                       | +—      | +—            |
| Засоби інтеграції                      | +                       | —       | +—            |

Таким чином, для створення програмного забезпечення для збору інформації про товари обрано середовище розробки Microsoft Visual Studio, що дозволяє розробляти програми, які використовують ресурси для збору, обробки та зберігання даних, що є важливим для програмних систем збору інформації про товари, які можуть потребувати хмарних потужностей для обробки та аналізу даних у реальному часі. Крім того, середовище розробки Microsoft Visual Studio підтримує багатоплатформову розробку, оскільки разом з .NET Core дозволяє створювати програми, які можуть працювати не лише на Windows, але й на інших операційних системах, що розширює можливості використання програмного забезпечення для збору інформації на різних платформах. Visual Studio пропонує інструменти та бібліотеки, що спрощують розробку програм для збору інформації про товари шляхом взаємодії з API, що є важливим для збору даних про товари з різних джерел.

### **2.3 Розробка моделі взаємодії користувача з програмним забезпеченням для збору інформації про товари**

З метою можливості розробки програмного забезпечення для збору інформації про товари створимо відповідну модель у вигляді діаграми прецедентів.

При розробленні моделі взаємодії користувача з програмним забезпеченням для збору інформації про товари враховуємо те, що користувачі програми мають можливості ознайомлення з каталогом товарів, реєстрації та авторизації, ознайомлення з діапазоном цін на товари та детальною інформацією про обраний товари, ознайомлення з цінами на обраний товар у різних магазинах, замовлення обраного товару.

Основний успішний сценарій користувача при роботі з програмним забезпеченням для збору інформації про товари можна описати таким чином:

- користувач програмного забезпечення для збору інформації про товари авторизується;
- після авторизації користувача сервер програмного забезпечення для збору інформації про товари у відповідь повертає основну сторінку вебсайту програми для збору інформації про товари;
- користувач бачить відповідну сторінку з інформацією про товари та шукає потрібний товар;
- користувач обирає товар та ознайомлюється з детальною інформацією про нього, зокрема, з діапазоном цін на товар у різних магазинах;
- користувач натискає відповідну кнопку для купівлі потрібного йому товару у відповідному магазині;
- користувач завершує роботу з програмним забезпеченням для збору інформації про товари.

Враховуючи сценарії роботи користувачів при роботі з програмою, модель взаємодії користувача з програмним забезпеченням для збору

інформації про товари подамо у вигляді діаграми прецедентів, зображеної на рис. 2.1.

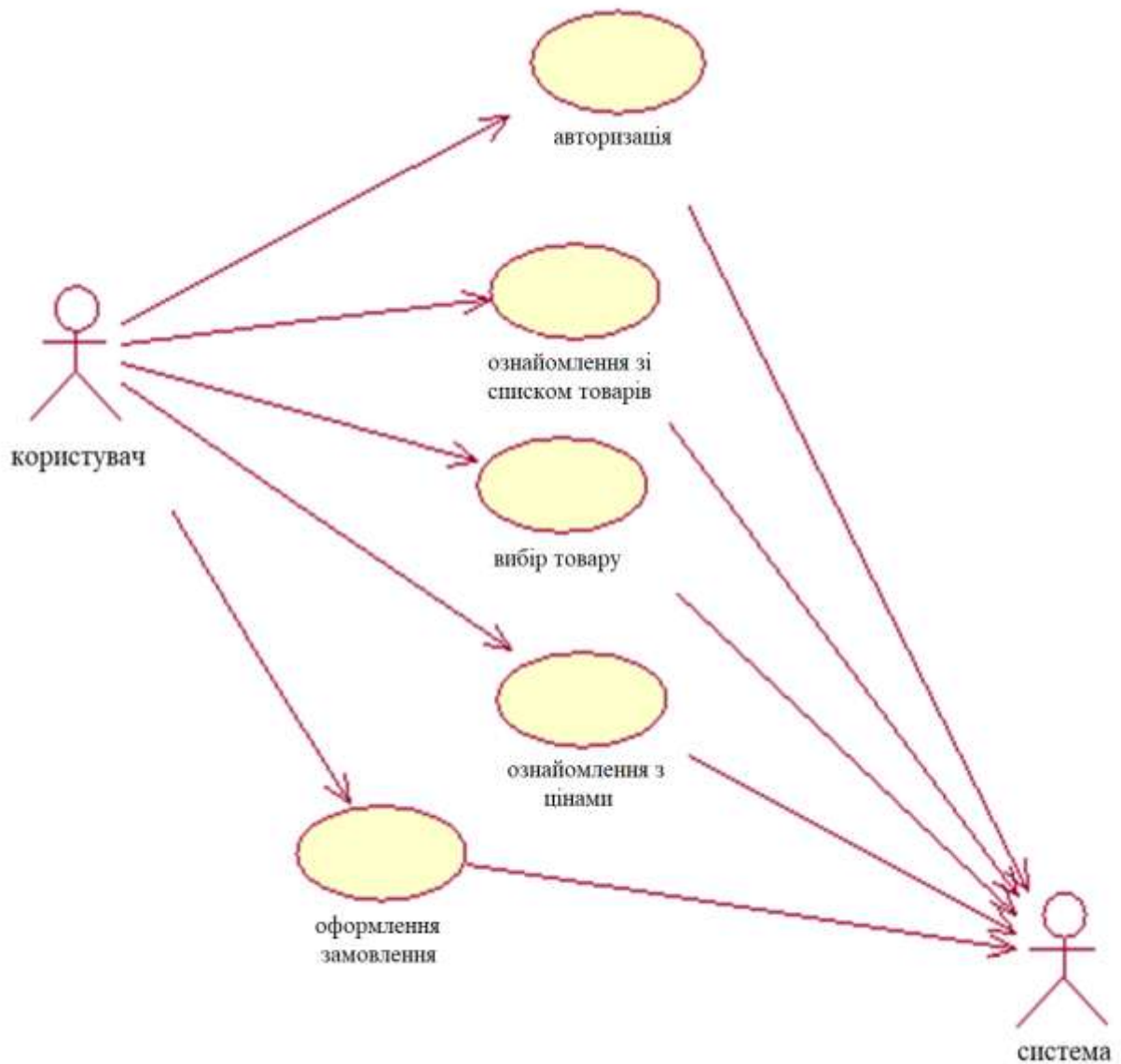


Рисунок 2.1 – Модель взаємодії користувача з програмним забезпеченням для збору інформації про товари

Таким чином, створено модель взаємодії користувача з програмним забезпеченням для збору інформації про товари, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для збору інформації про товари.

## 2.4 Висновки за розділом 2

Для реалізації програмного забезпечення для збору інформації про товари обрано мову програмування C#, яка має велику кількість готових бібліотек і фреймворків, що можуть допомогти при створенні програм для збору даних. Крім того, мова C# добре інтегрується з різними API, що дозволяє збирати дані з інтерфейсів сторонніх сервісів, зокрема, API торгових платформ або агрегаторів товарів. Іншою перевагою C# є кросплатформність через .NET Core, оскільки завдяки .NET Core, програми, написані на C#, можуть працювати не тільки на Windows, але і на інших операційних системах.

Для створення програмного забезпечення для збору інформації про товари обрано середовище розробки Microsoft Visual Studio, що дозволяє розробляти програми, які використовують ресурси для збору, обробки та зберігання даних, що є важливим для програмних систем збору інформації про товари, які можуть потребувати хмарних потужностей для обробки та аналізу даних у реальному часі. Крім того, середовище розробки Microsoft Visual Studio підтримує багатоплатформову розробку, оскільки разом з .NET Core дозволяє створювати програми, які можуть працювати не лише на Windows, але й на інших операційних системах, що розширює можливості використання програмного забезпечення для збору інформації на різних платформах.

Створено модель взаємодії користувача з програмним забезпеченням для збору інформації про товари, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для збору інформації про товари.

### 3 ОПИС ПРОГРАМИ

#### 3.1 Структура програмного забезпечення для збору інформації про товари

Структуру програмного забезпечення для збору інформації про товари наведено на рис. 3.1.

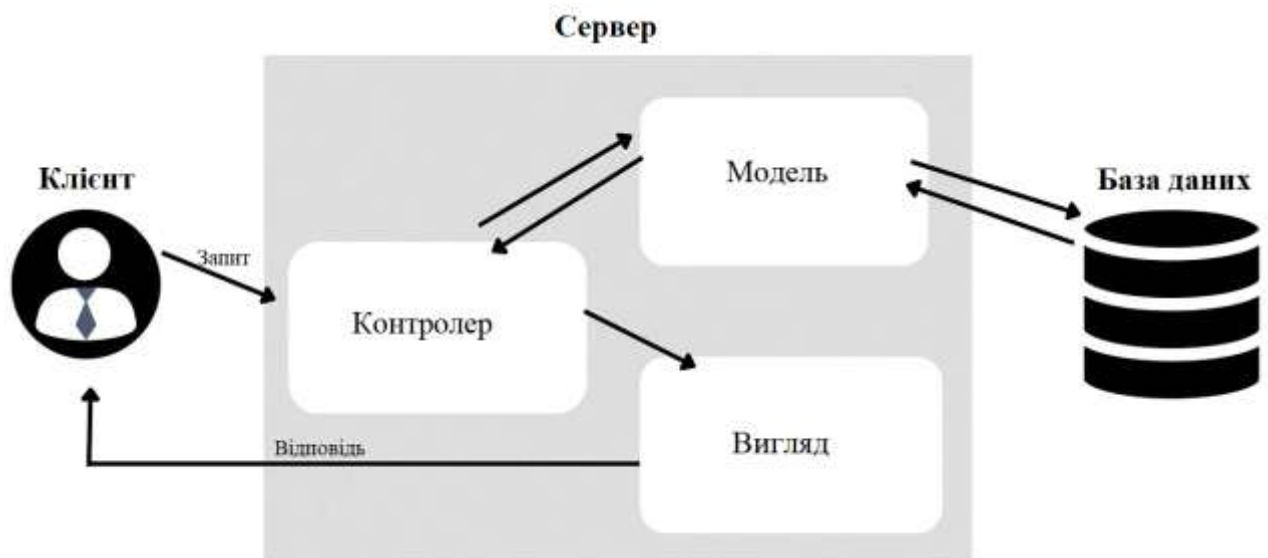


Рисунок 3.1 – Структура програмного забезпечення для збору інформації про товари

Як видно з рис. 3.1, як базис для створення програмного забезпечення для збору інформації про товари обрано архітектуру модель-вигляд-контролер (Model-View-Controller, MVC), що реалізує клієнт-серверний підхід [19].

Архітектура модель-вигляд-контролер пропонує структуру програмування, яку можна застосовувати в різних фреймворках і на багатьох мовах програмування.

Коли клієнт відправляє запит, сервер програмного забезпечення для збору інформації про товари отримує його та визначає, чи варто його обробляти. Після обробки запиту сервер надсилає дані про товари з бази даних, обробляє їх і повертає відповідь клієнту. Цей процес взаємодії називається циклом запит-відповідь.

Модель (Model) є частиною архітектури Model-View-Controller і представляє дані програмного забезпечення для збору інформації про товари. Модель у програмному забезпеченні для збору інформації про товари є екземпляром певного класу, що відповідає за управління даними, їх створення, зміну та збереження відповідно до встановлених правил. Вона також обробляє отримані дані та передає їх далі.

Важливо зауважити, що модель програмного забезпечення для збору інформації про товари зосереджена виключно на управлінні даними. Вона не відповідає за їх відображення на екрані чи збереження у базі даних. Модель програмного забезпечення для збору інформації про товари лише слідкує за змінами в даних і передає цю інформацію далі. Це автономний елемент, який знає лише про себе і про свою функціональність.

Вид (View) є об'єктом програмного забезпечення для збору інформації про товари, що займається візуалізацією моделі на пристрої виведення. Він не змінює самі дані, лише зчитує їх із моделі програмного забезпечення для збору інформації про товари та виводить на екран. Вид також відслідковує зміни в моделі програмного забезпечення для збору інформації про товари та автоматично оновлюється, коли модель змінюється.

Контролер програмного забезпечення для збору інформації про товари виконує функції завантаження та збереження даних з/до сховищ і зв'язує модель із поданням. У програмному забезпеченні для збору інформації про товари контролер також може обробляти дії користувача, такі як клацання мишкою чи натискання клавіш, і вносити зміни до моделі програмного забезпечення для збору інформації про товари відповідно до цих дій. У деяких версіях архітектури ці функції часто переходять до подання, яке може генерувати події та ініціювати зміни в моделях. У певній версії подання знає про модель, але не має інформації про контролер, який лише забезпечує їх взаємодію.

### 3.2 Функціонування програмного забезпечення для збору інформації про товари

Функціонування програмного забезпечення для збору інформації про товари подамо за допомогою схеми, зображеної на рис. 3.2.

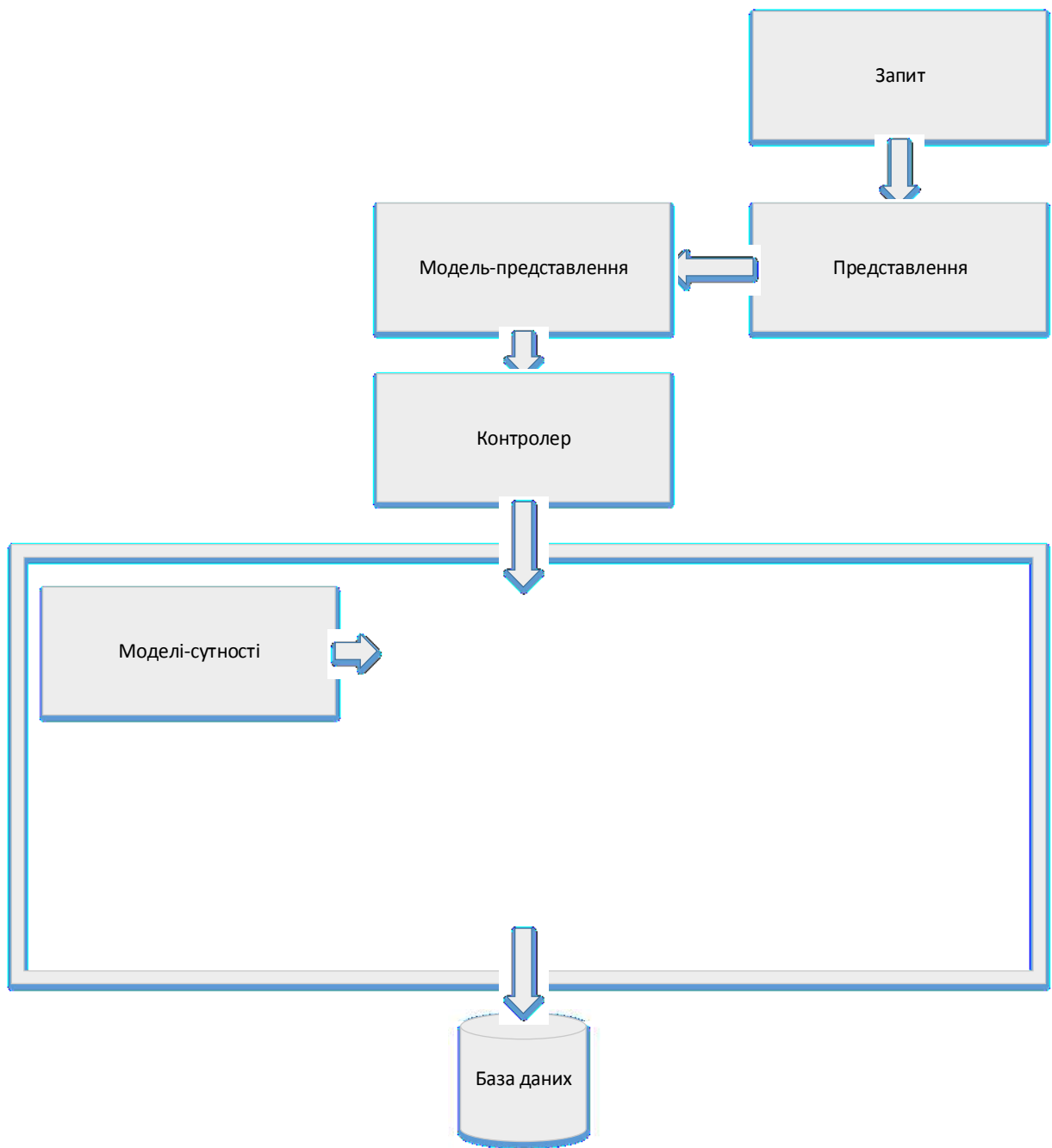


Рисунок 3.2 – Схема функціонування програмного забезпечення для збору інформації про товари

На початку функціонування програмного забезпечення для збору інформації про товари клієнт заходить на вебсайт сервісу. Покупець шукає товари різними методами, наприклад, за ключовими словами, категоріями або атрибутами товару. Потім він збирає детальну інформацію про продукт, переглядає список Інтернет магазинів, де цей товар доступний, і отримує інформацію про ціну та наявність. Далі покупець додає обрані товари до кошика і завершує процес, залишаючи необхідну інтерактивну інформацію. Після цього менеджери отримують електронне повідомлення про замовлення та, за необхідності, зв'язуються з клієнтом для уточнення наступних кроків.

Особливістю програми є можливість доступу до автентичного контенту щодо цін на різні товари, зібраного з Інтернету. На цьому етапі формуються (автоматично, без участі користувача сайту) запити щодо продуктів на вебсайтах Інтернет магазинів, потім обробляються результати, а ціни на товари реєструються та зберігаються у базі даних програмного забезпечення для збору інформації про товари.

Для роботи з базою даних програмного забезпечення для збору інформації про товари створюються різні поля. Також впроваджуються спеціальні посилання та механізми для управління залежностями. Збірка використовує об'єкти, що містять необхідну інформацію для відображення користувачеві, перевірки даних тощо. Спеціальні функції спрощують процес відображення моделей, перетворюючи їх у форми, зручні для візуалізації.

Програмне забезпечення для збору інформації про товари базується на трьох модулях: Dmn, WbUI та ArtNN. Dmn відповідає за роботу з базою даних, обробку запитів, додавання та редагування записів. Вона також містить бізнес-логіку, яка відповідає за правила ведення бізнесу для конкретних компаній. WbUI займається обробкою запитів користувачів, аналізом даних та створенням користувацьких профілів. Модуль ArtNN використовує нейронні мережі для виконання аналітичних завдань.

Модуль WbUI програмного забезпечення для збору інформації про товари має складну структуру, яка забезпечує виконання всіх необхідних

функцій, зокрема обробку вебзапитів і аналіз подій. Вона також включає основну архітектуру програмного забезпечення для збору інформації про товари разом із низкою додаткових інструментів. Хоча програмне забезпечення для збору інформації про товари містить багато інформації, воно базується на добре організованих виробничих рішеннях, що робить його зрозумілим. Файл `PrcsHlpr.cs`, розміщений у допоміжному каталозі, є частиною програми і відповідає за встановлення стандартних налаштувань обміну повідомленнями з зовнішніми Інтернет магазинами. Його головна функція полягає в тому, щоб створити копію відповідного класу, який використовується як стандартний клієнт для HTTP-запитів. Для отримання відповіді від сторонніх серверів застосовуються методи відповідно для GET- та POST-запитів. Ці методи приймають параметри, необхідні для взаємодії з сервером. Отримані результати обробляються стандартними потоковими інструментами C#.

Модуль `ArtNN` використовується для виконання завдань за допомогою нейронних мереж, щоб прогнозувати ціни на товари. Цей модуль є інструментом, який відповідає за отримання моделі для відповідного класу, обчислення необхідних параметрів та виконання прогнозів.

### **3.3 Розробка бази даних програмного забезпечення для збору інформації про товари**

База даних програмного забезпечення для збору інформації про товари призначена для зберігання даних про товари, магазини, де можна придбати відповідні товари, а також для зберігання інформації, необхідної для замовлення товарів.

Схему бази даних програмного забезпечення для збору інформації про товари наведено на рис. 3.3.

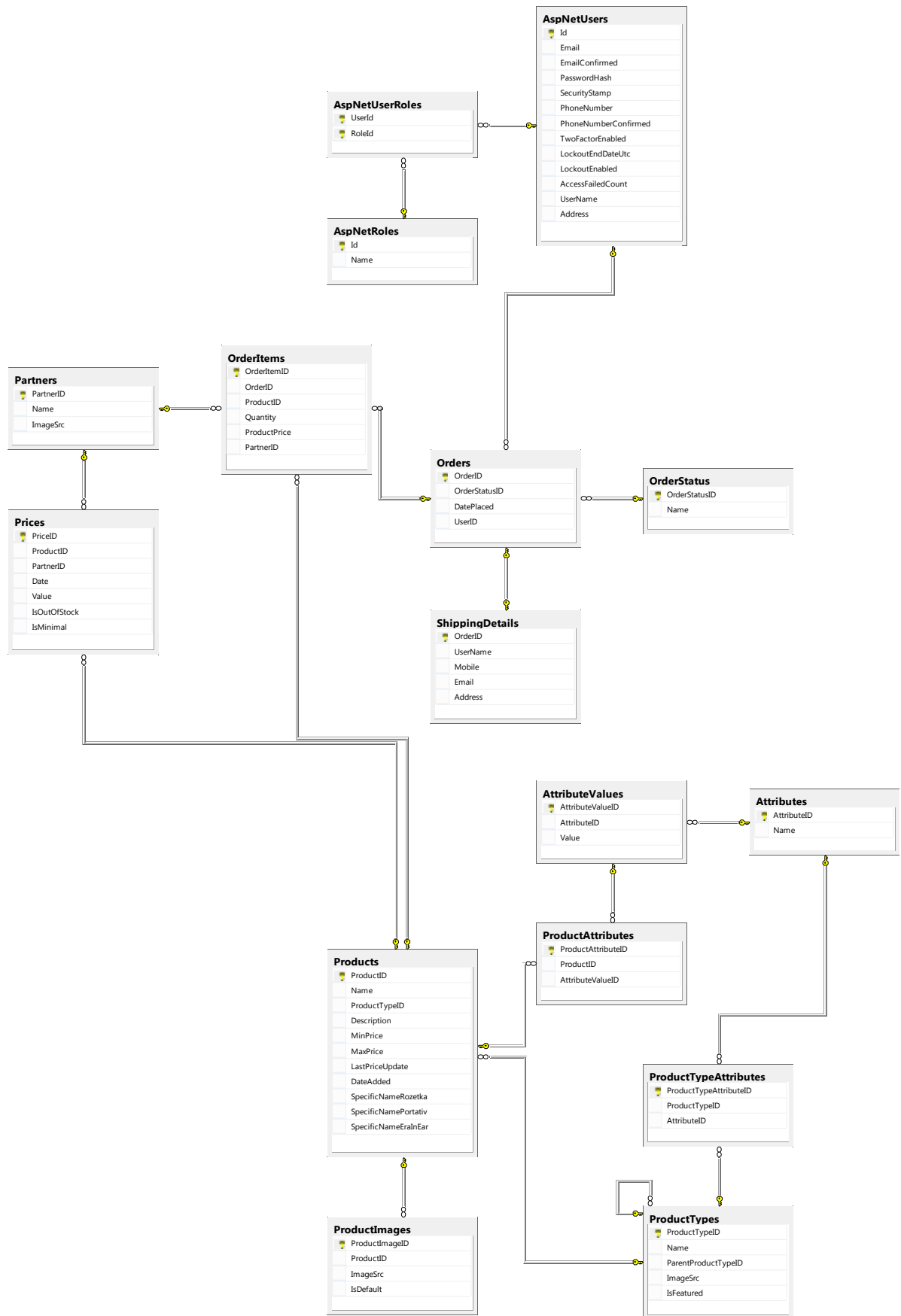


Рисунок 3.3 – Схема бази даних програмного забезпечення для збору інформації про товари

Схема бази даних програмного забезпечення для збору інформації про товари відображає концептуальні зв'язки між обраними елементами у відповідній предметній області. Основними сутностями бази даних обрано такі: товари (Products), магазини (Partners), ціни на товари (Prices), замовлення (Orders), опис товару (Attributes) та інші.

### **3.4 Проєктування інтерфейсу програмного забезпечення для збору інформації про товари**

Проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для збору інформації про товари виконано з урахуванням вимог технічного завдання. При розробленні програмного забезпечення для збору інформації про товари враховані функціональні вимоги до програми:

- відображення каталогу товарів;
- можливість реєстрації та авторизації користувачів програмного забезпечення для збору інформації про товари;
- можливість відображення діапазону цін на товари на основі зібраної інформації;
- підтримка можливості подання користувачу детальної інформації про обраний товари;
- підтримка можливості інформування користувача щодо цін на обраний товар у різних магазинах;
- можливість замовлення обраного товару;
- можливість додавання та видалення товарів у каталог (функція адміністратора).

Інтерфейс головної форми програмного забезпечення для збору інформації про товари наведено на рис. 3.4.

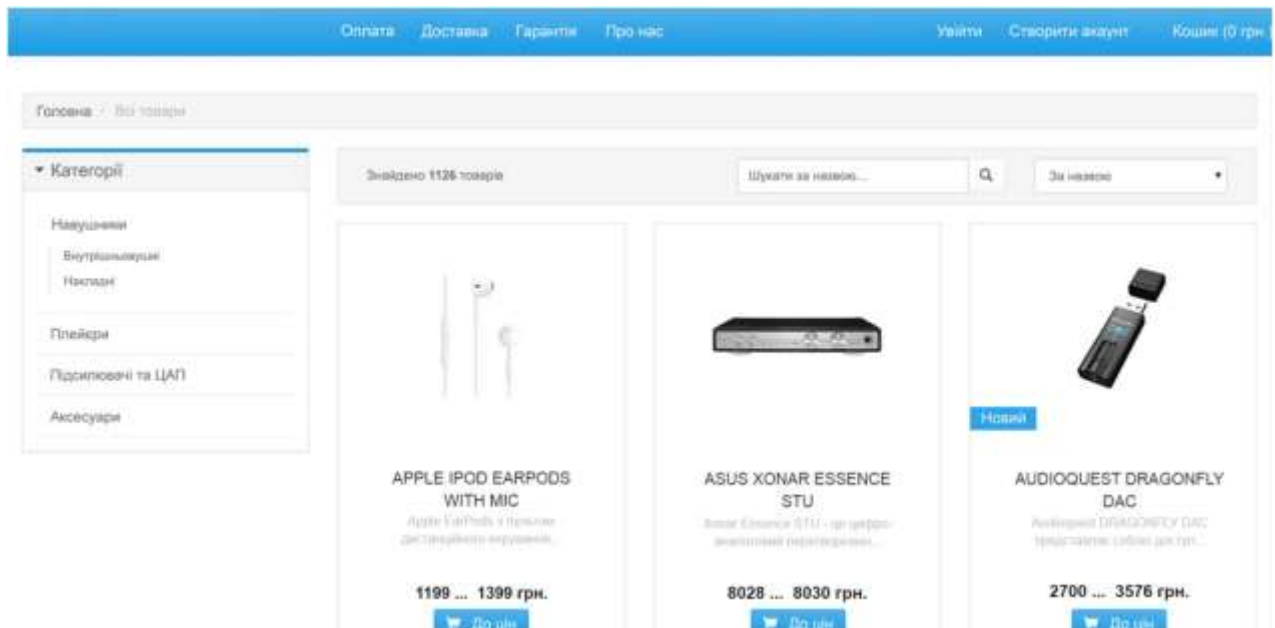


Рисунок 3.4 – Інтерфейс головної сторінки програмного забезпечення для збору інформації про товари

Інтерфейс сторінки авторизації користувача програми для збору інформації про товари наведено на рис. 3.5.

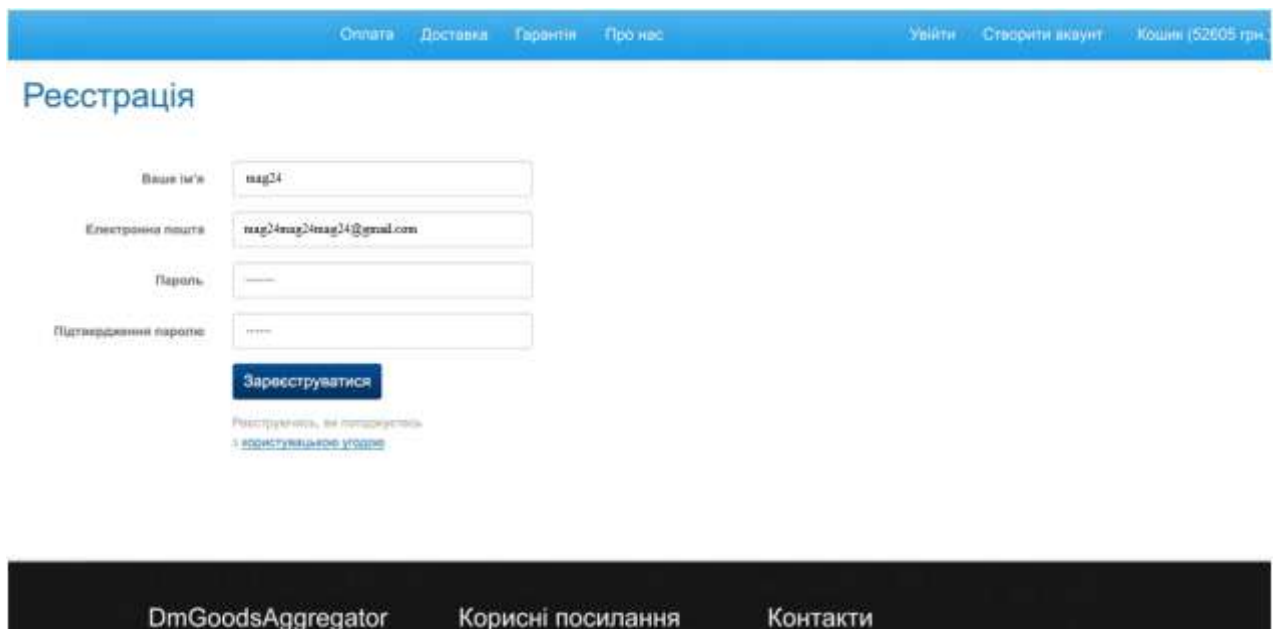


Рисунок 3.5 – Інтерфейс сторінки авторизації користувача програми для збору інформації про товари

Інтерфейс сторінки з детальною інформацією та характеристиками товару наведено на рис. 3.6.

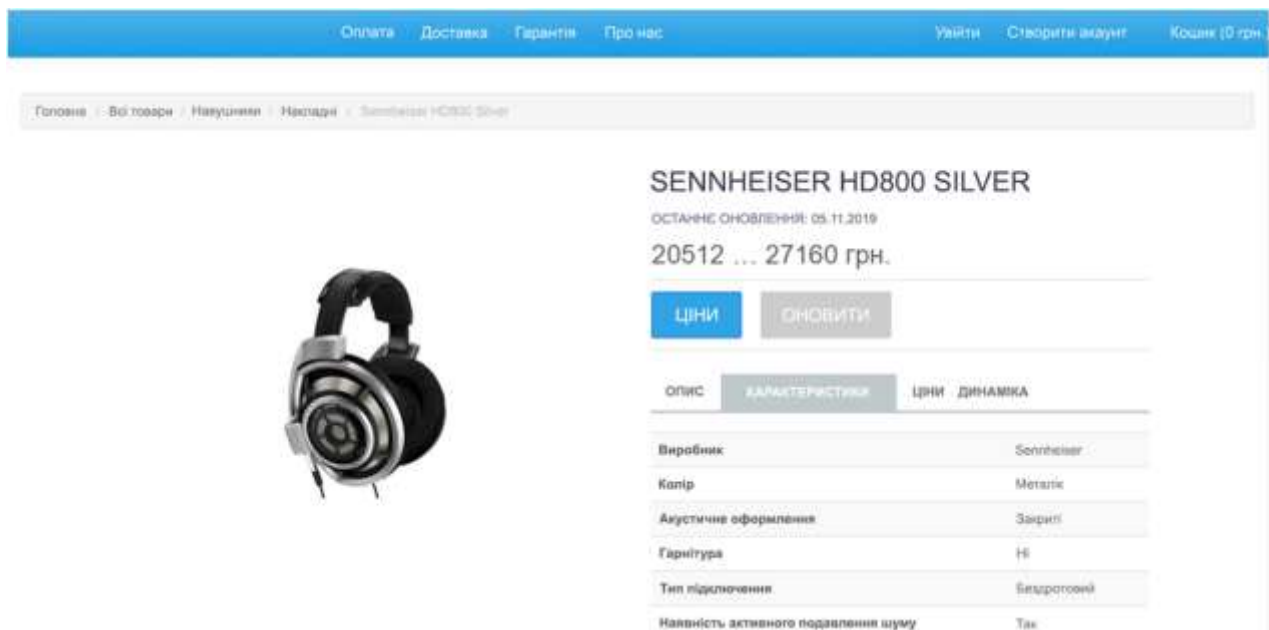


Рисунок 3.6 – Ознайомлення з детальною інформацією та характеристиками товару

Інтерфейс сторінки з аналізом цін на товар у різних магазинах наведено на рис. 3.7.

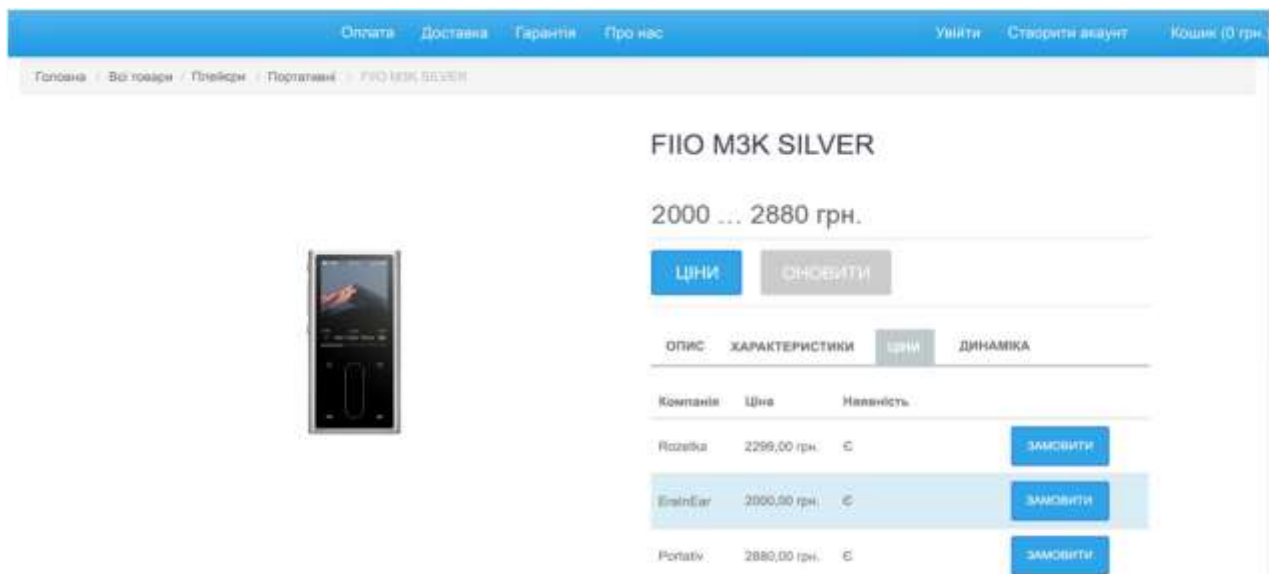


Рисунок 3.7 – Аналіз цін на товар у різних магазинах

Інтерфейс сторінки кошику замовлення товарів наведено на рис. 3.8.

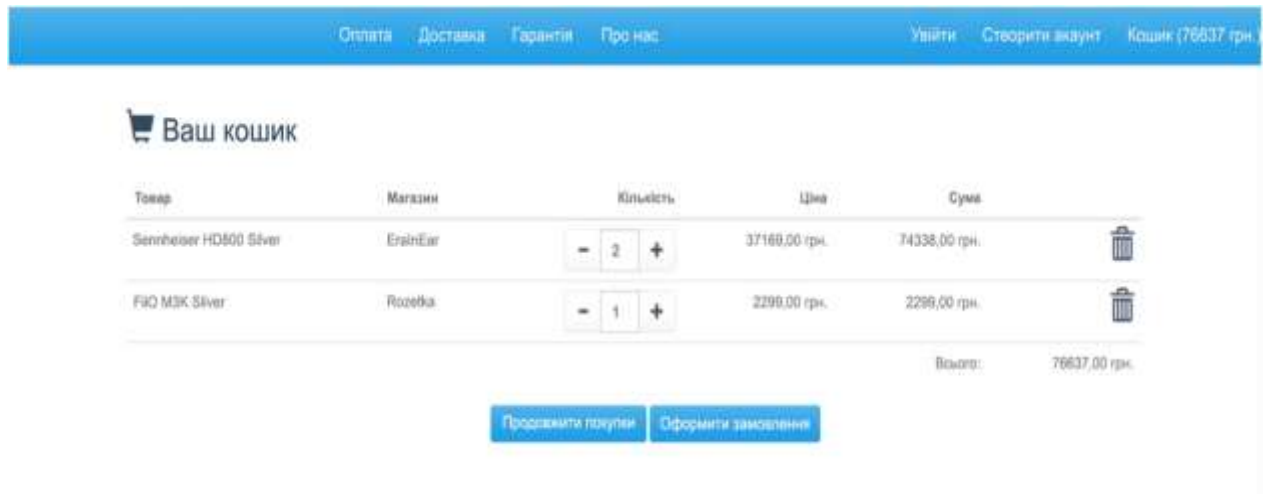


Рисунок 3.8 – Робота з кошиком замовлення товарів

Інтерфейс сторінки оформлення замовлення товарів наведено на рис. 3.9.

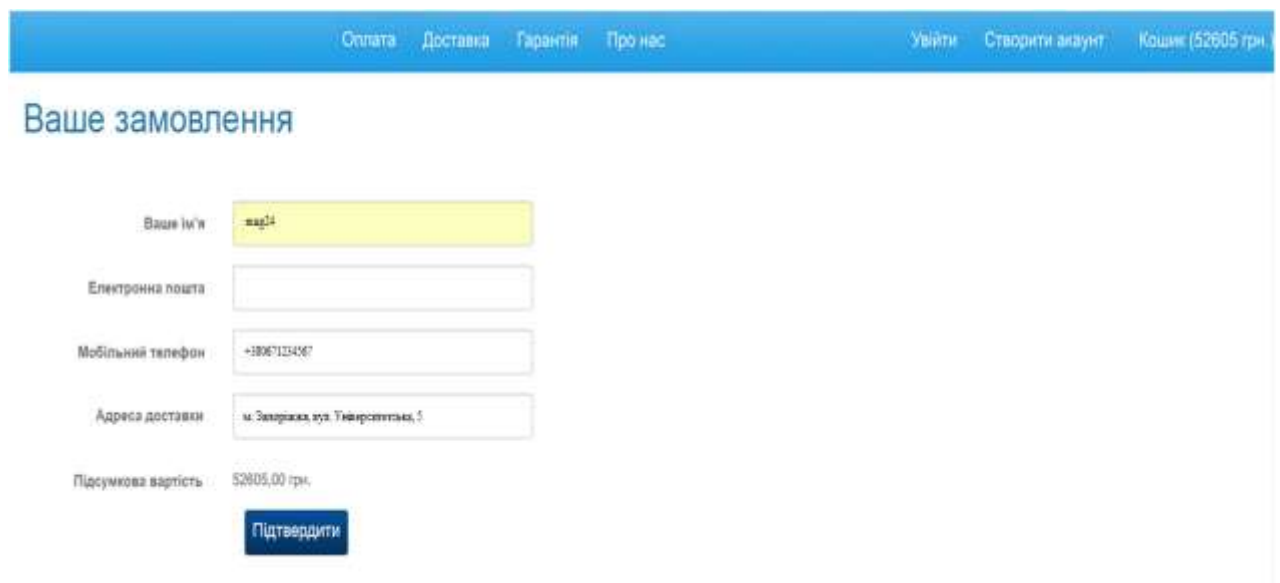


Рисунок 3.9 – Оформлення замовлення товарів

Сторінку з таблицею товарів для редагування адміністратором наведено на рис. 3.10.

| <div> <div>Оплата</div> <div>Доставка</div> <div>Гарантия</div> <div>Про нас</div> <div>Администрирование</div> </div> <div>Привет, Админ</div> <div>Выход</div> <div>Корзина (0 грн.)</div> |                             |                                                                                            |               |                 |           |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|--------------------------------------------------------------------------------------------|---------------|-----------------|-----------|
| <div> <div>▼ Список сущностей</div> <div>Товары</div> <div>Категории</div> </div>                                                                                                            |                             | <div> <div>Искать по названию...</div> <div>Q</div> <div>Добавить новый товар</div> </div> |               |                 |           |
| ID                                                                                                                                                                                           | Название                    | Категория                                                                                  | Цена          | Дата обновления |           |
| 1025                                                                                                                                                                                         | Apple iPod EarPods with Mic | Внутренняя                                                                                 | 1199.00 грн.  | 06.10.2024      | Взглянуть |
| 1026                                                                                                                                                                                         | Koss The Plug Black         | Внутренняя                                                                                 | 519.00 грн.   | 06.10.2024      | Взглянуть |
| 1027                                                                                                                                                                                         | Koss The Plug Blue          | Внутренняя                                                                                 | 539.00 грн.   | 06.10.2024      | Взглянуть |
| 1028                                                                                                                                                                                         | Koss The Plug Red           | Внутренняя                                                                                 | 529.00 грн.   | 06.10.2024      | Взглянуть |
| 1029                                                                                                                                                                                         | Sennheiser HD 4.50 BTNC     | Накладная                                                                                  | 8019.00 грн.  | 07.10.2024      | Взглянуть |
| 1030                                                                                                                                                                                         | Sennheiser HDX 125          | Накладная                                                                                  | 4179.00 грн.  | 06.10.2024      | Взглянуть |
| 1031                                                                                                                                                                                         | Sony WH-1000XM3 Black       | Накладная                                                                                  | 8999.00 грн.  | 06.10.2024      | Взглянуть |
| 1032                                                                                                                                                                                         | Sony WH-CH510 Black         | Накладная                                                                                  | 1399.00 грн.  | 06.10.2024      | Взглянуть |
| 1033                                                                                                                                                                                         | Sennheiser HD600 Silver     | Накладная                                                                                  | 37160.00 грн. | 07.10.2024      | Взглянуть |

Рисунок 3.10 – Таблица товаров

### 3.5 Висновки за розділом 3

Розроблено програмне забезпечення для збору інформації про товари.

Запропоновано структуру програмного забезпечення для збору інформації про товари. Визначено, що як базис для створення програмного забезпечення для збору інформації про товари обрано архітектуру модель-вигляд-контролер (Model-View-Controller), що реалізує клієнт-серверний підхід.

Розроблено базу даних програмного забезпечення для збору інформації про товари, яка відображає концептуальні зв'язки між обраними елементами у відповідній предметній області. Визначено, що основними сутностями бази даних є такі: товари (Products), магазини (Partners), ціни на товари (Prices), замовлення (Orders), опис товару (Attributes) та інші.

Описано функціонування програмного забезпечення для збору інформації про товари.

Описано особливості реалізації програмного забезпечення для збору інформації про товари. Виконано проектування інтерфейсу взаємодії користувача з програмним забезпеченням для збору інформації про товари.

## **4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ**

### **4.1 Призначення й умови застосування програми**

Програма призначена для підтримки процесу збору інформації про товари. Програма написана на мові C#, яка має велику кількість готових бібліотек і фреймворків, що можуть допомогти при створенні програм для збору даних. В якості середовища розробки для створення програмного забезпечення для збору інформації про товари обрано Microsoft Visual Studio, що дозволяє розробляти програми, які використовують ресурси для збору, обробки та зберігання даних, що є важливим для програмних систем збору інформації про товари, які можуть потребувати хмарних потужностей для обробки та аналізу даних у реальному часі.

Програмне забезпечення для збору інформації про товари забезпечує виконання таких функцій:

- відображення каталогу товарів;
- можливість реєстрації та авторизації користувачів програмного забезпечення для збору інформації про товари;
- можливість відображення діапазону цін на товари на основі зібраної інформації;
- підтримка можливості подання користувачу детальної інформації про обраний товари;
- підтримка можливості інформування користувача щодо цін на обраний товар у різних магазинах;
- можливість замовлення обраного товару;
- можливість додавання та видалення товарів у каталог (функція адміністратора).

Для експлуатації програмного забезпечення для збору інформації про товари необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять об'ємом 4 ГБ і більше для забезпечення стабільної роботи при обробці даних;
- система зберігання даних для забезпечення швидкого читання і запису даних;
- процесор із тактовою частотою не менше 2 ГГц;
- наявність вебсерверу для обробки великого обсягу даних, запитів і обчислювальних операцій;
- встановлена операційна система та веббраузер на стороні клієнта.

Функціональні характеристики розробленого програмного забезпечення для збору інформації про товари наведено у технічному завданні (додаток А). Фрагмент тексту програмного забезпечення для збору інформації про товари наведено у додатку Б.

## **4.2 Характеристики програми для збору інформації про товари**

Програмне забезпечення для збору інформації про товари складається з серверної та клієнтської частин.

Програмне забезпечення для збору інформації про товари забезпечує перегляд списку товарів та відповідних цін з різних Інтернет магазинів, фільтрацію товарів за різними характеристиками, пошук товару, реєстрацію та авторизацію користувачів з можливістю редагування особистих даних. Воно також надає функцію авторизації, дозволяє адміністраторам керувати інформацією про товари на сайті, а також оновлювати інформацію про товари в реальному часі. Користувачі можуть замовляти товари через сайт з різних магазинів, отримувати розсилки, а дані зберігаються у базі даних. Програма також інформує про помилки та ключові події системи.

Програма дозволяє виконувати автоматичний збір інформації з різних вебсайтів Інтернет магазинів, що дозволяє користувачам отримувати актуальні дані про товари, їхні характеристики, ціни та наявність у різних магазинах. Інформація щодо цін на товари збирається серверною частиною програми, не

залежно від дій користувача. Програма дозволяє виконувати фільтрацію товарів за різними параметрами, що допомагає покупцям швидко знаходити необхідні товари та приймати рішення щодо їхньої покупки на основі актуальних даних.

### **4.3 Інструкція по експлуатації програми**

#### **4.3.1 Звернення до програми**

Для запуску програмного забезпечення для збору інформації про товари на серверній частині необхідно запустити відповідний ру-файл. Для звертання до клієнтської сторони програмного забезпечення для збору інформації про товари необхідно перейти на відповідну вебсторінку.

#### **4.3.2 Вхідні й вихідні дані**

Вхідними даними до програмного забезпечення для збору інформації про товари є інформація про товар (назва або категорія), який потрібен користувачу, інформація для замовлення товару, реєстраційні дані користувача та ін.

Вихідними даними програмного забезпечення для збору інформації про товари є створені у базі даних записи про товари, зокрема, ціни на товари у різних Інтернет магазинах та ін.

#### **4.3.3 Повідомлення**

Користувач програмного забезпечення для збору інформації про товари може отримати такі повідомлення: повідомлення про успішну реєстрацію, повідомлення про успішне замовлення, повідомлення про некоректно введені особисті дані та ін.

#### 4.4 Виконання програмного забезпечення для збору інформації про товари

Після запуску програмного забезпечення для збору інформації про товари користувачу відображається головна сторінка (рис. 4.1). Зверху головної сторінки знаходиться меню, завдяки якому є можливість переходу до сторінок з інформацією щодо оплати, доставки, гарантії, інформацією про магазин і т.ін. Також користувач може ознайомитися з переліком товарів, їх цінами. Крім того є можливість фільтрації товарів за категоріями та пошуку товарів, наприклад, за назвою. Відзначимо, що інформація щодо цін на товари збирається серверною частиною програми, не залежно від дій користувача.

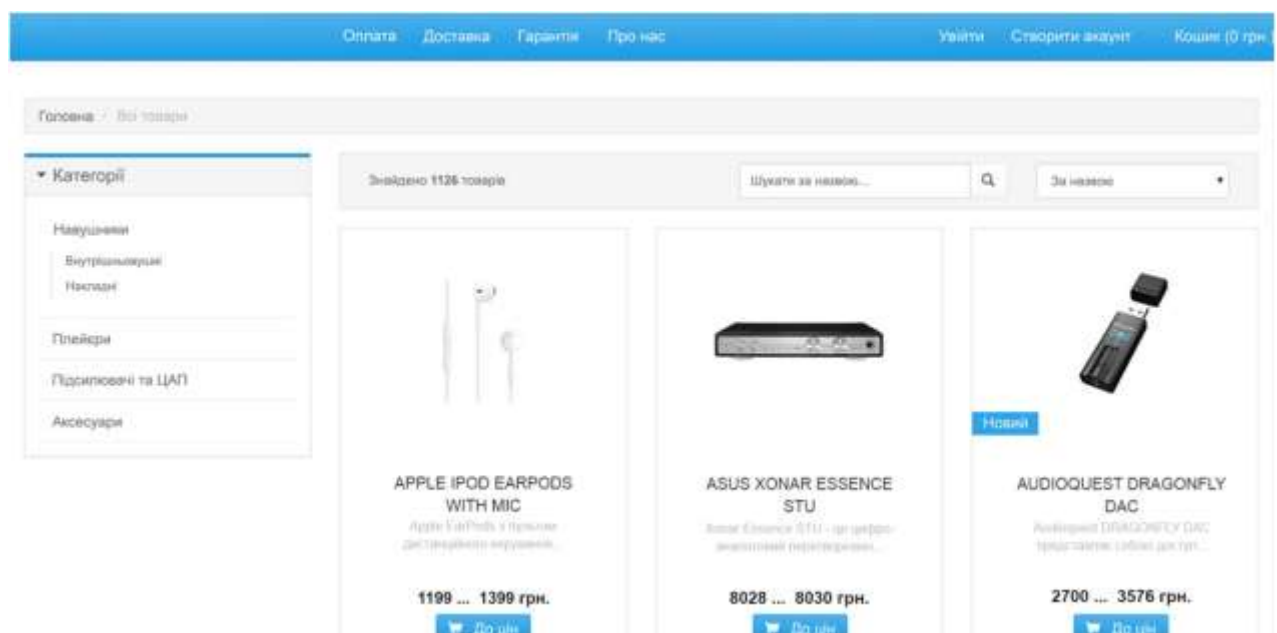


Рисунок 4.1 – Головна сторінка програмного забезпечення для збору інформації про товари

Для отримання можливості оформлення замовлення товарів користувачу необхідно зареєструватися (рис. 4.2) або авторизуватися. Необхідні дії можна виконати, перейшовши за посиланням у головному меню.

Оплата

Доставка

Гарантія

Про нас

Увійти

Створити акаунт

Кошик (52605 грн.)

## Реєстрація

Ваше ім'я

mag24

Електронна пошта

mag24mag24mag24@gmail.com

Пароль

.....

Підтвердження паролю

.....

Зареєструватися

Реєструючись, ви погоджуєтесь  
 з [користувачами угодами](#)

Рисунок 4.2 – Авторизація користувача програми для збору інформації про товари

Обравши потрібний товар та натиснувши на ньому, користувач переходить до сторінки з детальною інформацією та характеристиками товару (рис. 4.3).

Оплата

Доставка

Гарантія

Про нас

Увійти

Створити акаунт

Кошик (0 грн.)

Головна

Всі товари

Навушники

Накладки

Сennheiser HD800 Silver

### SENNHEISER HD800 SILVER

ОСТАННЄ ОНОВЛЕННЯ: 05.11.2019

20512 ... 27160 грн.

ЦІНИ

ОНОВИТИ

ОПИС

ХАРАКТЕРИСТИКИ

ЦІНИ

ДИНАМІКА

|                                     |             |
|-------------------------------------|-------------|
| Виробник                            | Sennheiser  |
| Колір                               | Metazik     |
| Акустичне оформлення                | Закриті     |
| Гарнітура                           | Ні          |
| Тип підключення                     | Бездротовий |
| Наявність активного подавлення шуму | Так         |

Рисунок 4.3 – Ознайомлення з детальною інформацією та характеристиками товару

Натиснувши на вкладку Ціни, відображаються пропозиції відповідного товару, зокрема, його наявність та ціни, у різних Інтернет магазинах (рис. 4.4).

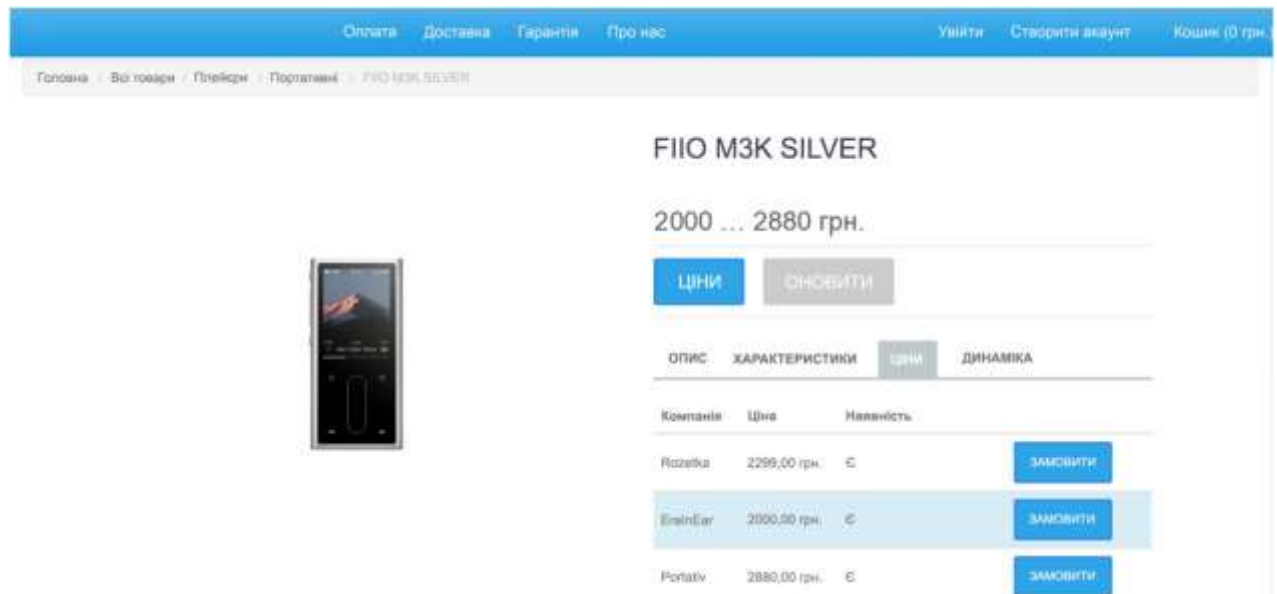


Рисунок 4.4 – Аналіз цін на товар у різних магазинах

Натискаючи кнопку Замовити, користувач додає товар до кошику (рис. 4.5). Після вибору всіх необхідних товарів користувач може перейти до кошику, натиснувши відповідне посилання у правому верхньому куті сторінки. У кошику відображається перелік обраних товарів, їх кількість та вартість. Користувач може змінити кількість товарів та видалити непотрібні товари з кошику.

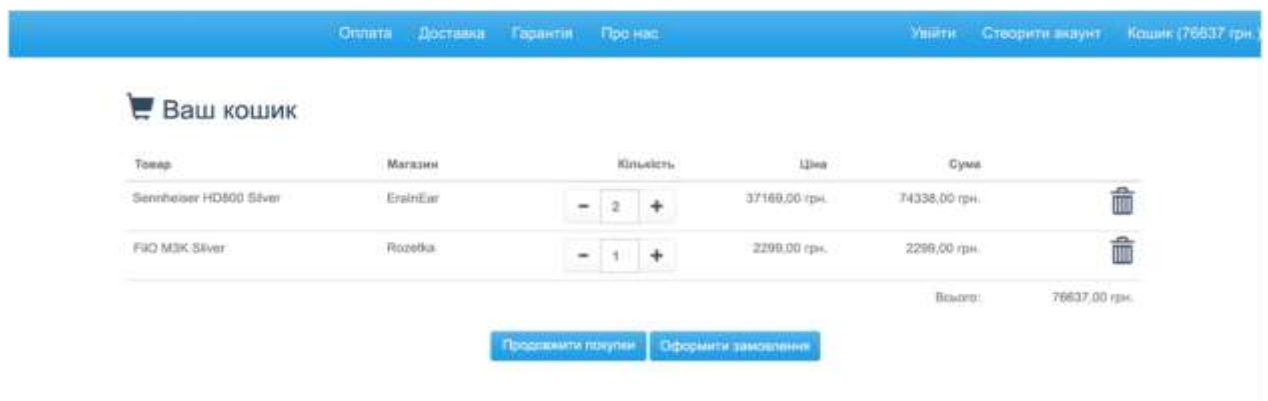


Рисунок 4.5 – Робота з кошиком замовлення товарів

Після натискання кнопки Оформити замовлення відбувається перехід до відповідної сторінки (рис. 4.6), де необхідно ввести деяку інформацію, зокрема, контактні дані отримувача.

Оплата

Доставка

Гарантія

Про нас

Увійти

Створити акаунт

Кошик (52605 грн.)

Ваше замовлення

Ваше ім'я

max24

Електронна пошта

Мобільний телефон

+380671234567

Адреса доставки

м. Закарпатка, вул. Тимирязівська, 5

Підсумкова вартість

52605,00 грн.

Підтвердити

Рисунок 4.6 – Оформлення замовлення товарів

Користувач з правами адміністратора має можливість роботи з переліком товарів, що пропонуються на продаж (рис. 4.7), зокрема, існує можливість додавання та видалення товарів.

Оплата

Доставка

Гарантія

Про нас

Адміністрування

Привіт, Адмін

Вихід

Кошик (0 грн.)

▼ Список сутностей

Товари

Категорії

Шукати за назвою...

Q

Додати новий товар

| ID   | Назва                       | Категорія         | Ціна          | Дата оновлення |          |
|------|-----------------------------|-------------------|---------------|----------------|----------|
| 1025 | Apple iPod EarPods with Mic | Внутрішньослухові | 1199,00 грн.  | 06.10.2024     | Видалити |
| 1026 | Koss The Plug Black         | Внутрішньослухові | 519,00 грн.   | 06.10.2024     | Видалити |
| 1027 | Koss The Plug Blue          | Внутрішньослухові | 539,00 грн.   | 06.10.2024     | Видалити |
| 1028 | Koss The Plug Red           | Внутрішньослухові | 529,00 грн.   | 06.10.2024     | Видалити |
| 1029 | Sennheiser HD 4.50 BTNC     | Навушні           | 8019,00 грн.  | 07.10.2024     | Видалити |
| 1030 | Sennheiser HDR 175          | Навушні           | 4179,00 грн.  | 06.10.2024     | Видалити |
| 1031 | Sony WH-1000XM3 Black       | Навушні           | 8999,00 грн.  | 06.10.2024     | Видалити |
| 1032 | Sony WH-CR510 Black         | Навушні           | 1399,00 грн.  | 06.10.2024     | Видалити |
| 1033 | Sennheiser HD600 Silver     | Навушні           | 37160,00 грн. | 07.10.2024     | Видалити |

Рисунок 4.7 – Таблиця товарів

#### **4.5 Тестування програмного забезпечення для збору інформації про товари**

Виконано тестування програмного забезпечення для збору інформації про товари.

Виявлено, що програма для збору інформації про товари функціонує правильно та злагоджено. Вона реалізує всі функціональні вимоги та успішно виконує свою основну задачу підтримки процесу збору інформації про товари.

Розроблене програмне забезпечення дозволяє забезпечувати автоматизацію процесів, пов'язаних з підтримкою процесу збору інформації про товари.

#### **4.6 Висновки за розділом 4**

Описано програмне забезпечення для збору інформації про товари. Виконано тестування розробленого програмного забезпечення для збору інформації про товари. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу збору інформації про товари.

## ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для збору інформації про товари.

Визначено, що збір даних з вебсайтів полягає у видобуванні інформації з відкритих онлайн джерел та збереженні її в базах даних. Цей процес зазвичай здійснюється за допомогою скриптів або програм, які можуть обходити захисні механізми вебсайтів. Проаналізовано різні методи збору даних з вебсайтів. Проаналізовано застосування програмного забезпечення для збору та аналізу даних.

За результатами проведеного аналізу зроблено висновок, що у наш час існує досить багато програмних засобів для збору інформації про товари. Проте використання у програмних засобах збору інформації про товари ненадійних джерел інформації, що надають неактуальні або неточні дані про товари, може призвести до хибних висновків або неправильних рішень. Крім того, впровадження та налаштування під конкретні вимоги торгівельної організації деякого існуючого програмного забезпечення для збору даних про товари може вимагати значних фінансових інвестицій, особливо якщо йдеться про складні програмні системи з інтеграцією штучного інтелекту або машинного навчання. В деяких випадках інтеграція з різними програмними платформами та системами може бути складною та вимагати спеціальних технічних знань, також можливі проблеми сумісності з різними джерелами даних або іншими бізнес-системами. Тому актуальною є розробка програмного забезпечення для збору інформації про товари.

Сформульовано функціональні вимоги до програмного забезпечення для збору інформації про товари.

Для реалізації програмного забезпечення для збору інформації про товари обрано мову програмування C#, яка має велику кількість готових бібліотек і фреймворків, що можуть допомогти при створенні програм для

збору даних. Крім того, мова C# добре інтегрується з різними API, що дозволяє збирати дані з інтерфейсів сторонніх сервісів, зокрема, API торгових платформ або агрегаторів товарів. Іншою перевагою C# є кросплатформність через .NET Core, оскільки завдяки .NET Core, програми, написані на C#, можуть працювати не тільки на Windows, але і на інших операційних системах.

Для створення програмного забезпечення для збору інформації про товари обрано середовище розробки Microsoft Visual Studio, що дозволяє розробляти програми, які використовують ресурси для збору, обробки та зберігання даних, що є важливим для програмних систем збору інформації про товари, які можуть потребувати хмарних потужностей для обробки та аналізу даних у реальному часі. Крім того, середовище розробки Microsoft Visual Studio підтримує багатоплатформову розробку, оскільки разом з .NET Core дозволяє створювати програми, які можуть працювати не лише на Windows, але й на інших операційних системах, що розширює можливості використання програмного забезпечення для збору інформації на різних платформах.

Новизна роботи полягає у тому, що створено модель взаємодії користувача з програмним забезпеченням для збору інформації про товари, яка подана у вигляді діаграми прецедентів, та дозволяє розроблювати програмне забезпечення для збору інформації про товари.

Практична цінність роботи полягає у тому, що розроблено програмне забезпечення для збору інформації про товари.

Запропоновано структуру програмного забезпечення для збору інформації про товари. Визначено, що як базис для створення програмного забезпечення для збору інформації про товари обрано архітектуру модель-вигляд-контролер (Model-View-Controller), що реалізує клієнт-серверний підхід.

Розроблено базу даних програмного забезпечення для збору інформації про товари, яка відображає концептуальні зв'язки між обраними елементами у відповідній предметній області. Визначено, що основними сутностями бази

даних є такі: товари (Products), магазини (Partners), ціни на товари (Prices), замовлення (Orders), опис товару (Attributes) та інші.

Описано функціонування програмного забезпечення для збору інформації про товари.

Описано особливості реалізації програмного забезпечення для збору інформації про товари. Виконано проєктування інтерфейсу взаємодії користувача з програмним забезпеченням для збору інформації про товари.

Виконано тестування розробленого програмного забезпечення для збору інформації про товари. Результати тестування програмного забезпечення показали, що розроблена програма дозволяє забезпечити автоматизацію процесів, пов'язаних з підтримкою процесу збору інформації про товари.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Website Data Collection the Right Way [Electronic resource]. – Access mode: <https://coresignal.com/blog/website-data-collection/>.
2. What Is Web Scraping? How to Automate Web Data Collection [Electronic resource]. – Access mode: <https://www.g2.com/articles/web-scraping>.
3. How To Scrape Data from Any Website: 5 Code and No-Code Methods [Electronic resource]. – Access mode: <https://www.scrapin.io/blog/web-scraping>.
4. Collecting Information from Websites: A Basic Guide [Electronic resource]. – Access mode: <https://medium.com/@johnsoul182/collecting-information-from-websites-a-basic-guide-5be66c649ae8>.
5. How to extract data from a website? Ultimate guide to pull data from any website [Electronic resource]. – Access mode: <https://www.scrapingbee.com/blog/how-to-extract-data-from-a-website/>.
6. How to collect data from a website: a comprehensive guide [Electronic resource]. – Access mode: <https://blog.apify.com/collect-data-from-website/>.
7. Best Practices and Tools for Web Data Collection [Electronic resource]. – Access mode: <https://netnut.io/web-data-collection/>.
8. Best Tools for Online Data Collection [Electronic resource]. – Access mode: <https://codeless.io/online-data-collection-tools/>.
9. Tools to extract data from website [Electronic resource]. – Access mode: <https://www.softtr.io/blog/tools-to-extract-data-from-website>.
10. The Best 19 AI Website Scrapers You Haven't Heard Of [Electronic resource]. – Access mode: <https://nealschaffer.com/ai-website-scraper/>.
11. Best Web Scraping Tools in 2024 [Electronic resource]. – Access mode: <https://www.zenrows.com/blog/web-scraping-tools#what-are-they>.
12. Diffbot [Electronic resource]. – Access mode: <https://www.diffbot.com/data/product/>.
13. Webscraper.io [Electronic resource]. – Access mode: <https://webscraper.io/>.

14. Apify [Electronic resource]. – Access mode: <https://apify.com/>.
15. C# Tutorial. Introduction [Electronic resource]. – Access mode: <https://csharp.net-tutorials.com/getting-started/introduction/>.
16. C#. The modern, innovative, open-source programming language for building all your apps [Electronic resource]. – Access mode: <https://dotnet.microsoft.com/en-us/languages/csharp>.
17. Visual Studio 2022 (préversion) [Electronic resource]. – Access mode: <https://visualstudio.microsoft.com/fr/vs/preview/>.
18. View the structure of code by using different tool windows [Electronic resource]. – Access mode: <https://learn.microsoft.com/en-us/visualstudio/ide/viewing-the-structure-of-code>.
19. MVC Architecture [Electronic resource]. – Access mode: <https://medium.com/@simransrivastava01/mvc-architecture-9c73e8b0eb31>.

**ДОДАТОК А**  
**Технічне завдання**

## **Вступ**

Програмне забезпечення може використовуватися для підтримки процесу збору інформації про товари.

### **A.1 Підстава для розробки**

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Дослідження та програмна реалізація засобів збору інформації про товари», затверджене наказом Національного університету «Запорізька політехніка» № 398 від 07 жовтня 2024 р.

### **A.2 Призначення розробки**

Програмний продукт призначений для збору інформації про товари.

### **A.3 Основні вимоги до програми, що розробляється**

#### **A.3.1 Вимоги до функціональних характеристик**

Програмне забезпечення для збору інформації про товари забезпечує виконання таких функцій:

- відображення каталогу товарів;
- можливість реєстрації та авторизації користувачів програмного забезпечення для збору інформації про товари;
- можливість відображення діапазону цін на товари на основі зібраної інформації;
- підтримка можливості подання користувачу детальної інформації про обраний товари;
- підтримка можливості інформування користувача щодо цін на обраний товар у різних магазинах;

- можливість замовлення обраного товару;
- можливість додавання та видалення товарів у каталог (функція адміністратора).

### **А.3.2 Вимоги до інтерфейсу програми**

Інтерфейс програмного забезпечення для збору інформації про товари повинен бути зручним для користувачів. Повинна бути забезпечена можливість роботи в візуальному режимі.

### **А.3.3 Вимоги до надійності**

Програмне забезпечення для збору інформації про товари повинно забезпечити надійне функціонування.

### **А.3.4 Умови експлуатації**

Для експлуатації програмного забезпечення для збору інформації про товари необхідна наявність персонального комп'ютера.

### **А.3.5 Вимоги до складу та параметрів технічних засобів**

Для експлуатації програмного забезпечення для збору інформації про товари необхідно таке апаратне та програмне забезпечення:

- оперативна пам'ять об'ємом 4 ГБ і більше для забезпечення стабільної роботи при обробці даних;
- система зберігання даних для забезпечення швидкого читання і запису даних;

- процесор із тактовою частотою не менше 2 ГГц;
- наявність вебсерверу для обробки великого обсягу даних, запитів і обчислювальних операцій;
- встановлена операційна система та веббраузер на стороні клієнта.

### **А.3.6 Вимоги до маркування і пакування**

Програма для підтримки процесу збору інформації про товари може бути записана на будь-якому носії інформації.

На пакуванні повинна бути назва програми – «Програмне забезпечення для збору інформації про товари».

### **А.4 Вхідні дані до роботи**

Вхідними даними до програмного забезпечення для збору інформації про товари є інформація про товар (назва або категорія), який потрібен користувачу, інформація для замовлення товару, реєстраційні дані користувача та ін.

Вихідними даними програмного забезпечення для збору інформації про товари є створені у базі даних записи про товари, зокрема, ціни на товари у різних Інтернет магазинах та ін.

**ДОДАТОК Б**  
**Фрагмент тексту програми**

```

imgtspace SmrtChdsAst.WebUI
{
    public class ZastUserMngrt : UserMngrt<ZastUser>
    {
        public ZastUserMngrt(IUserStore<ZastUser> store)
            : base(store)
        { }

        public static ZastUserMngrt
Create(IdnttFctrOptions<ZastUserMngrt> options, IOwinContext
context)
        {
            var Mngrt = new ZastUserMngrt(new
UserStore<ZastUser>(context.Get<EFDbContext>()));
            Mngrt.UserValidator = new
UserValidator<ZastUser>(Mngrt)
            {
                AllowOnlyAlphnmrcUsrNms = false,
                VyngUniquePost = true
            };

            Mngrt.PasswordValidator = new PasswordValidator
            {
                VyngdLength = 6
            };

            Mngrt.UserLcktEnabledByDflt = true;
            Mngrt.DfltAccntLcktTimeSpan =
TimeSpan.FromMinutes(1);
            Mngrt.MaxFailedAccssAttmptsBeforeLckt = 5;

            var dtPrtctnPrvdr = options.DtPrtctnPrvdr;
            if (dtPrtctnPrvdr != null)
            {
                Mngrt.UserTokenPrvdr =
                    new
DataProtectorTokenPrvdr<ZastUser>(dtPrtctnPrvdr.Create("bht"));
            }
            return Mngrt;
        }
    }

    public class ZastSignInMngrt : SignInMngrt<ZastUser, string>
    {

```

```

        public ZastSignInMngrt(ZastUserMngrt userMngrt,
IAvtncnMngrt avtnctnMngrt)
            : base(userMngrt, avtnctnMngrt)
        { }

        public override Task<ClmsIdntt>
CreateUserIdnttAsync(ZastUser user)
        {
            return
user.GnrtUserIdnttAsync((ZastUserMngrt)UserMngrt);
        }

        public static ZastSignInMngrt
Create(IdnttFctrOptions<ZastSignInMngrt> options, IOwinContext
context)
        {
            return new
ZastSignInMngrt(context.GetUserMngrt<ZastUserMngrt>(),
context.Avtncn);
        }
    }

    public class PostSrvc : IIidnttMsgSrvc
    {
        public Task SendAsync(IdnttMsg msg)
        {
            bool enabled =
SmrtChdsAst.WebUI.Globals.Sttns.AstCnfguration.Post.Enabled;
            if (enabled)
            {
                string usrNm =
SmrtChdsAst.WebUI.Globals.Sttns.AstCnfguration.Post.Address;
                string pwd =
SmrtChdsAst.WebUI.Globals.Sttns.AstCnfguration.Post.Password;

                System.Stnt.Post.SmtpKlnt Klnt = new
System.Stnt.Post.SmtpKlnt(SmrtChdsAst.WebUI.Globals.Sttns.AstCnf
guration.Post.SmtpKlnt);

                Klnt.Port =
SmrtChdsAst.WebUI.Globals.Sttns.AstCnfguration.Post.SmtpPort;
                Klnt.DlvrtMethod =
System.Stnt.Post.SmtpDlvrtMethod.Stntwork;
                Klnt.UseDfltCredentials =
SmrtChdsAst.WebUI.Globals.Sttns.AstCnfguration.Post.UseDfltCrede
ntials;
            }
        }
    }

```

```

        System.Stnt.StntworkCredential credentials = new
System.Stnt.StntworkCredential(usrNm, pwd);
        Klnt.EnableSsl =
SmrtChdsAst.WebUI.Globals.Sttns.AstCnfguration.Post.EnableSsl;
        Klnt.Credentials = credentials;

        var post = new
System.Stnt.Post.PostMsg("admin@smrtChdsAst.com",
msg.Destination);
        post.Subject = msg.Subject;
        post.Body = msg.Body;
        return Klnt.SendPostAsync(post);
    }
    return null;
}
}
}

```

```

public static void OnovlCenhs(int prdctId)
{
    mnmlCenhId = 0;
    mnmlCenhZnch = 0;
    mxmlCenhZnch = 0;
    lchlnk = 0;
    CenhsSmtr = false;

    foreach (var prtnr in prtnrRpstr.Prtnrs)
    {
        sttds = false;

        string spcfcImgt =
prdctRpstr.GetSpcfcImgt(prdctId, prtnr.Imgt);
        string imgt = !string.IsNullOrEmpty(spcfcImgt) ?
spcfcImgt : prdctRpstr.GetPrdctImgt(prdctId);

        switch (prtnr.Imgt)
        {
            case "Rztk":
                PrcsRztkZapyt(prdctId, imgt);
                break;
            case "Erbht":
                PrcsErbhtZapyt(prdctId, imgt);
                break;
            case "Prtttn":
                PrcsPrtttnZapyt(prdctId, imgt);
                break;
        }
    }
}

```

```

    }

    if (sttds)
    {
        var cenhToAdd = new Cenh()
        {
            PrdctID = prdctId,
            PrtnrID = prtnr.PrtnrID,
            Tdfg = TdfgTime.Now,
            Znch = decimal.Parse(cenh),
            IsOutStck = isOutStck
        };

        var savedCenh = cenhRpstr.Insert(cenhToAdd);

        if (lchlnk == 0)
        {
            mnmlCenhZnch = cenhToAdd.Znch;
            mnmlCenhId = savedCenh.CenhID;
        }
        else
        {
            if (cenhToAdd.Znch == mnmlCenhZnch)
            {
                CenhsSmtr = true;
            }
            else if (cenhToAdd.Znch < mnmlCenhZnch)
            {
                mnmlCenhZnch = cenhToAdd.Znch;
                mnmlCenhId = savedCenh.CenhID;
                CenhsSmtr = false;
            }
        }

        if (cenhToAdd.Znch > mxmlCenhZnch)
        {
            mxmlCenhZnch = cenhToAdd.Znch;
        }

        lchlnk++;
    }
}

if (!CenhsSmtr)
{
    cenhRpstr.SetAsMnml(mnmlCenhId);
}

```

```

    }

    prdctRpstr.OnovlCenhZnch(prdctId, mnmlCenhZnch,
mxmlCenhZnch);
    }

    private static void PrcsRztkZapyt(int prdctId, string
imgt)
    {
        using (WebKlnt wc = new WebKlnt())
        {
            wtc.Codrvk = Codrvk.UTF8;
            string rspns = wtc.DownloadString(bstl + zapyt);

            if (!string.IsNullOrEmpty(rspns))
            {
                string odncStrt = "<div class=\"g-i-list ";
                if (Regex.Matches(rspns, odncStrt).Count > 0)
                {
                    var odncs = rspns.Split(new string[] {
odncStrt }, StringSplitOptions.None);

                    for (int i = 1; i< odncs.Length; i++)
                    {
                        int tTdr =
odncs[i].LastIndexOf(zaglvkEnd);
                        string zaglvk =
odncs[i].Substring(tFrm, tTdr - tFrm);

                        string imgtStrt = ">\r\n\t";
                        int nFrm = zaglvk.IndexOf(imgtStrt) +
imgtStrt.Length;
                        string odncImgt =
zaglvk.Substring(nFrm);

                        if
(odncImgt.ToUpper().Contains(imgt.ToUpper()))
                        {
                            string cenhElmntStrt = "<div
class=\"g-cenh-uah\">";
                            string cenhElmntEnd = "<span
class=\"g-cenh-uah-sign\">";

                            int pfrm =
odncs[i].IndexOf(cenhElmntStrt) + cenhElmntStrt.Length;

```

```

                                int pTgt =
odnecs[i].LastIndexOf(cenhElmntEnd);

                                cenh = odnecs[i].Substring(pfrm,
pTgt - pfrm).Replace("&thinsp;", string.Empty);
                                isOutStck =
(odnecs[i].Contains("Товары нема") || odnecs[i].Contains("Hemae"));
                                sttds = true;
                                break;
                                }
                                }
                                }
                                }
                                }

private static void PrcsErbhtZapyt(int prdctId, string
imgt)
{
    string myPrmtmrs = "klvtr=" + imgt + "&reset_old=1";
    using (WebKlnt wc = new WebKlnt())
    {
        wtc.Codrvk = Codrvk.UTF8;
        wtc.Hdrts[HttpZapytHdrt.ContentType] = "Zast/x-
www-form-urlencoded";

        try
        {
            string rspns = wtc.UploadString(URI,
myPrmtmrs);

            if (!string.IsNullOrEmpty(rspns))
            {
                if (!rspns.Contains("<zaglvk>Помык"))
                {
                    string cenhElmntStrt = "<span
odncprop=\"cenh\">";

                    string cenhElmntEnd = "
ррН</span></div></div><div id=\"pb-left-column\"><h2>";

                    int pfrm =
rspns.IndexOf(cenhElmntStrt) + cenhElmntStrt.Length;
                    int pTgt =
rspns.LastIndexOf(cenhElmntEnd);

```

```

        cenh = rspns.Substring(pfrm, pTgt -
pfrm).Replace(".", ","); ;
        isOutStck = cenh.Contains("<span
class=\"warning\">HI</span>");
        sttds = true;
    }
}
}
catch(WbExceptn ex)
{
    if (ex.Sttds == WbExceptnSttds.ProtocolError)
    {
        var rspns = ex.Rspns as HttpWebRspns;
        if (rspns != null && rspns.SttdsCode ==
HttpSttdsCode.NotFound)
        {
            sttds = false;
            return;
        }
    }
    throw ex;
}
}

private static void PrcsPrtttnZapyt(int prdctId, string
imgt)
{
    var bstl = "http://prtttn.ua/";
    var zapyt = "ktlgsearch/result/?q=" + imgt;

    using (WebKlnt wc = new WebKlnt())
    {
        wtc.Codrvk = Codrvk.UTF8;
        string rspns = wtc.DownloadString(bstl + zapyt);

        if (!string.IsNullOrEmpty(rspns))
        {
            string odncStrt = "<div class=\"ktlglistline-
odnc-container\";

            if (Regex.Matches(rspns, odncStrt).Count > 0)
            {
                var odncs = rspns.Split(new string[] {
odncStrt }, StringSplitOptions.None);

                for (int i = 1; i < odncs.Length; i++)

```

```

{
    string zaglvkStrt = "<div
class=\"ktlglist-odnc-imgt\">\r\n          <a\";
    string zaglvkEnd = "</a>\r\n
</div>\r\n          <div class=\"clli-code-avail-wrap\">\r\n\";
    int tFrm =
odncs[i].IndexOf(zaglvkStrt) + zaglvkStrt.Length;
    int tTdr =
odncs[i].LastIndexOf(zaglvkEnd);
    string zaglvk =
odncs[i].Substring(tFrm, tTdr - tFrm);

    string imgtStrt = "\">\";
    int nFrm = zaglvk.IndexOf(imgtStrt) +
imgtStrt.Length;

    string odncImg = string.Empty;

    if (zaglvk.Contains(" "))
    {
        int nTgt = zaglvk.LastIndexOf("
");
        odncImg = zaglvk.Substring(nFrm,
nTgt - nFrm);
    }
    else
    {
        odncImg =
zaglvk.Substring(nFrm);
    }

    if
(odncImg.ToUpper().Equals(imgt.ToUpper()))
    {
        string cenhElmntStrt = "<span
class=\"cenh-znch UAH\">\";
        string cenhElmntEnd =
"</span><span class=\"cenh-symbol\";

        int pfrm =
odncs[i].IndexOf(cenhElmntStrt) + cenhElmntStrt.Length;
        int pTgt =
odncs[i].IndexOf(cenhElmntEnd);

        cenh = odncs[i].Substring(pfrm,
pTgt - pfrm).Replace("&nbsp;", string.Empty);

```



```

var rtgdData = new RtgdData();
rtgdData.Znchs["kntrk"] = "Errors";
rtgdData.Znchs["dilr"] = "General";
rtgdData.Znchs["excptn"] = excptn;

Rspns.ContentType = "text/html";
Rspns.SttdsCode = 500;
if (httpExcptn != null)
{
    Rspns.SttdsCode = httpExcptn.GetHttpCode();
    switch (Rspns.SttdsCode)
    {
        case 403:
            rtgdData.Znchs["dilr"] = "Http403";
            break;
        case 404:
            rtgdData.Znchs["dilr"] = "Http404";
            break;
    }
}

IKntrk errorsKntrk = new
SmrtChdsAst.WebUI.Kntrks.ErrorsKntrk();
var rc = new ZapytContext(new
HttpContextWrapper(Context), rtgdData);
errorsKntrk.Execute(rc);
}
}
}

```

**ДОДАТОК В**  
**Слайди презентації**

**Міністерство освіти і науки України**  
**Національний університет «Запорізька політехніка»**  
**Кафедра програмних засобів**  
**Дослідження та програмна реалізація**  
**засобів збору інформації про товари**

Виконав: Олег ФАЛАЛЄЄВ

ст. групи КНТ-113м

Керівник: Андрій ОЛІЙНИК

д.т.н., професор НУ «Запорізька політехніка»

Рисунок В.1 – Слайд 1

## **Об'єкт, предмет та мета роботи**

*Об'єкт дослідження* – процес розробки програмного забезпечення для збору інформації про товари.

*Предмет дослідження* – програмні засоби для підтримки збору інформації про товари.

*Метою роботи є* дослідження та розробка програмного забезпечення для збору інформації про товари.

Рисунок В.2 – Слайд 2

## Завдання роботи

Для досягнення поставленої мети у дипломній кваліфікаційній роботі бакалавра необхідно розв'язати такі задачі:

- виконати аналіз предметної області та програмних засобів для збору інформації про товари;
- розробити модель взаємодії користувача з програмним забезпеченням для збору інформації про товари;
- здійснити проектування програмного забезпечення для збору інформації про товари;
- створити програмне забезпечення для збору інформації про товари;
- виконати тестування розробленого програмного забезпечення для збору інформації про товари.

3

Рисунок В.3 – Слайд 3

## Порівняння існуючих аналогів

| Критерій порівняння                                           | Diffbot | Webscraper.io | Apify |
|---------------------------------------------------------------|---------|---------------|-------|
| Використання засобів штучного інтелекту та машинного навчання | +       | –             | +-    |
| Масштабованість                                               | +-      | +             | +     |
| Розширення для веббраузера                                    | –       | +             | –     |
| Гнучкість у налаштуванні                                      | +-      | +             | +     |
| Стійкість до змін у верстці вебсторінок                       | +-      | +             | +-    |
| API для інтеграції                                            | +       | +-            | +     |

4

Рисунок В.4 – Слайд 4

## Порівняння мов програмування

| Критерій порівняння мов програмування                      | Мова програмування |        |      |
|------------------------------------------------------------|--------------------|--------|------|
|                                                            | C#                 | Python | Ruby |
| Засоби інтеграції з різними API для можливості збору даних | +                  | +-     | +-   |
| Бібліотеки для збору даних                                 | +                  | +      | +-   |
| Багатоплатформність                                        | +                  | +      | +-   |
| Зручність для створення ПЗ для збору інформації про товари | +                  | -      | -    |
| Активна підтримка спільноти                                | +                  | +      | +-   |

5

Рисунок В.5 – Слайд 5

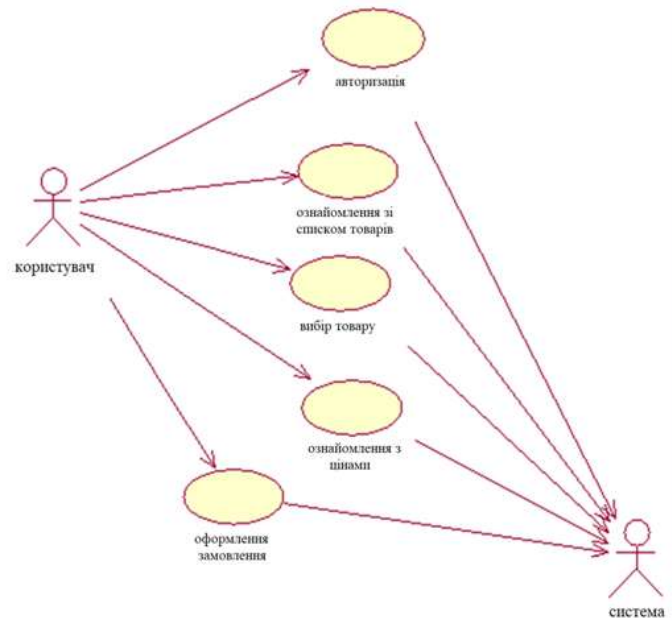
## Порівняння середовищ розробки

| Критерій порівняння середовищ розробки | Середовища розробки     |         |               |
|----------------------------------------|-------------------------|---------|---------------|
|                                        | Microsoft Visual Studio | Eclipse | IntelliJ IDEA |
| Підтримка багатоплатформової розробки  | +                       | +       | +-            |
| Потужний редактор коду                 | +                       | +-      | -             |
| Інструменти для налагодження           | +                       | +-      | +             |
| Можливість роботи з хмарними сервісами | +                       | +-      | +-            |
| Засоби інтеграції                      | +                       | -       | +-            |

6

Рисунок В.6 – Слайд 6

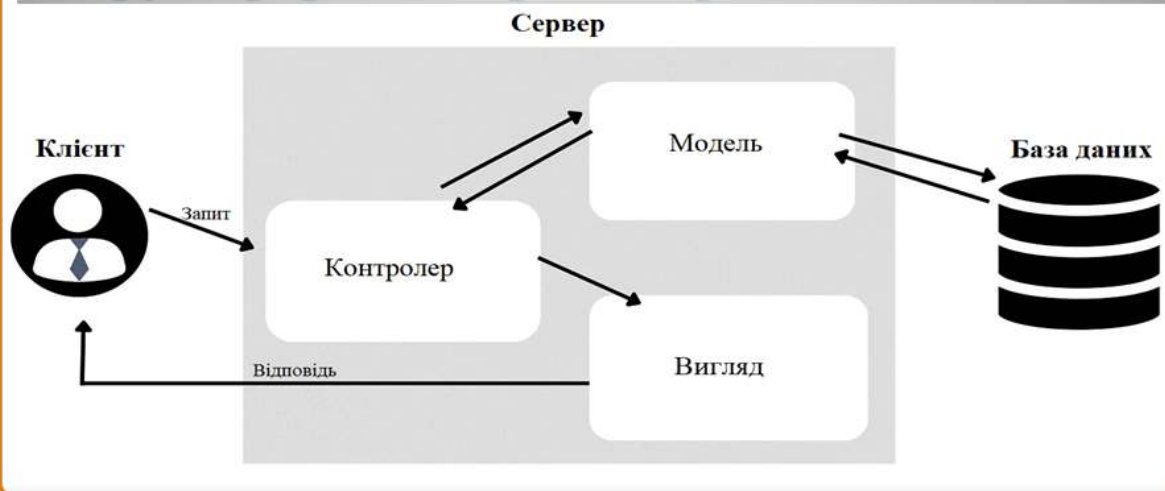
## Модель взаємодії користувача з програмним забезпеченням



7

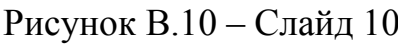
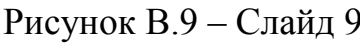
Рисунок В.7 – Слайд 7

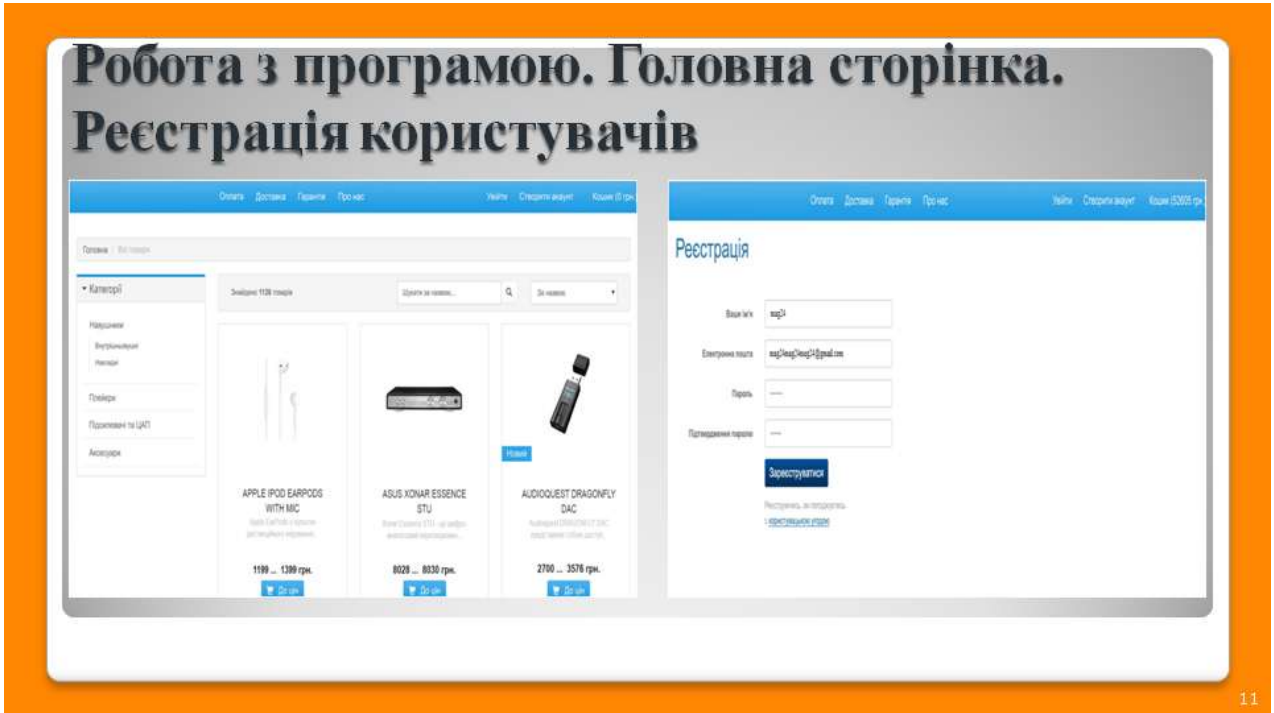
## Структура програмного забезпечення для збору інформації про товари



8

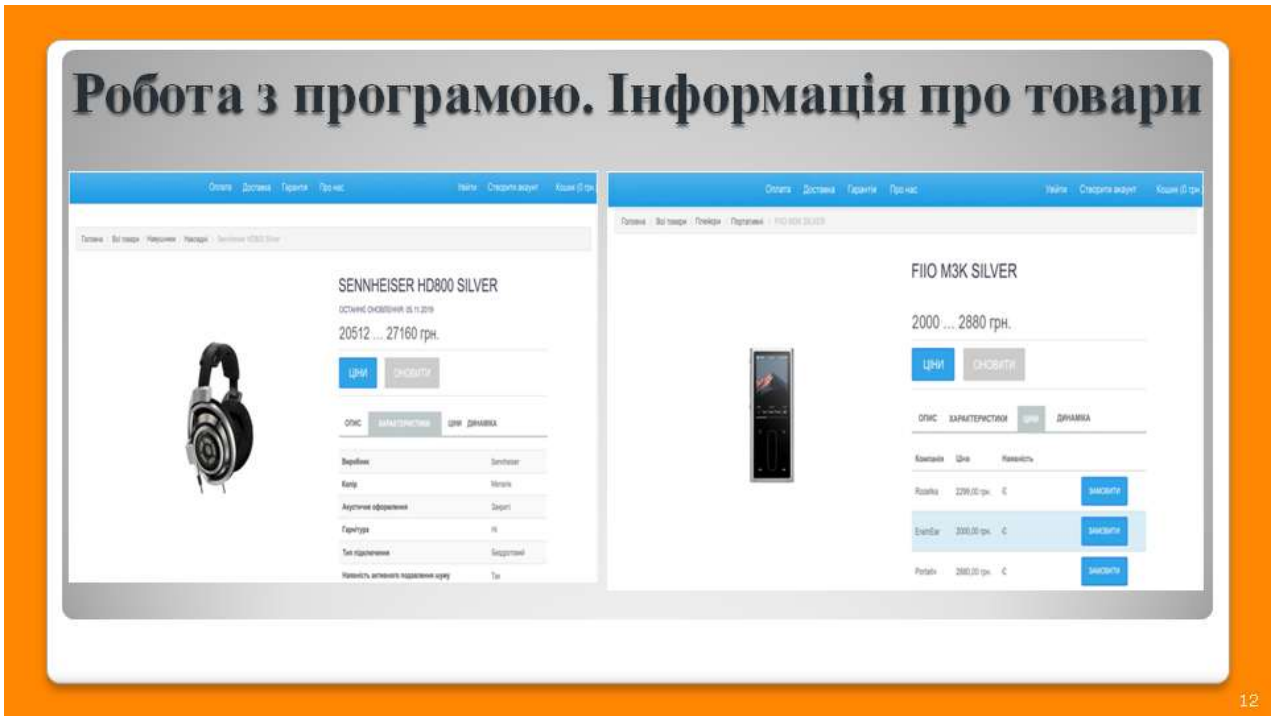
Рисунок В.8 – Слайд 8





11

Рисунок В.11 – Слайд 11



12

Рисунок В.12 – Слайд 12

## Робота з програмою. Кошик замовлення. Таблиця товарів

The screenshot displays two panels from a web application. The left panel, titled 'Ваш КОШИК' (Your Cart), shows a table with two items: 'Samsung HDD Silver' and 'FSC M24 Silver'. Each item has columns for 'Товар' (Product), 'Версія' (Version), 'Кількість' (Quantity), 'Ціна' (Price), and 'Сума' (Total). The right panel, titled 'Список сущностей' (List of entities), shows a table with columns for 'ID', 'Назва' (Name), 'Категорія' (Category), 'Ціна' (Price), and 'Дата виходу' (Release Date). The table lists various products like 'Apple iPad Air' and 'Samsung HDD'.

| Товар              | Версія  | Кількість | Ціна          | Сума          |
|--------------------|---------|-----------|---------------|---------------|
| Samsung HDD Silver | Екстерн | 2         | 27180,00 грн. | 54360,00 грн. |
| FSC M24 Silver     | Розетка | 1         | 2286,00 грн.  | 2286,00 грн.  |

Всього: 19657,00 грн.

Buttons: [Продовжити покупки](#) [Зформувати замовлення](#)

| ID   | Назва                      | Категорія               | Ціна          | Дата виходу |
|------|----------------------------|-------------------------|---------------|-------------|
| 1021 | Apple iPad Air16GB with Mo | Електроніка             | 1199,00 грн.  | 08.10.2024  |
| 1022 | Kasa The Plug Black        | Електроніка             | 629,00 грн.   | 08.10.2024  |
| 1023 | Kasa The Plug Blue         | Електроніка             | 629,00 грн.   | 08.10.2024  |
| 1024 | Kasa The Plug Red          | Електроніка             | 629,00 грн.   | 08.10.2024  |
| 1025 | Samsung HDD 4 TB 8TB       | Накопичувальні пристрої | 8019,00 грн.  | 07.10.2024  |
| 1026 | Samsung HDD 1TB            | Накопичувальні пристрої | 4179,00 грн.  | 08.10.2024  |
| 1027 | Sony WH-1000XM5 Black      | Навушники               | 9999,00 грн.  | 08.10.2024  |
| 1028 | Sony WH-CH710 Black        | Навушники               | 1399,00 грн.  | 08.10.2024  |
| 1029 | Samsung HDD Silver         | Накопичувальні пристрої | 27180,00 грн. | 07.10.2024  |

13

Рисунок В.13 – Слайд 13

## Висновки

В ході виконання дипломної кваліфікаційної роботи магістра було проаналізовано та досліджено процес розробки програмного забезпечення для збору інформації про товари.

Новизна роботи полягає в тому, що створено модель взаємодії користувача з програмним забезпеченням для збору інформації про товари, яка подана у вигляді діаграми прецедентів, та дозволяє розробляти програмне забезпечення для збору інформації про товари.

Практичне значення роботи полягає у тому, що розроблено програмне забезпечення для збору інформації про товари.

Усі завдання випускної дипломної роботи магістра повністю виконано.

14

Рисунок В.14 – Слайд 14