

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 2025 р.

**Дипломний проєкт  
на здобуття ступеня бакалавра  
за освітньо-професійною програмою «Інформаційні управляючі системи  
та технології»  
спеціальності 126 «Інформаційні системи та технології»  
на тему: «Інформаційна система моніторингу динаміки цін на обраний  
товар»**

Виконав:

студент ІV курсу, групи ІС-12

Главацький Ярослав Олександрович \_\_\_\_\_

Керівник:

Доцент каф. ІСТ, к.т.н., доцент

Деведжіогуллари Алла Вікторівна \_\_\_\_\_

Рецензент:

Доцент каф. ОТ, к.т.н., доцент,

Роковий Олександр Петрович \_\_\_\_\_

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_

Київ - 2025 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра інформаційних систем та технологій**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

**Главацькому Ярославу Олександровичу**

1. Тема проєкту «Інформаційна система моніторингу динаміки цін на обраний товар», керівник проєкту: доцент каф. ІСТ, к.т.н, доцент Деведжіогуллари Алла Вікторівна, затверджені наказом по університету від «23» травня 2025 р. № 1705-с
2. Термін подання студентом проєкту: 9 червня 2025р.
3. Вихідні дані до проєкту: мова програмування Python, система управління базами даних PostgreSQL, інтегроване середовище розробки Visual Studio Code.
4. Зміст пояснювальної записки: опис предметної області, аналіз існуючих рішень, формування вимог до системи, вибір технологій розробки, розробка інформаційної системи
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): Діаграма архітектури системи (А3), Діаграма варіантів використання (А3), Діаграма моделі бази даних (А3), Діаграма процесів (А3).

6. Дата видачі завдання 15 лютого 2025р.

### Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Вивчення вимог до інформаційної системи	14.04.2025–20.04.2025	
2	Аналіз джерел інформації та особливостей їх обробки	14.04.2025–20.04.2025	
3	Формалізація концепції системи та її модулів	14.04.2025–20.04.2025	
4	Дослідження сучасних технік збору та обробки інформації	21.04.2025–27.04.2025	
5	Визначення функціональних задач та вимог до реалізації	21.04.2025–27.04.2025	
6	Створення архітектури бази даних	28.04.2025–04.05.2025	
7	Побудова ядра системи збору даних	28.04.2025–04.05.2025	
8	Імплементация серверної частини та API	28.04.2025–11.05.2025	
9	Створення інтерфейсу користувача для взаємодії з системою	05.05.2025–11.05.2025	
10	Перевірка функціональності та продуктивності рішення	05.05.2025–11.05.2025	
11	Подання ДП на попередній захист		
	Подання ДП рецензенту		
	Подання ДП на основний захист		

Студент

Ярослав ГЛАВАЦЬКИЙ

Керівник

Алла ДЕВЕДЖІОГУЛЛАРИ

## АНОТАЦІЯ

Інформаційна система моніторингу динаміки цін на обраний товар

Проект містить 71 с. тексту, 25 рисунків, 8 таблиць, посилання на 15 літературні джерела, додатки та 4 конструкторські документи.

ВЕБЗАСТОСУНОК, МОНІТОРИНГ ЦІН, ІНФОРМАЦІЙНА СИСТЕМА, ПРОГНОЗУВАННЯ.

Об'єкт дослідження є автоматизований процес моніторингу та аналізу динаміки цін на товар з онлайн джерел.

Метою розробки є покращення процесу моніторингу та прогнозування цін на товари в інтернет-магазинах шляхом автоматизації збору, обробки та аналізу цінової інформації.

У дипломному проєкті проведено аналіз існуючих рішень для моніторингу цін, спроектовано архітектуру системи та базу даних. Розроблено серверну частину на і базу дани, а також клієнтський вебзастосунок. Система інтегрована з модулями автоматизованого збору даних та прогнозування цін. Практична цінність розробленої системи полягає в автоматизації відстежування змін цін та своєчасному сповіщенні користувачів про цінові тенденції, що дозволяє приймати обґрунтовані рішення щодо купівлі товарів.

## **SUMMARY**

Information system for monitoring the price dynamics of the selected product.

The project contains 71 pages of text, 25 figures, 8 tables, references to 15 literary sources, appendices, and 4 design documents.

**FORECASTING, INFORMATION SYSTEM, PRICE MONITORING, WEB APPLICATION.**

The object of research is the automated process of monitoring and analyzing price dynamics of goods from online sources.

The purpose of development is to automate the tracking and forecasting of the price of a selected product with user notification functionality.

In the diploma project, an analysis of existing solutions for price monitoring was carried out, the architecture of the system and the database were designed. The backend and the database were developed, as well as the client web application. The system is integrated with automated data collection and price forecasting modules. The practical value of the developed system lies in automating price tracking and timely notifying users about price trends, enabling informed purchasing decisions.

№ рядка	Формат	Позначення	Найменування	Кіл. аркушів	№ екз.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IC12.090БАК.005 ПЗ	Інформаційна система	71		
6			моніторингу динаміки цін на			
7			обраний товар.			
8			Пояснювальна записка			
9	A3	IC12.090БАК.005 Д1	Інформаційна система	1		
10			моніторингу динаміки цін на			
11			обраний товар.			
12			Діаграма архітектури системи			
13	A3	IC12.090БАК.005 Д2	Інформаційна система	1		
14			моніторингу динаміки цін на			
15			обраний товар.			
16			Діаграма варіантів використання			
17	A3	IC12.090БАК.005 Д3	Інформаційна система	1		
18			моніторингу динаміки цін на			
19			обраний товар.			
20			Діаграма моделі бази даних			
21	A3	IC12.090БАК.005 Д4	Інформаційна система	1		
22			моніторингу динаміки цін на			
23			обраний товар.			
24			Діаграма процесів			

				<b>IC12.090БАК.005 ТП</b>			
Зм.	Лист	№ докум.	Підпис				
Розробив	Главацький			Інформаційна система моніторингу динаміки цін на обраний товар. Відомість дипломного проекту	Літ.	Арк.	Аркушів
Перевірив	Деведжіогуллари				Т	1	1
					КПІ ім. Ігоря Сікорського Група IC-12		
Затв.							

**Пояснювальна записка**  
**до дипломного проєкту**  
**на тему: «Інформаційна система моніторингу динаміки**  
**цін на обраний товар»**

Київ – 2025 року

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП .....	5
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ .....	7
1.1 Опис процесу діяльності .....	7
1.2 Постановка задачі.....	7
1.2.1 Призначення системи .....	8
1.2.2 Цілі та задачі розробки .....	9
Висновки до розділу 1 .....	10
2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	11
2.1 Програмне рішення «PriceSpy» .....	11
2.2 Програмне рішення «CamelCamelCamel» .....	12
2.3 Програмне рішення «Price.ua» .....	13
2.4 Програмне рішення «Hotline».....	14
Висновки до розділу 2 .....	16
3 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ .....	18
3.1 Вимоги до системи в цілому .....	18
3.2 Вимоги до функціональних характеристик .....	20
3.3 Вимоги до видів забезпечення .....	22
Висновки до розділу 3 .....	23
4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ.....	24
4.1 Мова програмування Python .....	24
4.2 Фреймворк FastAPI .....	25
4.3 База даних PostgreSQL.....	25
4.4 Мова програмування JavaScript та фреймворк React .....	26
4.5 Технології контейнеризації та оркестрації контейнерів .....	26

					<b>IC12.090BAK.005 ПЗ</b>			
Зм.	Лист	№ докум.	Підпис		Інформаційна система моніторингу динаміки цін на обраний товар. Пояснювальна записка			
Розробив	Главацький	Перевірив	Деведжіогуллари					
Затв.					Т	2	71	
					КПІ ім. Ігоря Сікорського Група IC-12			

Висновки до розділу 4 .....	27
5 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	29
5.1 Структура системи.....	29
5.2 Функціональна модель системи .....	30
5.3 Модель бази даних.....	31
5.4 Передавання та обробка даних .....	36
5.5 Архітектура програмного забезпечення .....	37
5.6 Інтерфейс системи.....	50
Висновки до розділу 5 .....	54
6 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ .....	55
6.1 Змістовна постановка задачі .....	55
6.2 Математична постановка задачі .....	56
6.3 Обґрунтування методів розв'язання .....	58
6.3.1 Наївне прогнозування.....	58
6.3.2 Класичні моделі часових рядів .....	59
6.3.3 Регресійний аналіз із ознаками.....	59
6.4 Опис методу розв'язання.....	60
Висновки до розділу 6 .....	61
7 ТЕСТУВАННЯ СИСТЕМИ .....	62
7.1 Мета випробувань .....	62
7.2 Загальні положення.....	62
7.3 Результати випробувань .....	63
Висновки до розділу 7 .....	67
ВИСНОВКИ.....	68
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	70

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД — база даних.

Бекенд (Backend) — це програмно апаратна частина вебзастосунку або сервісу, що відповідає за його внутрішню логіку та функціонування.

Контейнеризація — це технологія віртуалізації на рівні операційної системи, яка дозволяє запакувати програмний застосунок разом з усіма його бібліотеками, залежностями та файлами конфігурації, в єдиний ізольований блок, що називається контейнером.

Парсинг (Parsing) — це процес автоматизованого збору та структурування даних з певного джерела за допомогою спеціально створеної програми.

СКБД — система керування базами даних.

Фреймворк — це набір інструментів, бібліотек, концепцій та стандартів, які допомагають розробникам швидше та ефективніше створювати програми, надаючи вже готові елементи розробки та структуру.

Фронтенд (Frontend) — це частина вебзастосунку, з якою безпосередньо взаємодіє користувач. Вона відповідає за візуальне представлення інтерфейсу, відображення даних та обробку дій користувача у браузері.

JSON (JavaScript Object Notation) — це легкий формат обміну даними, який використовує текстовий формат для представлення структурованих даних. Він широко застосовується для зберігання та передачі даних між сервером і клієнтом.

JWT (JSON Web Token) — це відкритий стандарт для безпечної передачі інформації між фронтендом та бекендом у вигляді JSON об'єкта. Він є компактним та самодостатнім, що робить його дуже популярним для аутентифікації та авторизації в сучасних вебдодатках та API.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

## ВСТУП

В сучасному світі ціни на товари змінюються дуже швидко та часто, це пов'язано з багатьма факторами, зокрема дуже поширені спекулятивні операції над ними. Динамічне ціноутворення — практика автоматичного коригування цін залежно від ринкових умов та ситуації в світі — це дуже популярний метод контролю ціни, який застосовується великою кількістю інтернет магазинів для того, щоб максимізувати прибутки та зменшити витрати. Споживачам та бізнесу потрібна можливість слідкувати за подібного роду змінами з метою прийняття обґрунтованих та оптимальних рішень щодо купівлі конкретного товару. Як зазначають дослідники, актуальність цієї теми обумовлена тим, що інтернет все частіше використовується для замовлення товарів, відтак доцільно створювати сервіси, які допомагають аналізувати ціни в різних інтернет магазинах та обирати оптимальні пропозиції. Цінові коливання можуть відбуватися щоденно або навіть щогодини внаслідок акцій, зміни попиту чи стратегій продавців, і відстежувати такі зміни вручну складно. Отже, розробка автоматизованої системи моніторингу та прогнозування цін на обраний товар є своєчасним та важливим завданням.

На сьогодні існує низка онлайн сервісів, що частково розв'язують проблему моніторингу цін. Зокрема, платформа PriceSpy забезпечує відстеження акцій та історії цін товарів у різних інтернет магазинах, включно з Amazon. Спеціалізований сервіс CamelCamelCamel дає змогу переглядати історію зміни ціни товарів на Amazon та отримувати сповіщення про зниження ціни до визначеного рівня. В Україні поширені такі агрегатори цін, як Price.ua та Hotline.ua, які дозволяють порівнювати вартість товарів у багатьох інтернет магазинах на одній платформі. Зокрема, ресурс Hotline.ua є одним із найпопулярніших: його щомісячна аудиторія становить близько 7,5 мільйонів користувачів.

Враховуючи наведене, метою дипломного проекту є розробка інформаційної системи, що забезпечує автоматизацію відстеження та прогнозування ціни обраного товару. Для досягнення зазначеної мети необхідно було виконати такі завдання: розробити модуль парсер для автоматизованого збору даних про ціни з

					ІС12.090БАК.005 ПЗ	Арк.
						5
Зм.	Лист	№ докум.	Підпис	Дата		

вибраних інтернет магазинів; створити конфігураційні засоби для гнучкого налаштування параметрів моніторингу; спроектувати та реалізувати базу даних для збереження зібраної інформації; розробити вебінтерфейс користувача для відображення динаміки цін, прогнозів та забезпечення зручної взаємодії користувача із системою; реалізувати алгоритм прогнозування цін на основі зібраних історичних даних; налаштувати систему сповіщень для інформування користувачів про зміну цін чи досягнення визначених порогових значень; а також виконати тестування та налагодження всіх компонентів системи.

Для реалізації поставлених завдань обрано сучасні технології та архітектурні підходи, що забезпечують ефективність та масштабованість системи.

Розроблена інформаційна система має широкий спектр можливих сфер застосування. В галузі онлайн торгівлі її можуть використовувати інтернет магазини для моніторингу цін конкурентів і оперативного коригування власної цінової політики. Аналітичні відділи компаній здатні застосовувати систему для дослідження ринкових тенденцій та прогнозування динаміки цін на різні товари з метою планування закупівель або маркетингових стратегій. Окрім того, сервіс буде корисним і для кінцевих споживачів, наприклад, у вигляді вебдодатку, що відстежує ціни на обрані користувачем товари та надсилає сповіщення про зниження ціни до бажаного рівня. Результати проєкту можуть знайти застосування скрізь, де важливо своєчасно отримувати інформацію про зміну цін та прогноз їх подальшої динаміки — від електронної комерції та бізнес аналітики до інформування пересічних покупців для прийняття вигідних рішень.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

# 1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Опис процесу діяльності

Сьогодні ціни на товари можуть змінюватися швидко та несподівано. Це може бути пов'язано з багатьма речами: коливанням курсу валют, змінами попиту чи пропозиції, сезонністю або навіть з політичною ситуацією в країні. Саме тому для багатьох компаній та покупців важливо відстежувати, як змінюється ціна на конкретний товар з часом.

У дипломному проєкті розглядається інформаційна система, яка дозволяє займатися моніторингом цін на товари та досліджувати їх динаміку з часом. Основна мета цієї системи — покращення процесу моніторингу та прогнозування цін на товари в інтернет-магазинах шляхом автоматизації збору, обробки та аналізу цінової інформації. Така система є актуальною в наш час, адже вона підвищує шанси покупця на прийняття оптимального рішення стосовно купівлі або продажу товарів, в яких він є зацікавлений. Відстеження та своєчасне повідомлення про можливе зниження або підвищення цін на товари наразі одна з найактуальніших тем сьогодення, адже практично кожен користується інтернет магазинами та робить онлайн замовлення товарів. З точки зору звичайного покупця пріоритетом є купівля товарів за найменшою ціною, яка тільки можлива. Інформаційна системи, що існують повинні надавати користувачу інформацію, що ціни в конкретному інтернет магазині не є спекулятивними та відповідають реальним даним.

## 1.2 Постановка задачі

Після детального аналізу існуючих рішень в рамках предметної області виявилось, що сьогодні на ринку вже є рішення, які допомагають користувачу в відстеженні та аналізі цінових коливань товару, загалом більшість з них тільки показують, як змінювалась ціна у минулому, але не можуть спрогнозувати, що буде з ціною далі. Також багато з них не мають зрозумілого інтерфейсу, або надають інформацію лише по одному магазину чи невеликій кількості товарів. деякі сервіси

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		7

взагалі оновлюють інформацію дуже повільно, через що користувачі можуть не виявити зміну ціни на товар.

Орієнтуючись на популярність вже існуючих рішень серед користувачів мережі можна зробити висновок, що задача відстеження та прогнозування цін на товари досі є актуальною і залишиться такою до тих пір, доки інтернет магазини існують.

З огляду на зазначене вище задачею дипломної роботи є розробка інформаційної системи для моніторингу динаміки та прогнозування цін на обрані користувачем товари. Система повинна надавати функціонал для користувачів, що дозволить їм спостерігати та аналізувати ціни на товари, що були обрані. Метою цієї системи є не тільки відстеження поточних цін, а й надання приблизних прогнозів щодо їх змін, такий функціонал дозволив би користувачам зрозуміти основний тренд в ціноутворенні конкретного товару та покращити їх досвід щодо купівлі товару за оптимальною ціною.

### 1.2.1 Призначення системи

Відповідно до поставленої задачі необхідно зрозуміти основні призначення системи та для кого така система буде розроблятися. Інформаційні системи такого плану повинні автоматично збирати інформацію про товари, які відслідковує користувач системи, тобто система має запропонувати користувачу інтерфейс та можливості, які замінять постійний ручний моніторинг ціни. Така система призначена для збору інформації з різних онлайн магазинів і зберігання цієї інформації для подальшого порівняння та аналізу. До користувачів такої системи можна віднести усіх людей, які користуються мережею Інтернет для здійснення онлайн покупок в магазинах. До користувачів такої системи можуть відноситися звичайні люди, які хочуть придбати щось особисто для себе або своєї сім'ї, або компанії, які купують товари оптом і повинні знати, коли краще зробити покупку.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		8

## 1.2.2 Цілі та задачі розробки

Сформована мета дипломної роботи дозволяє сформулювати відповідні цілі та задачі розробки на основі існуючих рішень на ринку та досвіду автора, отриманого у ролі звичайного покупця в інтернет магазинах. Основна ціль розробки інформаційної системи насправді може бути сформована наступним чином: необхідно спростити відстеження та аналіз зміни ціни на товар для покупців, що здійснюють покупки онлайн. Тепер відповідно можна сформулювати відповідні задачі розробки, які будуть безпосередньо перетинатись з цілю та метою розробки.

Першочергово потрібно розробити алгоритм збору даних, який має підтримувати різні інтернет магазини та торгівельні платформи. Такий алгоритм має збирати ціни на товари, що відслідковують користувачі, він має працювати автоматично та швидко для того, щоб забезпечити позитивний досвід для покупців, що користуються інформаційною системою.

Наступною задачею буде розробка бази даних, яка повинна зберігати дані, які обробив розроблений алгоритм збору, та надавати ці дані для подальшого аналізу. Необхідно обрати правильну СКБД для того, щоб операції INSERT — додавання даних до БД та GET — отримання даних з БД, які будуть основними для цієї системи, виконувалися швидко та безперебійно.

Після успішного виконання цих двох завдань можна перейти до частини аналізу зібраних та збережених даних: потрібно вибрати метод аналізу даних, що буде відносно простим в реалізації та ефективним у прогнозуванні, щоб надавати користувачеві актуальний прогноз та тренд, якому можна довіритися. Потрібно також розуміти, що час затрачений на тренування моделі та подальше прогнозування не повинен займати велику кількість часу, адже система має надавати користувачам саме оперативний прогноз, тобто можна пожертвувати точністю прогнозу, але він повинен бути швидким та відносно надійним.

Після виконання задачі прогнозування можна сфокусуватись на розробці зручного та зрозумілого інтерфейсу, що має на меті надати користувачу зручний

					ІС12.090БАК.005 ПЗ	Арк.
						9
Зм.	Лист	№ докум.	Підпис	Дата		

перегляд результатів моніторингу, відображення зміни ціни у вигляді графіку та повідомлень про зміну ціни на товар.

## Висновки до розділу 1

У цьому розділі ми розглянули як буде має працювати система для моніторингу та прогнозування змін цін на товари. Після аналізу ринку таких інформаційних систем було сформовано задачу розробки інформаційної системи, яка має на меті відстеження та прогнозування цін на товари в різних інтернет магазинах.

Призначення системи полягає в тому, щоб автоматизувати збір, обробку та аналіз інформації з інтернет магазинів, даючи змогу користувачам своєчасно зреагувати на зміни цін. Завдяки використанню такої системи користувачі зможуть економити час, уникнути переплати та загалом планувати свої витрати більш раціонально та ефективно.

В розділі також була сформована мета дипломної роботи, а саме — покращення процесу моніторингу та прогнозування цін на товари в інтернет-магазинах шляхом автоматизації збору, обробки та аналізу цінової інформації. Також були описані основні завдання, які мають бути вирішені для успішного завершення дипломної роботи такі як: розробка ефективного алгоритму збору даних, розробка системи для зберігання даних, впровадження методу аналізу цих даних для цінових коливань та створення зрозумілого інтерфейсу, до дозволить користувачам користуватися всім функціоналом інформаційної системи. Реалізація зазначених завдань має зробити таку систему затребуваною серед широкого кола покупців, що роблять покупки в мережі Інтернет.

Розробка цієї системи дозволить зробити моніторинг цін простішим і зручнішим для користувачів, підвищить ефективність прийнятих рішень та збереже від спекулятивної ситуації на ринку товарів.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		10

## 2 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

Для того, щоб розробити інформаційну систему моніторингу динаміки цін на обраний товар потрібно проаналізувати вже існуючі рішення, які є на ринку та користуються попитом серед користувачів. Потрібно також розуміти, що інформаційна система, яка розробляється має фокусуватись не тільки на ринку України, але на всьому світі. В цьому розділі аналізуються як рішення, що націлені тільки на український ринок, так і рішення, що використовуються по всьому світу.

### 2.1 Програмне рішення «PriceSpy»

Користувачі мережі Reddit [1] зазначають, що PriceSpy [2] є одним з вебсайтів, які найчастіше обирають для відстеження цін на товари, особливо для комп'ютерних комплектуючих. PriceSpy — це одна з найбільших платформ для порівняння цін на товари в Інтернеті, яка дозволяє користувачам отримувати інформацію про ціни на різних сайтах, а також порівнювати їх для вибору найкращої пропозиції. Платформа пропонує не лише моніторинг цін, а й історію змін вартості товарів. Головну сторінку платформи наведено на рисунку 2.1.

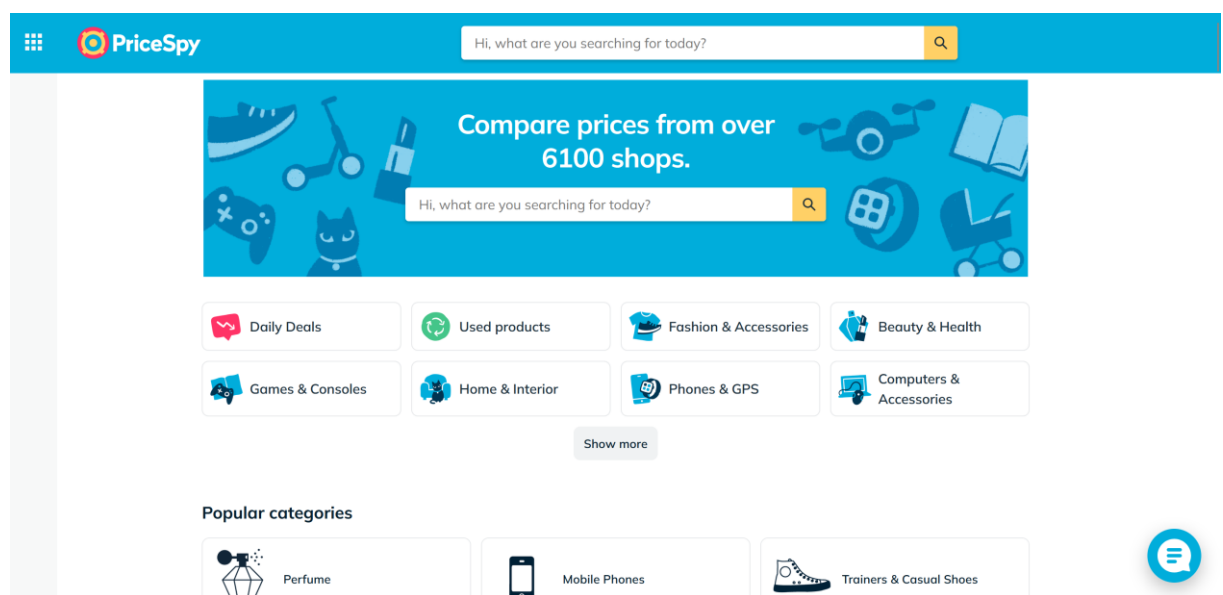


Рисунок 2.1 — Головна сторінка вебсайту PriceSpy

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		11

До переваг платформи можна віднести те, що платформа охоплює велику кількість категорій товарів, від електроніки до побутової техніки, що дозволяє зручно порівнювати ціни на товари. Платформа має внутрішні незалежні відгуки на товари, що допомагають в прийнятті рішення, яке не залежить від інтернет магазину. Також на платформі доступна історія змін ціни товару, що є дуже корисним, коли користувач хоче оцінити тренд та ситуацію на ринку загалом. До значної переваги в порівнянні з конкурентами можна віднести мобільний додаток, що пропонується для користувачів.

На жаль цей вебсайт має свої недоліки. З основних та значних можна відокремити відсутність великої кількості інтернет магазинів — платформа має дуже маленький каталог магазинів, що підтримуються — це робить її не дуже зручною для користувачів, яким важливо відслідковувати товари в широкому спектрі інтернет магазинів. Платформа не дає можливості додати товар для відстеження навіть в магазинах, що підтримуються. Також вебверсія застосунку немає функціоналу з оповіщення користувача про зміну ціни на товар, а також не надає функціональності, яка б прогнозувала ціни на товари.

## 2.2 Програмне рішення «CamelCamelCamel»

Amazon [3] є однією з найбільших платформ продажу товарів різних категорій. Для відстеження цін на платформі Amazon існує вебзастосунок CamelCamelCamel [4] — це платформа для моніторингу цін, що спеціалізується на відстеженні товарних позицій тільки для вебсайту Amazon. На рисунку 2.2 можна побачити головну сторінку цього вебзастосунку. З переваг CamelCamelCamel можна зазначити простий та зрозумілий інтерфейс застосунку, можливість перегляду ціни на товар в дуже великому проміжку часу, що може бути більше декількох років, наявність браузерного розширення, яке показує зміну ціни на товар та попереджує користувача про можливу нестабільність в ціні прямо під час перегляду товарів на вебсайті Amazon без необхідності переходити на сам вебсайт застосунку CamelCamelCamel. Також в цьому програмному рішенні реалізована

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		12

система сповіщення, що відстежує зміни на ціну товару та повідомляє користувача, якщо вона змінилася. Для активації цієї системи користувач має зареєструватися та додати товар до списку відстеження.

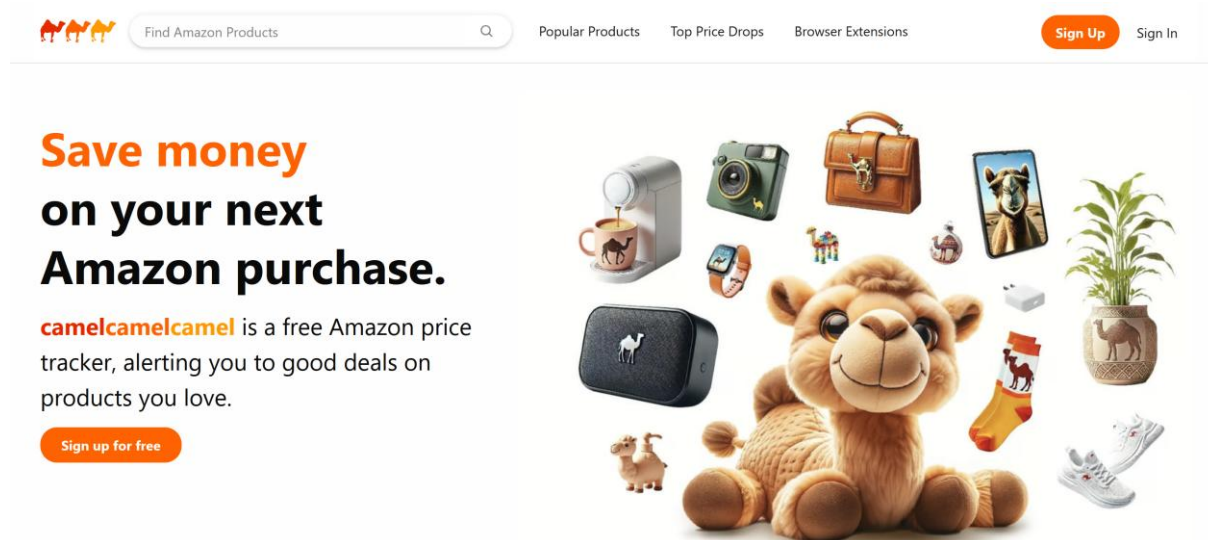


Рисунок 2.2 — Головна сторінка вебсайту CamelCamelCamel

Основним недоліком цієї платформи є підтримка тільки одного магазину — Amazon. Для багатьох користувачів це може стати проблемою, адже в багатьох країнах світу неможливо замовити товари звідси напряму. Наприклад, Amazon не відправляє посилки до України, що робить цю платформу нерелевантною для внутрішнього ринку нашої держави. Також з недоліків можна зазначити відсутність мобільного додатку та функціоналу з прогнозування ціни на товар, що може ускладнити прийняття рішення про покупку.

### 2.3 Програмне рішення «Price.ua»

Price.ua [5] — одна з найбільших українських платформ для порівняння цін, яка дозволяє користувачам порівнювати ціни на різноманітні товари, допомагаючи знаходити найкращі пропозиції та послуги серед великої кількості інтернет магазинів по всій Україні. На рисунку 2.3 зображено головну сторінку платформи Price.ua.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		13

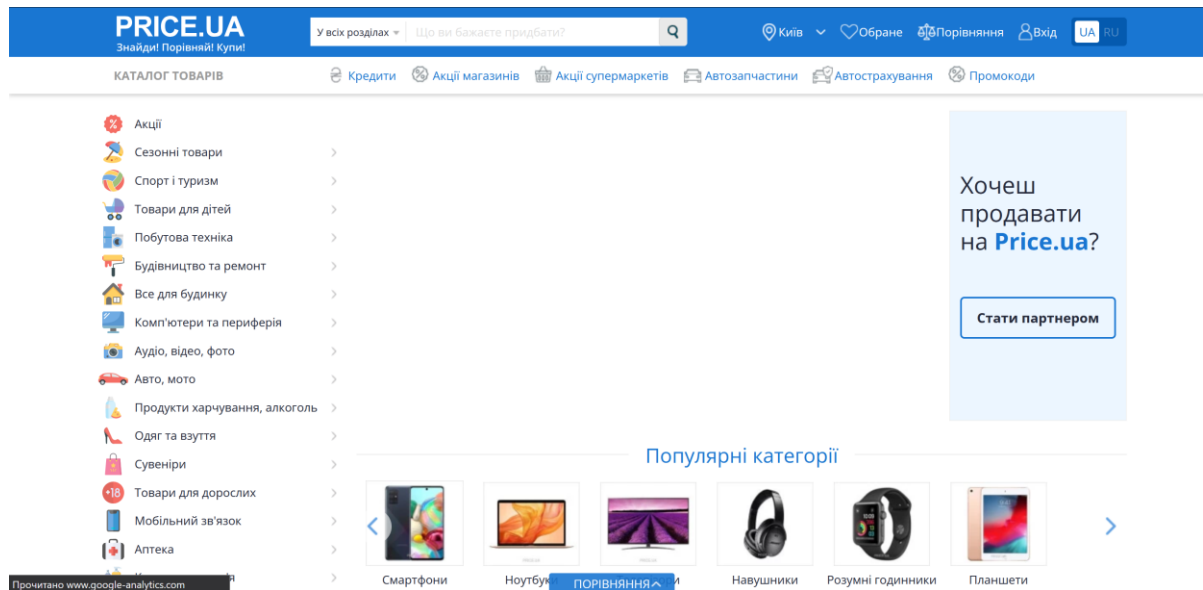


Рисунок 2.3 — Головна сторінка вебсайту Price.ua

До переваг цієї платформи можна навести підтримку багатьох магазинів та широкий вибір категорій товарів, що відслідковуються. Також тут присутня можливість додати товар до списку відслідковування. Це єдина платформа з зазначених, що надає вибір товарів відповідно до локації користувача.

З недоліків можна зазначити те, що попри наявність списку відстежування для зареєстрованого користувача, платформа не надає функціоналу з оповіщення про зміну ціни товару. Також платформа не надає можливість відстежувати зміну ціни на товар, тобто історія повністю недоступна та покупець не може оцінити та прийняти рішення, щодо купівлі конкретного товару. Як і в інших платформах, тут відсутня можливість прогнозувати ціну на товар та оцінювати тренд її зміни.

## 2.4 Програмне рішення «Hotline»

Hotline [6] — це найпопулярніша українська платформа для порівняння цін на товари, яка надає користувачам змогу швидко та зручно порівнювати вартість різноманітних продуктів у багатьох українських інтернет магазинах. Платформа орієнтована на широкий спектр товарних категорій — від побутової техніки і

електроніки до косметики та аксесуарів. На рисунку 2.4 зображена головна сторінка вебсайту Hotline.

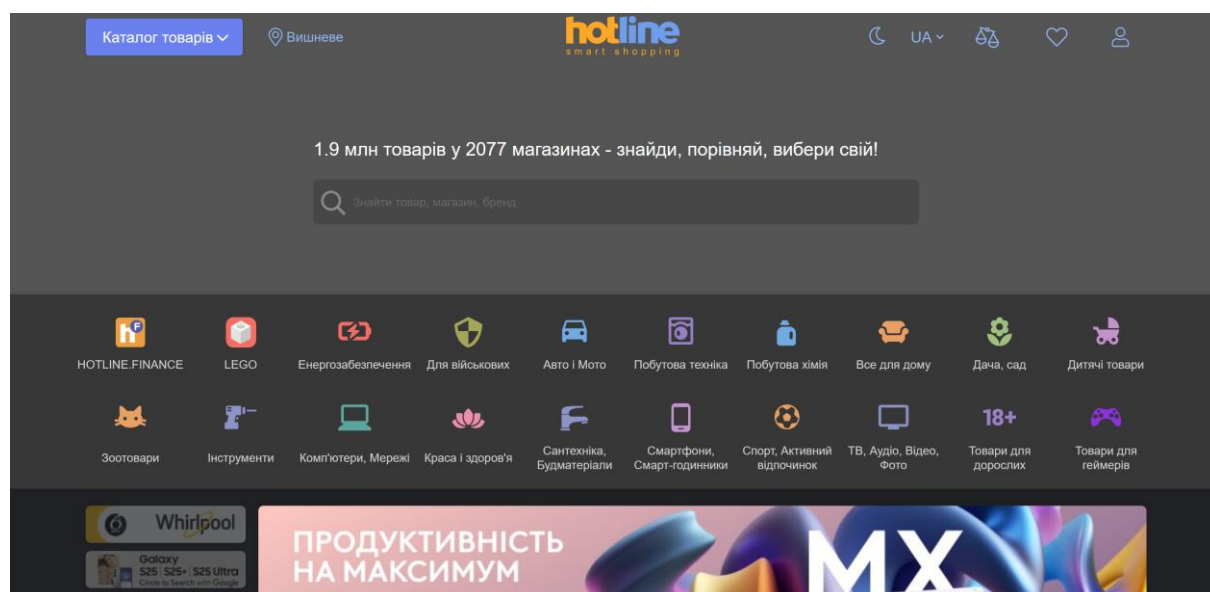


Рисунок 2.4 — Головна сторінка вебсайту Hotline

На українському ринку платформа Hotline є найрозвинутішою, саме тому вони має велику кількість переваг порівнюючи з конкурентами. Платформа має дуже великий список підтримуваних інтернет магазинів, їх тут тисячі. Також вона дозволяє порівнювати товари по характеристикам, порівнювати ціни на товари по інтернет магазинах. На платформі присутні незалежні відгуки від звичайних покупців про конкретні інтернет магазини, що спрощує вибір магазину для користувача.

Навіть найпопулярніша платформа має свої мінуси. Передивитись історію зміни ціни на товар можна лише зареєстрованому користувачу, що в більшості випадків просто витрачає час користувача на непотрібний ввід персональних даних та відтермінує його рішення про покупку товару. Також платформа не надає можливостей з прогнозування ціни на товар та не має свого мобільного застосунку.

					ІС12.090БАК.005 ПЗ	Арк.
						15
Зм.	Лист	№ докум.	Підпис	Дата		

Таблиця 2.1 — Порівняння готових рішень та розроблюваної інформаційної системи

	PriceSpy	Camel Camel Camel	Price.ua	Hotline	Інформаційна система
Самостійне додавання товарів користувачем.	-	-	-	-	+
Перегляд історії змін цін	+	+	+	+	+
Можливість прогнозування цін	-	-	-	-	+
Сповіщення про зміни цін	-	+	-	-	+
Підтримка різних інтернет магазинів	+	-	+	+	+

В таблиці 2.1 наведені результати порівняльного аналізу готових рішень та інформаційної системи, що розробляється в дипломному проєкті. З результатів видно, що жодна з систем не має можливості прогнозування ціни на товар, а також обмежує користувача в товарах, йому доступні тільки товари, що вже існують на платформі.

## Висновки до розділу 2

Після проведення ретельного аналізу вже наявних рішень на ринку стало зрозуміло, що існуючі платформи вже досягли певного рівня функціональності та продовжують розвиватися. Попри кожна з них не є ідеальною та має суттєві недоліки.

Платформа PriceSpy серед усіх виділяється тільки наявністю мобільного додатку. По іншим факторам вона є дуже схожою на конкурентів та надає аналогічний функціонал.

В свою чергу CamelCamelCamel підтримує лише один наявний інтернет магазин Amazon та жодного іншого. Для користувачів з територій, куди цей магазин не здійснює доставки, ця платформа є непотрібною. Але попри все ця

платформа має найкращий функціонал з аналітики цін на товари, вона має сповіщення про зміну цін та пропонує перегляд історії цін на обраний товар, також є браузерне розширення, що дозволяє аналізувати товари навіть не переходячи на вебсайт платформи, прямо в магазині Amazon.

Наступною проаналізованою платформою стала Price.ua, яка є націлена на український ринок. Ця платформа хоч і підтримує велику кількість інтернет магазинів але не має навіть базового аналітичного функціоналу, наприклад, перегляду історії зміни ціни на товар. Також її система відстеження товару не має функціоналу з оповіщення користувача про зміну ціни.

Найпопулярніша на українському ринку платформа Hotline також була проаналізована. Вона має найцікавіші функціональні рішення, які допомагають при покупці товарів. Платформа підтримує тисячі магазинів, де відслідковує товари, має широкий аналітичний інструментарій, але пропонує його тільки після реєстрації.

Загалом після порівняльного аналізу стало зрозуміло, що жодна з перелічених платформ не пропонує користувачу функціонал з прогнозування ціни на товар, який відстежується, а також обмежує користувача в каталозі товарів. Ці платформи не надають функціонал, щоб самостійно додати товар, який потрібен для аналізу та подальшого відстеження динаміки ціни, користувач має обирати серед вже запропонованого каталогу товарів, що може не містити товарів, які його цікавлять.

### 3 ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ

#### 3.1 Вимоги до системи в цілому

Система повинна мати мікросервісну архітектуру, тобто складатися з кількох незалежно працюючих компонентів, кожен з яких відповідає за окремий набір функцій. Основні компоненти системи бути такими:

- модуль парсингу та прогнозування цін;
- база даних;
- вебсервер;
- вебінтерфейс.

Модуль парсингу та прогнозування цін — повинен відповідати за автоматизований збір даних про ціни товарів з різних інтернет магазинів. Після чого за допомогою зібраних та збережених в БД даних повинен прогнозувати ціну на товар, що дозволить користувачу краще проаналізувати момент для придбання товару. Має бути реалізований як окремий модуль для того, щоб в подальшому використовувати його в інших сервісах.

База даних — необхідно мати реляційну базу даних для збереження зібраних даних та інформації про конфігурації.

Вебсервер — модуль, що тісно співпрацює з вебінтерфейсом, який звертається до вебсерверу, щоб отримати інформацію, яка потім відображається користувачеві. Також це модуль має відповідати за автентифікацію та авторизацію користувача інформаційної системи. Фактично, це інтерфейс, що дозволяє отримувати та записувати інформацію в базу даних, а також відправляти її для відображення на вебсторінці в браузері.

Вебінтерфейс — має існувати підсистема для взаємодії з користувачем та адміністратором. Для користувача повинен надаватись інтерфейс взаємодії з товарами: перегляд списку товарів та їхніх цін, додавання товару для збору даних, аналіз інформації про товар, що відстежується. Для адміністратора система має надавати функціонал для керування: редагування, додавання та видалення конфігурацій збору даних для інтернет магазинів).

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		18

Перераховані модулі мають взаємодіяти так, щоб забезпечити узгоджену роботу всієї інформаційної системи. Модуль парсингу цін має використовувати конфігурації, вивантажені з БД, для кожного інтернет магазину, щоб знати, звідки і яким чином витягувати цінові дані. Він періодично або за запитом має звертатись до вебсайтів магазинів, завантажувати HTML сторінки товарів та аналізувати їх.

Система має підтримувати декілька режимів роботи, що дозволить використовувати всі її можливості:

- автоматичний режим;
- ручний режим через вебінтерфейс.

Автоматичний режим. Модуль парсингу і прогнозування запускається автоматично за розкладом за допомогою планувальника завдань. У цьому режимі система самостійно ініціює обхід усіх товарів, що зберігаються в БД, та оновлює їхні ціни. Автоматичний запуск забезпечує регулярне оновлення даних без участі адміністратора.

Ручний режим через вебінтерфейс. Користувач може самостійно додати товар для відслідковування по URL, якщо його немає в БД, якщо товар в БД існує, то є можливість подивитись на детальну аналітику товару, дані для якої збираються за допомогою автоматичного режиму. Адміністратор через окремий інтерфейс може додавати нову конфігурацію інтернет магазину, після чого може одразу запустити парсинг товару з доданого інтернет магазину. Результати обробки негайно відображаються в інтерфейсі. Такий ручний режим зручний для тестування нових конфігурацій або оперативного отримання даних за запитом адміністратора.

Архітектура системи має бути спроектована з урахуванням можливого розширення функціональності в майбутньому.

Система має дозволяти відносно легко підключати нові інтернет магазини для моніторингу цін. Для цього має бути реалізований алгоритм, який є універсальним для будь-якого джерела даних. Для підтримки нового інтернет магазину має бути достатньо створити конфігурацію у форматі JSON файлу з правилами парсингу для сторінки товару. Отже, додавання підтримки нового

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		19

інтернет магазину має зводитися до налаштування та відправки JSON файлу, що дозволить в подальшому, при додаванні нового інтернет магазину, уникнути переробки системи.

У майбутньому можливе впровадження збору даних інтернет магазинів через їхні API. Використання API дозволило б отримувати інформацію про ціни більш надійно і швидко, мінімізуючи залежність від змін у структурі вебсторінок. API-запити, як правило, дають структуровані дані (JSON або XML) і менш схильні до збоїв, ніж парсинг HTML, що підвищить стабільність системи.

З ростом кількості товарів та частоти оновлень може знадобитися масштабування системи. Система може бути розгорнута в хмарному середовищі, що спростить динамічне виділення ресурсів. При збільшенні навантаження варто також розглянути перехід на черги повідомлень для розподілу завдань між процесами.

### 3.2 Вимоги до функціональних характеристик

Система призначена для виконання ряду ключових функцій, пов'язаних з моніторингом і аналізом цін. Нижче подані функціональні можливості системи.

Моніторинг цін в інтернет магазинах. Ця функція має періодично збирати ціни на товари з існуючих в системі магазинів. Система повинна підтримувати моніторинг декількох магазинів одночасно. Моніторинг має бути точним — ціна, зчитана парсером, повинна точно відповідати значенню на сайті (потрібна коректна робота парсеру з різними форматами чисел, валютою, регіональними форматами). Також оновлення цін повинно відбуватися з заданим інтервалом (наприклад, щодня) і не виходити за рамки допустимого відхилення часу. Якщо збір даних здійснюється на вимогу користувача, середній час відповіді від натискання кнопки "Оновити" до відображення нових даних має бути мінімальним. Система повинна реагувати на помилки: у разі помилки під час парсингу якогось товару — пропустити його, записати помилку і продовжити моніторинг інших позицій, щоб один збій не зупиняв всю функцію.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		20

Збереження даних у базі даних. Усі зібрані ціни та пов'язані дані мають бути надійно збережені в БД. Це дозволяє вести історію змін цін та забезпечувати даними модуль прогнозування. База даних повинна зберігати релевантну інформацію. Формат збереження має бути уніфікований: ціни нормалізовані до числа з плаваючою комою, а валюта має бути виокремлена та збережена окремо. Дати та час мають бути у стандартному форматі UTC. Вставка чергової порції даних не повинна блокувати таблиці надовго. Тому має використовуватись пакетна даних при парсингу усіх товарів в БД. Час виконання запитів для відображення даних користувачу має бути не більше часток секунди для типових запитів.

Авторизація та автентифікація користувачів. Система повинна мати функцію розмежування доступу: лише зареєстровані користувачі можуть отримувати повідомлення про зміну ціни товару та додавати товар до списку відслідковування. Має бути реалізована реєстрація нового користувача, вхід та вихід. Зберігання паролів має бути тільки в хешованому вигляді. Коректність перевірки прав: приватні дані повинні бути ізольовані — один користувач не може отримати через API дані іншого без відповідних прав, цього можна досягнути за допомогою JWT токена в тілі якого буде надаватись необхідна для авторизації та автентифікації інформація. Система повинна реагувати на некоректні спроби доступу стандартними повідомленнями. Час автентифікації має бути мінімізований — перевірка облікових даних у БД займає мілісекунди, тому загальний час відповіді при логіні в основному визначається мережею. Для зручності має бути передбачено JWT токен, дійсний певний час, щоб користувач не вводив пароль при кожній дії.

Прогнозування цін. На основі накопиченої історії змін цін система має генерувати прогноз майбутньої ціни та виводити тенденцію поведінки ціни (зростання/падіння). Модель має базуватися на надійному методі машинного навчання і демонструвати прийнятну похибку. Формат вихідної інформації має бути зручним, у вигляді прогнозованої ціни та індексу зміни тенденції. Розрахунок прогнозу повинен виконуватися достатньо швидко для інтерактивного використання — оптимально менше 1 — 2 секунд для одного товару. Це досягається завдяки тому, що обсяг даних для одного товару відносно невеликий

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		21

— один рік, а моделі обрані невисокої складності. Коли для товару недостатньо даних для прогнозу, модель повинна це розпізнати і виводити ціну, що не відрізняється від фактичної ціни товару.

Управління конфігураціями парсингу. Адміністратору необхідно мати можливість оновлювати налаштування системи: додавати нові інтернет магазини, змінювати налаштування існуючих, якщо сайт оновив структуру. Ця функція забезпечується через редагування конфігураційних файлів або відповідного інтерфейсу в вебзастосунку. Структура конфігурації має бути зручною та добре структурованою у вигляді JSON. Система при запуску повинна перевіряти наявність та правильність конфігурації і повідомляти про помилки в налаштуваннях. Бажано, щоб зміни конфігурації можна було застосувати без повного перезапуску всієї системи.

Усі ці функціональні можливості мають бути розроблені так, щоб забезпечити користувачу позитивний досвід: актуальні дані, швидкий відгук та зрозумілий інтерфейс взаємодії.

### 3.3 Вимоги до видів забезпечення

Найбільш складна математична частина буде застосована в модулі парсингу та прогнозування ціни на товар. Задача прогнозу зводиться до задачі регресії на часовому ряді: маючи послідовність значень ціни, необхідно передбачити її значення. Було обрано алгоритм машинного навчання, що найкраще відповідає такій природі даних. Одним із підходів є використання методів лінійної регресії. У простішому випадку можна вважати, що ціна змінюється приблизно лінійно або експоненціально, і необхідно підібрати параметри рівняння лінійної регресії методом найменших квадратів за наявними даними. Інший підхід — застосувати ARIMA-модель (AutoRegressive Integrated Moving Average) для часових рядів, яка враховує тренд і сезонність у даних. Також розглядалося використання моделей машинного навчання, таких як Random Forest Regression чи навіть згорткові

					IC12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		22

нейронні мережі, але з огляду на обсяг даних та швидкість прогнозування класичні методи лінійної регресії є більш зручними для дипломної роботи.

Виходячи з функціональних вимог, має бути спроектовано реляційну базу даних, яка буде містити кілька основних таблиць. Першою буде таблиця в якій зберігаються товари для відстежування, також мають бути присутні таблиці, де зберігаються історичні дані результатів парсингу та прогнозування ціни. Має існувати таблиця, де будуть зберігатися дані користувачів — які товари вони додали до списку відстежування. Не менш значущою таблицею буде таблиця зберігання конфігурацій інтернет магазинів, за допомогою якої ми можемо гарантувати, що конфігурації будуть доступні всім сервісам та адміністраторам інформаційної системи.

### Висновки до розділу 3

В цьому розділі було окреслено модульну мікросервісну архітектуру з окремими сервісами, що будуть працювати разом для забезпечення стабільної роботи інформаційної системи моніторингу динаміки цін на обраний товар. Першим модулем було описано модуль парсингу та прогнозування цін, який має виконувати функції збору даних та їх подальшого прогнозування Другим модулем було окреслено базу даних та вимоги до неї. Загалом, база даних повинна бути швидким та надійним джерелом для отримання та запису необхідної інформації іншими модулями системи. Третім модулем було описано вебсервер, який використовується в системі для того, щоб пов'язати роботу всіх інтерфейсів системи. Останнім модулем в системі став вебінтерфейс, що надає візуальні інструменти для взаємодії з системою через інтерфейс вебсерверу.

Також в розділі були описані функціональні вимоги та вимоги до видів забезпечення, серед них були автентифікація та авторизація користувача, прогнозування цін на товари за допомогою алгоритмів машинного навчання та управління правилами збору даних на онлайн магазинах, що створюються адміністратором системи.

					ІС12.090БАК.005 ПЗ	Арк.
						23
Зм.	Лист	№ докум.	Підпис	Дата		

## 4 ВИБІР ТЕХНОЛОГІЙ РОЗРОБКИ

У цьому розділі проаналізовано технологічний стек для майбутньої розробки інформаційної системи моніторингу динаміки цін на обраний товар та обґрунтовано вибір кожної технології. Кожна технологія розробки інформаційної системи по типу мови програмування, вебфреймворків для вебсерверу та вебінтерфейсу, бази даних, засобів збору даних, бібліотек та інструментів, розглядається з точки зору її переваг та особливостей в порівнянні з конкурентами. Загалом вибір кожної технології був оснований на швидкості та зручності розробки за допомогою цієї технології, саме тому в багатьох випадках були обрані найпопулярніші та передбачувані рішення. В подальшому під час підтримки розробленої системи її основною перевагою буде значно знижена коштовність в підтримці та розробці функціоналу, адже спеціалістів із знанням зазначених технологій на ринку тільки більше згідно статистики популярного сайту пошуку роботи Djinni [7].

### 4.1 Мова програмування Python

Серверна частина системи та модуль парсингу та прогнозування реалізовані за допомогою мови програмування Python [8]. Ця мова обрана насамперед через її простоту, універсальність та швидкість розробки з використанням необхідних фреймворків та бібліотек. Python має зрозумілий синтаксис з найпростіших структур в порівнянні з іншими мовами програмування. Ця мова вважається легкою для розробників завдяки читабельності коду та є мовою загального призначення, придатною для вирішення різноманітних завдань, тоді як інші мови мають дещо складніший синтаксис та розробка з їх допомогою може займати більше часу.

Використання Python як основної мови програмування в дипломному проєкті дозволило охопити як розробку вебсерверу так і модулю парсингу та прогнозування.

					ІС12.090БАК.005 ПЗ	Арк.
						24
Зм.	Лист	№ докум.	Підпис	Дата		

Python у даному проєкті виконує одразу кілька функцій. По-перше, на ньому побудований серверний застосунок, що обробляє запити від вебінтерфейсу. По-друге, Python використовується для програмного коду в модулі збору даних. По-третє, за допомогою цієї мови програмування було реалізовано тренування моделей машинного навчання та подальше їх використання у прогнозуванні зміни ціни на товари. Універсальність мови програмування Python дозволила реалізувати всі перелічені модулі у межах єдиної системи розробки програмного забезпечення, що дозволило дотримуватися конвенцій та загальноприйнятих вимог до написання коду, а також дозволила прискорити розробку великої частини кодової бази, адже потрібність у вивченні інших технологій була відсутня.

#### 4.2 Фреймворк FastAPI

Для реалізації вебсерверу був використаний фреймворк мови програмування Python під назвою FastAPI [9]. Для задач, що поставлені в дипломному проєкті цей фреймворк надає оптимальне співвідношення простоти розробки та швидкості роботи. Він є побудований на сучасних бібліотеках, які використовують ефективні методи обробки даних, таким чином дозволяючи ефективно розпоряджатися ресурсами машини на якій працює вебсервер. Цей фреймворк пропонує обширну та детальну документацію, яка навчає будувати робочі рішення вже з перших годин знайомства з технологією, що прискорює розробку серверної частини та дозволяє сфокусуватись на інших задачах дипломної роботи.

#### 4.3 База даних PostgreSQL

Для побудови інформаційної системи моніторингу динаміки цін ключовим є правильний вибір СКБД для зберігання, обробки та доступу до даних. У цьому проєкті як основу сховища даних було обрано PostgreSQL [10] — потужну реляційну СКБД з відкритим кодом та детальною документацією [11]. В дипломному проєкті використовується реляційний підхід, який забезпечує

					ІС12.090БАК.005 ПЗ	Арк.
						25
Зм.	Лист	№ докум.	Підпис	Дата		

структуроване зберігання інформації у вигляді таблиць із чітко визначеною схемою зберігання даних та підтримкою зв'язків між таблицями за допомогою зовнішніх ключів. Важливою вимогою проекту є підтримка аналітичних запитів до даних, це досягається потужним інструментарієм у вигляді SQL декларативної мови, це дозволяє легко виконувати запити агрегації, об'єднувати дані з кількох таблиць, використовувати вкладені запити, віконні функції, що безпосередньо виконується на стороні бази даних. Особливо слід підкреслити гнучкість PostgreSQL у роботі з JSON форматом. Для проекту це означає можливість зберігати файли конфігурацій у вигляді JSON, що є великою перевагою, адже алгоритм збору даних постійно використовує саме JSON конфігурації.

#### 4.4 Мова програмування JavaScript та фреймворк React

Вебінтерфейс в дипломному проекті реалізований з використанням бібліотеки React [12] на мові програмування JavaScript [13]. Сьогодні це є найпоширенішим рішенням для того, щоб розробляти браузерні вебінтерфейси швидко та ефективно. React забезпечує компонентний підхід до побудови інтерфейсу: додаток розбитий на незалежні компоненти, що спрощує розробку та дозволяє повторно використовувати вже написаний код в задачах побудови вебсторінки.

#### 4.5 Технології контейнеризації та оркестрації контейнерів

Для реалізації модульного підходу в дипломному проекті було використано технологію контейнеризації Docker [14] для розгортання всіх компонентів системи.

Контейнеризація в дипломному проекті надала кілька важливих переваг. По-перше, портативність: всю систему можна запустити на будь-якому сервері чи комп'ютері, де встановлено Docker, без необхідності вручну налаштовувати середовище, що дозволило уникнути проблеми в сумісності технологій, що встановлені на локальній машині, з технологіями, що використовуються в

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		26

дипломному проєкті. Технологія Docker гарантує, що середовище всередині контейнера завжди однакове, незалежно від системи. Також використовується ізоляція, за допомогою якої кожен сервіс працює в своєму контейнері, маючи тільки ті залежності, які йому потрібні, що забезпечує відносну безпеку компонентів системи до зовнішніх атак та дозволяє ефективніше слідкувати за подіями в кожному з модулів за допомогою логування. Альтернативою контейнеризації могло б бути розгортання на віртуальному чи фізичному сервері. Однак це складніше у підтримці — потрібно відслідковувати конфігурації, версії, забезпечувати, щоб на машині були всі залежності, а Docker значно спрощує цей процес.

Для управління контейнеризованими розробленими компонентами інформаційної системи моніторингу динаміки цін було обрано інструмент Docker Compose [15], що дозволяє ефективно управляти компонентами системи та налаштовувати їх взаємодію один з одним. У цьому інструменті використовується єдиний файл конфігурацій у форматі YAML [16], де визначаються усі налаштування для компонентів системи, такий формат файлу є дуже зручним для подальшого використання у налаштуванні взаємодії адже має чітку відокремлену структуру, яка також є інтуїтивно зрозумілою. Підхід з використанням оркестратора Docker Compose загалом допоможе не витратити час на розгортання контейнерів Docker окремо один від одного, а зробить цей процес швидким та незатратним по часу.

#### Висновки до розділу 4

В цьому розділі було детально описано вибір технологій, які будуть використовуватись в дипломному проєкті при розробці інформаційної системи. Загалом, вибрані технології ефективно узгоджені між собою та відповідають вимогам роботи.

Використання мови програмування Python та фреймворку FastAPI значно пришвидшить розробку інформаційної системи, адже ці технології є добре

					IC12.090BAK.005 ПЗ	Арк.
						27
Зм.	Лист	№ докум.	Підпис	Дата		

документованими та простими у використанні навіть для людини, що не володіє достатніми знаннями в предметній області. В подальшому, для підтримки розробленої інформаційної системи, можна буде легко знайти відповідних спеціалістів, адже поріг входу в використання цих технологій є відносно низьким, відповідно, підтримка системи буде обходитися дешевше ніж системи, які є розроблені на більш вузько направлених технологіях.

Система керування базами даних PostgreSQL була обрана для дипломної роботи через її надійність та відповідність сформованим вимогам. Вона є швидкою для запису даних та їх отримання, а також пропонує широкий спектр можливостей для аналітики, які також підтримуються та відпрацьовують доволі швидко. Загалом така система є достатньою для задач дипломного проекту.

Мова програмування JavaScript та її фреймворк React були обрані для розробки вебінтерфейсу інформаційної системи через їх популярність та компонентний підхід фреймворку, який дозволив швидко будувати необхідні сторінки представлення інформації та взаємодії з іншими компонентами системи. Загалом, вибір саме мови JavaScript та React також був обумовлений бібліотеками представлення даних, які вони пропонують, адже інформаційна система, яка розробляється, активно використовує підходи візуалізації даних для користувача, щоб спростити сприйняття динаміки зміни ціни.

Для зручності в запуску розроблених модулів інформаційної системи в проекті були обрані технології контейнеризації Docker та управління контейнерами Docker Compose відповідно. Ці дві технології сьогодні є стандартом при розробці будь яких систем, адже пропонують створення та управління ізольованих середовищ розробки, які не залежать від конфігурації локальної машини на якій проходить розробка. Це надає можливість також запускати усі модулі інформаційної системи незалежно один від одного.

					IC12.090BAK.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		28

## 5 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 5.1 Структура системи

Структура інформаційної системи моніторингу динаміки цін на обраний товар має багаторівневу архітектуру, яка представлена на кресленнику ІС12.090БАК.005 Д1. Вона складається з клієнтської та серверної частин, окремого сервісу збору даних та прогнозування, бази даних.

Frontend (WEB UI) — це модуль вебінтерфейсу, що працює у браузері користувача, через який користувач взаємодіє з усією системою. Відповідає за відображення інтерфейсу, введення та відображення даних, а також за взаємодію з сервером через мережеві запити.

Backend — модуль, що фактично складається з декількох частин. Першою частиною є Backend — сервер, що забезпечує бізнес логіку застосунку: прийом і обробку запитів від фронтенду, управління даними, а також надання необхідних даних у відповідь. Бекенд вебсервер запускається у Docker контейнері, що підтверджує сучасний підхід до розгортання контейнеризованих застосунків. Також цей вебсервер використовує API сервісу збору даних, підключене це API у вигляді бібліотеки, що дозволяє покращити швидкість виконання збору даних, якщо товар, який шукають, не існує в БД.

Parser and Predictor — окремий модуль парсингу вебсайтів та подальшого прогнозування ціни на товар. Це автономний сервіс, який періодично збирає актуальні дані про ціни товарів з зовнішнього вебсайту маркетплейсу та після успішного збирання даних прогнозує наступну ціну на товар. Цей модуль винесений в окремий контейнер, що дає змогу виконувати її незалежно від вебсервера.

Database — реляційна база даних, до якої мають доступ Backend (для операцій з даними користувачів та товарів) і сервіс парсингу та прогнозування (для запису отриманих цінових даних та читання списку товарів для прогнозування та парсингу). База даних також винесена в окремий контейнер, що підвищує її надійність роботи та полегшує взаємодію.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		29

Взаємодія між компонентами відбувається наступним чином: користувач працює через браузер з фронтендом (надсилає запити HTTP, отримує інтерфейс і результати). Фронтенд надсилає API запити до бекенду (наприклад, при додаванні нового товару або отриманні історії цін). Бекенд опрацьовує ці запити, звертаючись до бази даних для зберігання або вибірки інформації, і повертає відповідь фронтенду у форматі JSON. Сервіс парсингу та прогнозування незалежно від взаємодії з користувачем виконується за розкладом і здійснює HTTP запити до зовнішнього сайту магазину для отримання актуальних цін. Отримані дані він зберігає у базі даних, щоб потім використати для прогнозування ціни товару. База даних слугує централізованим сховищем: вона накопичує інформацію про зареєстрованих користувачів, перелік відстежуваних товарів та історію зміни цін.

Компонентна архітектура, побудована за принципами мікросервісів: фронтенд, бекенд, сервіс парсингу та прогнозування і СКБД працюють окремо один від одного, взаємодіючи по мережі, яку надає Docker Compose. Це підвищує гнучкість і масштабованість системи, спрощує її розгортання (усі частини ізольовано в Docker контейнерах) та підтримку.

## 5.2 Функціональна модель системи

В рамках розробленої системи було виділено дві ключових ролі користувачів.

Звичайний користувач — особа, що використовує систему для відстеження цін на обрані товари. Користувач може реєструватися в системі та виконувати вхід, додавати нові товари для моніторингу, переглядати поточні та історичні ціни на свої товари, додавати та видаляти товари зі списку відстеження, а також отримувати прогноз щодо зміни ціни.

Адміністратор — користувач з розширеними правами, що відповідає за підтримку системи. Адміністратор має можливість управляти даними в системі: переглядати, редагувати та додавати конфігурації сторонніх вебсайтів інтернет магазинів. Адміністратор також володіє всіма правами звичайного користувача (може авторизуватися і здійснювати ті ж операції моніторингу).

					ІС12.090БАК.005 ПЗ	Арк.
						30
Зм.	Лист	№ докум.	Підпис	Дата		

На кресленику ІС12.090БАК.005 Д2 овалами позначено варіанти використання системи. Також присутні актори в системі. Обидва актори мають сценарій входу до системи, але реєстрація доступна лише звичайному користувачу (адміністратор реєструється поза межами системи). Основні функції моніторингу — додавання товару, перегляд цін та аналітики можуть виконуватися як звичайним користувачем так й адміністратором. Адміністратор також має функції взаємодії з конфігураціями, які використовуються для парсингу цін з вебсайтів інтернет магазинів.

### 5.3 Модель бази даних

База даних складається з 6 таблиць, на кресленику ІС12.090БАК.005 Д3 наведено архітектуру бази даних та відображено взаємозв'язки між основними сутностями системи.

В таблиці 5.1 наведена схема таблиці бази даних products, яка призначена для зберігання основної інформації про товари, що відстежують користувачі системи. Кожен запис в ній відповідає окремому товару. Вона має відносно невелику кількість полів, які описані в таблиці 5.1, при необхідності нові поля можуть бути додані для більш детального опису товару.

Таблиця 5.1 — Опис таблиці products

Таблиця	Назва поля	Тип даних	Опис
products	id	SERIAL	Унікальний ідентифікатор товару
	marketplace_key	VARCHAR	Назва конфігурації для відповідного маркетплейсу
	url	VARCHAR	посилання на сторінку товару на відповідному маркетплейсі
	name	VARCHAR	Назва товару
	currency	VARCHAR	Валюта в якій вказана ціна товару

В таблиці 5.2 описана схема таблиці users в базі даних. Ця таблиця призначена для зберігання даних про користувачів системи. Кожен запис відповідає окремому зареєстрованому користувачу, який може підписуватися на товари для відстеження подальшої зміни ціни. Загалом, структура таблиці users є доволі простою, вона зберігає в собі логін користувача, його пошту та захешований пароль – ці поля використовуються для подальшої автентифікації користувача системи. Також додатково в таблиці реалізоване зберігання ролі користувача в текстовому форматі, це просте але ефективне рішення, якщо очікується, що в системі має бути лише дві активні ролі — адміністратор системи та звичайний користувач. В цьому випадку роль використовується системою для подальшої авторизації дій користувача в ній, це дозволяє ефективно та просто розмежувати доступи до окремих методів та функціоналу в системі.

Таблиця 5.2 — Опис таблиці users

Таблиця	Назва поля	Тип даних	Опис
users	id	SERIAL	Унікальний ідентифікатор користувача
	username	VARCHAR	Логін користувача
	hashed_password	VARCHAR	Пароль користувача, що зберігається в хешованому вигляді
	email	VARCHAR	Пошта користувача
	role	VARCHAR	Роль користувача в системі

В таблиці 5.3 описана схема таблиці parsed\_products в базі даних системи. Таблиця parsed\_products зберігає результати регулярного збору інформації про товари. Кожен запис у цій таблиці відповідає фактичній ціні товару в певний момент часу. Таблиця parsed\_products слугує історією змін даних про товари та використовується системою, коли необхідно візуалізувати динаміку зміни ціни на конкретний товар, або для прогнозування ціни товару. Для цього вона містить усі необхідні поля такі як зовнішній ключ товару до якого прив'язана ціна, конкретна

ціна на товар на момент часу, що зафіксований як часова мітка у відповідній колонці.

Таблиця 5.3 — Опис таблиці `parsed_products`

Таблиця	Назва поля	Тип даних	Опис
<code>parsed_products</code>	<code>id</code>	SERIAL	Унікальний ідентифікатор запису
	<code>product_id</code>	SERIAL	Зовнішній ключ з посиланням на таблицю товарів
	<code>price_proceeded</code>	NUMERIC	Ціна на товар в момент парсингу
	<code>etl_date</code>	TIMESTAMP	Час збору інформації з вебсайту

В таблиці 5.4 дипломного проєкту описана схема таблиці `users_products_subscriptions` в базі даних. Ця таблиця реалізує зв'язок багато до багатьох між користувачами та товарами, тобто зберігає підписки користувачів на оновлення конкретних товарів. Кожен запис у цій таблиці відповідає факту підписки певного користувача на певний товар. Таблиця виступає зв'язною між таблицями `users` та `products`.

Таблиця 5.4 — Опис таблиці `users_products_subscriptions`

Таблиця	Назва поля	Тип даних	Опис
<code>users_products_subscriptions</code>	<code>id</code>	SERIAL	Унікальний ідентифікатор запису
	<code>user_id</code>	SERIAL	Зовнішній ключ з посиланням на таблицю користувачів
	<code>product_id</code>	SERIAL	Зовнішній ключ з посиланням на таблицю товарів





Таблиця	Назва поля	Тип даних	Опис
	message	VARCHAR	Текст повідомлення для користувача
	notification_type	VARCHAR	Тип повідомлення
	created_at	TIMESTAMP	Дата та час створення запису повідомлення
	is_read	BOOLEAN	Ознака, що користувач прочитав повідомлення

Ця таблиця дозволяє системі повідомляти користувача про те, що ціна на товар, на який він є підписаний, змінилася. Фактично, логіку повідомлень в інформаційній системі реалізовано саме за допомогою даних з цієї таблиці, система регулярно звертається до таблиці, щоб перевірити, чи є повідомлення для користувача, який наразі активний в системі. Якщо повідомлення було прочитане користувачем, то в таблиці змінюється відповідне поле is\_read, що дозволяє системі зрозуміти, що повідомлення було отримане користувачем та він самостійно прочитав його.

#### 5.4 Передавання та обробка даних

У процесі роботи системи здійснюється передавання даних між компонентами (користувацьким інтерфейсом, сервером, базою даних, зовнішнім сайтом) та відповідна обробка цих даних.

Існує декілька типових операцій обробки даних в системі:

- додавання нового товару для відстежування;
- автоматичне відстежування ціни;
- перегляд історії та динаміки цін.

Додавання нового товару для відстежування — користувач через вебінтерфейс вводить посилання на товар. Ці дані передаються на вебсервер через HTTP запит. Вебсервер перевіряє отриману інформацію, створює новий запис

товару у базі даних та робить запит до зовнішнього сайту для отримання поточної ціни, після чого записує інформацію про ціну товару у відповідну таблицю.

Автоматичне відстежування ціни — сервіс парсингу працює у фоновому режимі. Його задача — через визначений інтервал часу опитувати зовнішні джерела цін для всіх товарів, що знаходяться у базі. Алгоритм роботи парсера наступний:

- отримати зі складу бази даних список всіх товарів, що потребують моніторингу;
- для кожного товару виконати HTTP запит до відповідного URL та завантажити HTML сторінку;
- проаналізувати HTML;
- записати отриману ціну в базу даних як новий запис в таблиці.

Перегляд історії та динаміки цін — коли користувач бажає переглянути динаміку цін на свій товар, вебінтерфейс надсилає GET запит до API вебсерверу із ідентифікатором товару. Вебсервер звертається до бази даних, вибираючи всі записи з таблиці, де зберігається історія цін на товари, для цього товару, та формує відповідь у форматі JSON. На основі отриманих даних вебінтерфейс будує графік зміни ціни.

## 5.5 Архітектура програмного забезпечення

Система моніторингу цін реалізована модульно, одним з модулів, що буде розглянутий першим, є вебсервер, який відповідає за обробку запитів користувачів системи. Розроблений код вебсерверу структуровано на пакети: `models`, `schemas`, `crud`, `auth`, що відповідає поділу на рівні: модель даних, логіка доступу до даних, схема валідації даних та функції аутентифікації та авторизації. Структуру файлів вебсерверу наведено на рисунку 5.1.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		37

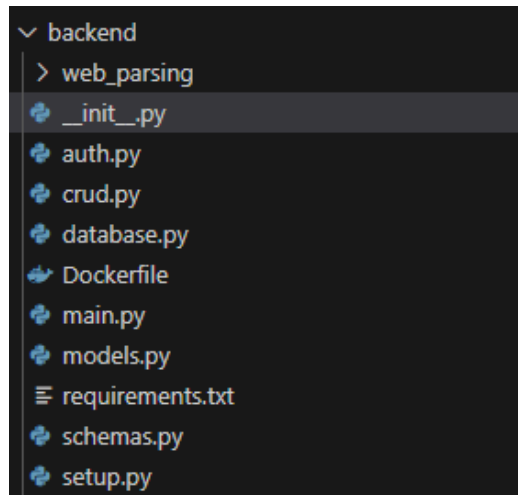


Рисунок 5.1 — Структура вебсерверу

Архітектура вебсерверу дотримується принципу розподілу відповідальності на рівні модулів. Застосунок має окремі файли для різних аспектів роботи:

- `models.py` — містить опис структури бази даних, моделі даних;
- `crud.py` — реалізує бізнес логіку та функції створення, читання, оновлення, видалення для об'єктів моделі даних;
- `auth.py` — відповідає за автентифікацію та авторизацію користувачів, містить функції для перевірки облікових даних і створення токенів доступу;
- `main.py` — головний модуль запуску вебсервера, який визначає точки доступу API та зв'язує HTTP запити з відповідними функціями обробки;
- `schemas.py` — містить схеми для визначення формату та валідації даних для запитів і відповідей;
- `database.py` — відповідає за встановлення з'єднання з базою даних та налаштування механізму підключення до неї.

У файлі `models.py` перелічені класи — моделі, вони відповідають структурі таблиць у базі даних. Фактично це лише шаблони для зберігання даних, які використовуються для валідації таблиць вже сформованих за допомогою інструменту автоматизації створення контейнерів Docker Compose.

Приклад моделі даних наведено на рисунку 5.2, в ній можна побачити структуру, а також допоміжні поля та атрибути, що допомагають у створенні та валідації таблиці. `ProductPricePrediction` — це модель, що відтворює таблицю для

зберігання результатів прогнозування ціни товару. В самому класі, що реалізує цю модель, в дужках вказується тип даних та додатково — атрибути колонки: `primary_key` — вказує, що колонка є первинним ключом, `index` — створює індекс бази даних для швидкого пошуку запису та оптимізації запитів, `ForeignKey` — атрибут, що встановлює зовнішній ключ для прив'язки до таблиці з якою колонка пов'язана відношенням.

```
class ProductPricePrediction(Base):
    __tablename__ = 'product_price_predictions'
    id = Column(Integer, primary_key=True, index=True)
    product_id = Column(Integer, ForeignKey('products.id'))
    predicted_price = Column(Numeric)
    change_index = Column(Numeric)
    etl_date = Column(DateTime)
```

Рисунок 5.2 — Приклад моделі даних ProductPricePrediction

Файл `crud.py` безпосередньо реалізує бізнес логіку роботи з даними. В ньому містяться функції, які виступають як проміжний рівень між рівнем API та рівнем доступу до бази даних.

В цьому файлі містяться наступні функції, всі вони приймають об'єкт сесії бази даних за замовченням.

`create_user` — створює нового користувача в базі даних. Функція приймає об'єкт `UserCreate`, що описаний в `schemas.py`. Після отримання початкових даних функція створює модель `User`, що описана в `models.py`, разом з цим вона викликає функцію хешування пароля користувача. Програмну реалізацію ініціалізації моделі користувача в цій функції наведено на рисунку 5.3.

```
db_user = models.User(
    username=user.username,
    hashed_password=hash_password(user.password),
    email=user.email,
    role=user.role,
)
```

Рисунок 5.3 — Приклад створення моделі User

`get_user_by_username` — функція, що знаходить користувача по переданому в параметрах імені. Повертає об'єкт `User`, якщо запис знайдено, якщо ні — пусте значення. Ця функція використовується в процесі автентифікації користувача.

`get_product_by_id` — отримує продукт за його ідентифікатором.

Вебінтерфейс побудовано на компонентах `React` для реалізації інтерфейсу користувача. Якщо запис знайдено, повертає об'єкт `Product`; якщо ні — поверне `None`. Ця функція використовується, коли потрібно отримати детальну інформацію про конкретний товар або, наприклад, перед видаленням або оновленням для того, щоб перевірити існування цього товару.

`get_products` — отримує список товарів, також впроваджена можливість пагінації. За замовчуванням дістає всі товари, але завдяки параметрам дозволяється обмежити результати та пропустити певну кількість записів.

`create_product` — додає новий товар до моніторингу для конкретного користувача. Приймає ідентифікатор користувача, який додає товар, об'єкт `ProductCreate`, описаний у файлі `schemas.py` та поточну ціну, яка отримана через функції сервісу парсингу, що будуть описані пізніше в матеріалі.

`create_parsed_product` — додає запис про товар, посилання на який пройшло через функції отримання актуальної інформації, до бази даних. Приймає унікальний ідентифікатор товару, ціну товару, яку отримали через парсинг вебсторінки цього товару, та позначку часу коли функція була викликана на вебсервері.

`get_subscription` — функція, що використовується для отримання товару, який відстежує користувач. Приймає унікальний ідентифікатор користувача та товару і повертає модель цього товару.

`create_subscription` — функція, що дозволяє створити підписку на товар для користувача.

`delete_subscription` — видаляє запис про підписку користувача на конкретний товар.

`get_price_stats` — функція, що використовується для отримання поточної ціни на товар, а також попередньої ціни на товар. Якщо ціну на товар знайдено, але

					ІС12.090БАК.005 ПЗ	Арк.
						40
Зм.	Лист	№ докум.	Підпис	Дата		

попередньої ціни в нього немає, то функція поверне значення поточної ціни та повідомлення, що ціна на товар не змінилась. Якщо ж знайдено і попередню ціну товару, то функція повертає окрім поточною ціни ще різницю між поточною та попередньою цінами, а також повідомлення, що ціна товару змінилась.

`get_basic_products_response` — отримує список товарів за допомогою функції `get_products`, передаючи додаткові параметри пагінації. Потім для кожного отриманого товару виконується формування базового представлення товару.

`get_detailed_product_response_by_id` — отримує детальну інформацію про товар за його ідентифікатором `product_id` з бази даних. Першим кроком вона знаходить основний запис товару у таблиці `products`, якщо товар не знайдено повертає `None`. Далі функція отримує останню зареєстровану ціну товару з таблиці `parsed_products`, впорядковану за датою завантаження, а також попередню ціну. Наступним кроком отримується останнє прогнозоване значення ціни товару із таблиці `product_price_prediction`. Після цього визначається поточна ціна, попередня ціна, обчислюється різниця цін та відхилення у відсотках. За різницею цін формується текстове пояснення: зростання, падіння або відсутність змін у ціні на товар. В результаті функція повертає словник із докладними даними про товар, включаючи ідентифікатор, ключ маркетплейсу, URL, назву, валюту, поточну ціну, різницю в ціні, рядок відхилення з відсотками, текстове пояснення відхилення, прогнозовану ціну та індекс зміни. Приклад структури словника, який повертає функція, можна побачити на рисунку 5.4.

```
return {
    "id": product.id,
    "marketplace_key": product.marketplace_key,
    "url": product.url,
    "name": product.name,
    "currency": product.currency,
    "current_price": current_price,
    "price_difference": price_diff,
    "deviation_string": f"{deviation:+.2f}%",
    "deviation_text": deviation_text,
    "predicted_price": predicted_price,
    "change_index": change_index
}
```

Рисунок 5.4 — Структура словника, що надає детальну інформацію про товар

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		41

`get_basic_subscribed_products_response` — функція, яка отримує всі продукти, на які є підписаний користувач. Вона здійснює запит до таблиці `products` з приєднанням до таблиці підписок `users_products_subscriptions` за умови відповідності унікального ідентифікатора користувача. Для кожного продукту з вибірки виконується обчислення статистики цін через функцію `get_price_stats`. Після чого формується список словників, де для кожного продукту містяться: його ідентифікатор, обрізане ім'я, поточна ціна, валюта, різниця в ціні в абсолютному значенні і у відсотках, а також статус зміни. Повертається сформований список базових даних про товари, на які підписаний користувач.

`get_user_subscriptions_count` — повертає загальну кількість продуктів, на які підписаний користувач за його унікальним ідентифікатором.

`get_marketplace_configurations` — функція, що виконує запит до таблиці `marketplace_configurations`, щоб отримати записи конфігурацій маркетплейсів для подальшого їх використання.

`update_marketplace_configuration` — функція, що оновлює запис конфігурації маркетплейсу, який зберігається в базі даних. Пошук та оновлення конфігурації відбувається за її унікальною назвою.

`add_marketplace_configuration` — функція, що додає новий запис конфігурації маркетплейсу до бази даних.

`delete_marketplace_configuration` — видаляє запис конфігурації маркетплейсу з бази даних за допомогою його унікального імені.

Наступним та не менш важливим є файл `auth.py`, який інкапсулює в собі функції роботи з автентифікацією та авторизацією користувача. На вебсервері моніторингу цін реалізовано токен-орієнтований механізм авторизації, щоб забезпечити захист приватних даних користувачів. Основна ідея полягає в тому, що після успішного входу користувач отримує токен, який має бути переданий з кожним наступним запитом, що потребує авторизації. У `auth.py` містяться функції для створення та перевірки таких токенів.

`hash_password` — приймає пароль і повертає його захешовану версію у вигляді рядка.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		42

`verify_password` — приймає пароль у звичайному та захешованому виглядах та перевіряє, що після хешування звичайного паролю значення збігаються.

`create_access_token` — створює JWT токен доступу на основі вхідних даних. Токен містить поле з датою закінчення дії. Для підпису використовується секретний ключ та алгоритм HS256.

`decode_access_token` — функція, що створена для валідації JWT токена. Спочатку вона перевіряє переданий токен за допомогою заданого секретного ключа та алгоритму HS256.

Наступним важливим компонентом є файл, що містить основні налаштування вебсервера, написаний з використанням FastAPI. У цьому файлі реалізовано API для взаємодії з базою даних, обробки запитів від користувачів та забезпечення авторизації та аутентифікації. Код організований безпосередньо так, щоб користувачі могли зареєструватися, увійти до системи та отримати доступ до захищених маршрутів після успішної автентифікації через токен.

`get_current_user` — функція отримує поточного користувача через cookie, після чого перевіряє дійсність токена та витягує роль і ім'я користувача з токена для подальшої авторизації. Потім шукає користувача в базі даних і повертає його, якщо все успішно.

`admin_required` — перевіряє, чи є у поточного користувача роль адміністратора. Якщо ні — повертає помилку.

`register_user` — відповідає за реєстрацію нового користувача, виконується коли користувач відправляє POST запит по маршруту `/register` до вебсерверу.

`login_user` — функція, що приймає дані для автентифікації користувача. Виконується коли користувач надсилає POST запит по маршруту `/login`. Після успішної валідації даних, що були надіслані, функція також встановлює cookie з JWT токеном, щоб в подальшому всі запити йшли вже від користувача, що був автентифікований. Приклад відповіді, що отримує користувач з JWT токеном після виконання цієї функції наведено на рисунку 5.5.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		43





маркетплейсі та валюта товару. Приклад масиву підписок для конкретного користувача наведено на рисунку 5.8.

```
"items": [  
  {  
    "marketplace_key": "rozetka",  
    "id": 6,  
    "url": "https://rozetka.com.ua/ua/apple-mpq03rk-a/p357707031/",  
    "currency": "€",  
    "name": "Планшет Apple iPad 10.9\ 2024 Wi-Fi 64GB Silver (MCM74NF/A)"  
  },  
  {  
    "marketplace_key": "rozetka",  
    "id": 7,  
    "url": "https://rozetka.com.ua/ua/465110714/p465110714/",  
    "currency": "€",  
    "name": "Планшет Xiaomi Redmi Pad Pro 8/256 GB Graphite Gray (VHU4750EU) Wi-Fi"  
  },  
  {  
    "marketplace_key": "rozetka",  
    "id": 10,  
    "url": "https://hard.rozetka.com.ua/ua/fractal-design-fd-c-ter1n-02/p409653465/",  
    "currency": "€",  
    "name": "Копиус Fractal Design Terra Silver (FD-C-TER1N-02)"  
  }  
]
```

Рисунок 5.8 — Масив підписок, що повертає функція `list_user_subscriptions`

Також потрібно зазначити, що якщо функція, що приймає JWT токен користувача не зможе підтвердити його актуальність або правильність, то запит поверне помилку `Unauthorized` та тіло запиту де вказується причина. Приклад тіла запиту, що повертається коли JWT токен не пройшов перевірку наведений на рисунку 5.9.

```
{  
  "detail": "Invalid or expired token"  
}
```

Рисунок 5.9 — Приклад тіла відповіді помилки `Unauthorized`

`subscribe_user` — функція, що приймає JWT токен та унікальний ідентифікатор товару, після валідації токена відбувається підписка користувача на товар та відповідний запис створюється у базі даних. Виконується ця функція коли користувач відправляє POST запит на маршрут `‘/subscriptions’`.

`unsubscribe_user` — видаляє товар, унікальний ідентифікатор якого був переданий в тілі запиту. Функція виконується коли користувач відправляє DELETE запит по маршруту `‘/subscriptions’` до вебсерверу.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		46

`get_price_history` — функція, яка виконується після отримання GET запиту по маршруту `‘/products/{product_id}/price_history’`. Вона фактично є допоміжною функцією, яку викликає вебінтерфейс для того, щоб побудувати графік динаміки ціни конкретного товару. Ця функція повертає масив елементів, кожний з яких містить в собі дату та ціну товару на момент додавання запису до бази даних сервісом парсингу. Приклад масиву даних, що повертає функція `get_price_history` наведений на рисунку 5.10.

```
[
  {
    "date": "2025-05-09T11:47:12.100880",
    "price": 10729.0
  },
  {
    "date": "2025-05-11T15:07:35",
    "price": 10729.0
  },
  {
    "date": "2025-05-12T09:42:55",
    "price": 10729.0
  },
]
```

Рисунок 5.10 — Масив елементів, що повертає функція `get_price_history`

`get_marketplace_configurations` — функція, що повертає об’єкти конфігурацій маркетплейсів збережених в базі даних. Сама конфігурація маркетплейсу це об’єкт, що складається з імені маркетплейсу, яке грає роль його унікального ідентифікатора, списку доменів цього маркетплейсу, які використовуються, щоб знайти відповідний маркетплейс, коли користувач вводить його у форму вебінтерфейсу, а також валідувати, що посилання на інтернет магазин не є скомпрометованим, та списку з елементів, що містять в собі метадані HTML елементів, що задані в цих HTML елементах. З них витягується конкретні дані по типу ціни товару, імені товару та валюти в якій представлений товар для того, щоб в подальшому зберігати його в базі даних. Приклад конфігурації, яка вже присутня в інформаційній системі та успішно виконує поставлену задчу, наведено на рисунку 5.11.

```

{
  "name": "Rozetka",
  "fields": [
    {
      "name": "title",
      "html_div_class": "product-main-content",
      "html_element_in_div_type": "h1",
      "html_element_in_div_class": [
        "title__font"
      ]
    },
    {
      "name": "price",
      "html_div_class": "product-price_wrap",
      "html_element_in_div_type": "p",
      "html_element_in_div_class": [
        "product-price_big product-price_big-color-red",
        "product-price_big"
      ]
    }
  ],
  "marketplace_url": [
    "https://hard.rozetka.com.ua",
    "https://rozetka.com.ua",
    "https://bt.rozetka.com.ua/"
  ]
},

```

Рисунок 5.11 — Конфігурація маркетплейсу Rozetka для парсингу

Також були реалізовані функції для редагування, додавання та видалення таких конфігурацій з бази даних відповідно.

Також в інформаційній системі присутній модуль парсингу. Він пропонує API для подальшого використання на вебсервері, а також може працювати окремо та незалежно від вебсерверу, що досягається через те, що модуль запущений як окремий процес через планувальник. Сам модуль складається з наступних файлів: `parse_products`, `price_prediction`, `product`, `marketplace_parser`, `configuration_loader` та `scheduler`.

Файл `marketplace_parser` містить в собі реалізований інтерфейс взаємодії з інтернет магазинами, основною метою якого є впровадження основних функцій для легкого парсингу товарів з будь-яких інтернет магазинів.

У файлі реалізований клас `MarketplaceParser`, що містить в собі єдину доступну для користувача функцію `parse_product_by_url`. Принцип роботи цієї функції полягає в тому, що спочатку шукається відповідна конфігурація інтернет магазину за допомогою посилання, що передали в параметрах функції. Після успішного знаходження конфігурації інтернет магазину, створюються заголовки запиту та сам запит, після чого він відправляється за посиланням. Після отримання відповіді генерується дерево HTML елементів та завантажуються поля з об'єкту

					ІС12.090БАК.005 ПЗ	Арк.
						48
Зм.	Лист	№ докум.	Підпис	Дата		

конфігурації, які необхідно знайти у дереві елементів. Після цього починається пошук полів у дереві елементів, якщо відповідних полів не знайшлось, то функція повертає помилку та процес парсингу товару припиняється. Якщо ж всі необхідні поля були знайдені, то функція повертає об'єкт класу Product, який описується у файлі product та містить наступні поля: назва інтернет магазину з якого була зібрана інформація, назва товару, ціна товару, валюта товару та часова мітка, коли була зібрана інформація.

Функціонал класу MarketplaceParser використовується у файлі parse\_products. На кресленику IC12.090БАК.005 Д4 наведена діаграма процесів модулю парсингу та прогнозування, які відбуваються в модуль парсингу та прогнозування. Перший йде процес парсингу товару, де видно, що спочатку функція main в цьому файлі звертається до бази даних, щоб отримати дані, які необхідно передати методу parse\_product\_by\_url, після успішного отримання даних та обробки цих даних зазначеним методом, якщо знайшлись необхідні конфігурації, функція додає їх до бази даних. Все це робиться за допомогою пакетної обробки даних, що дозволяє обробляти велику кількість даних та не виходити за межі доступної оперативної пам'яті на обладнанні, де виконується програма.

На кресленику IC12.090БАК.005 Д4 також описаний процес прогнозування цін на товари. Після успішного отримання даних із бази даних аналізуються та прибираються викиди в даних, після чого проводиться тренування та прогнозування за допомогою методів машинного навчання, зокрема лінійної регресії, після чого прогнозовані дані записуються у відповідну таблицю в базі даних для того, щоб інформаційна система змогла відобразити приблизний прогноз на ціну товару.

Файл scheduler використовується для взаємодії із планувальником операційної системи, функція main в цьому файлі налаштована, щоб виконувати функції збору та прогнозування даних з файлів price\_prediction та parse\_products. За допомогою цього файлу створюється процес, що виконує функцію main кожного дня у зазначений користувачем час.

					IC12.090БАК.005 ПЗ	Арк.
						49
Зм.	Лист	№ докум.	Підпис	Дата		

Функціонал цього модулю відіграє ключову роль для інформаційної системи, адже дозволяє автоматизувати процес збору та обробки інформації.

## 5.6 Інтерфейс системи

Користувацький інтерфейс реалізовано як вебзастосунок, що складається з окремих сторінок, кожна з яких відповідає за певний функціонал. Комунікація з бекендом здійснюється через RESTful API: сторінки надсилають HTTP запити до відповідних маршрутів для отримання або відправлення інформації. Система використовує механізм токенів доступу JWT після автентифікації користувача токен зберігається у локальному сховищі браузера, і для захищених запитів він додається до заголовків запитів як Authorization Bearer або використовуються cookie для підтримки сесії.

Головна сторінка є стартовим екраном системи, що дозволяє користувачу розпочати роботу з моніторингом цін. Вона надає інтерфейс для введення посилання на товар на підтримуваному маркетплейсі та ініціалізації процесу парсингу даних про цей товар. Також на головній сторінці відображається перелік підтримуваних маркетплейсів. У центрі сторінки розміщено заголовок та форму для парсингу товару. Заголовок інтерфейсу "Parse product" відображається великим шрифтом і пояснює користувачу, що потрібно зробити. Під заголовком знаходиться текстове поле для введення URL адреси товару. Це поле має плейсхолдер "Enter product URL", підказуючи формат введених даних. Користувач повинен скопіювати та вставити або ввести вручну посилання на сторінку товару з одного з підтримуваних маркетплейсів. Приклад головної сторінки наведений на рисунку 5.12.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		50

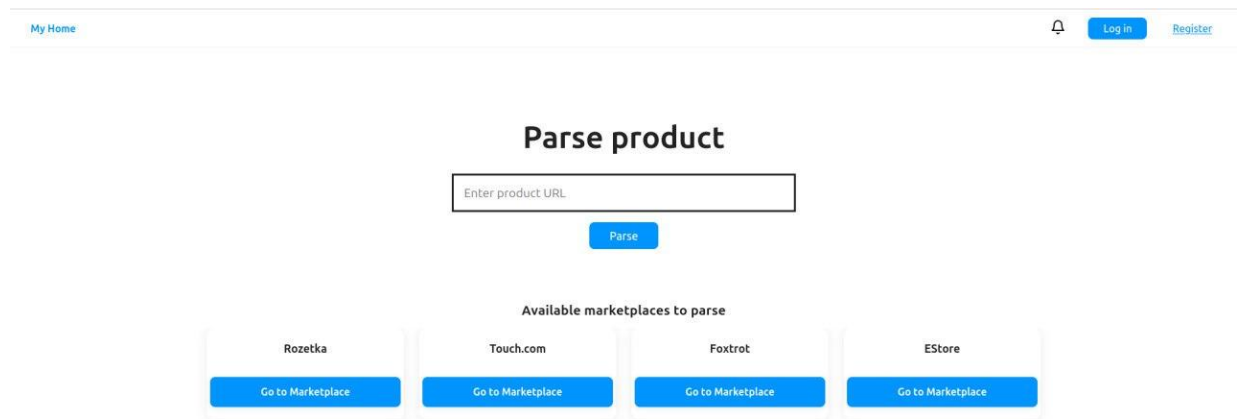


Рисунок 5.12 — Головна сторінка інформаційної системи

Але для того, щоб повноцінно взаємодіяти з інформаційною системою користувачу необхідно зареєструватися в системі та після пройти автентифікацію. Для цього були реалізовані сторінки реєстрації та входу для користувачів. Сторінка реєстрації наведена на рисунку 5.13, вона вимагає від користувача основних реєстраційних даних, таких як ім'я користувача, адреса електронної пошти та пароль. Інтерфейс сторінки виглядає просто та зрозуміло — це центрована форма на нейтральному фоні, яка оформлена у вигляді білого контейнера з тінню.

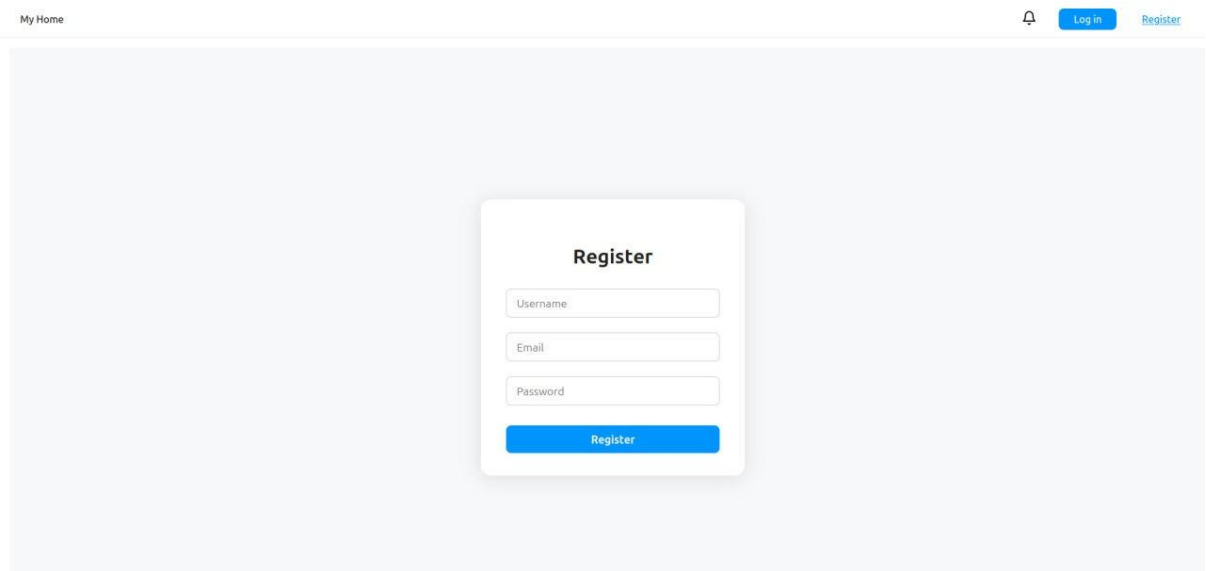


Рисунок 5.13 — Сторінка реєстрації користувача інформаційної системи

Сторінка входу дозволяє користувачу пройти автентифікацію. Фактично вона має дуже схожий інтерфейс до сторінки реєстрації: це також центрована форма оформлена у вигляді білого прямокутника з закругленими кутами та тінню,

що надає їй підкресленого вигляду модального елемента. Має ця форма лише два поля призначених для введення користувачем — це поле імені користувача та його паролю. Обидва поля мають відповідні підказки до заповнення і є обов'язковими при автентифікації. Крім цього на сторінці входу також реалізовано відображення про помилки та посилання для навігації до сторінки реєстрації. Якщо сервер повертає помилку, то текст цієї помилки відображається червоним кольором під кнопкою входу. У разі успішної відповіді сервер повертає JWT токен, який зберігається у локальному сховищі браузера. Успішний вхід до застосунку автоматично переводить користувача до основного інтерфейсу. Слід зазначити, що після входу механізм авторизації працює так, що після входу токен зберігається і використовується неявно: подальші компоненти додатка самостійно перевіряють його наявність у локальному сховищі браузера та при необхідності перенаправляють користувача на сторінку входу. Такий підхід гарантує, що захищені сторінки не відобразатимуться без автентифікації. Реалізовану сторінку входу зображено на рисунку 5.14.

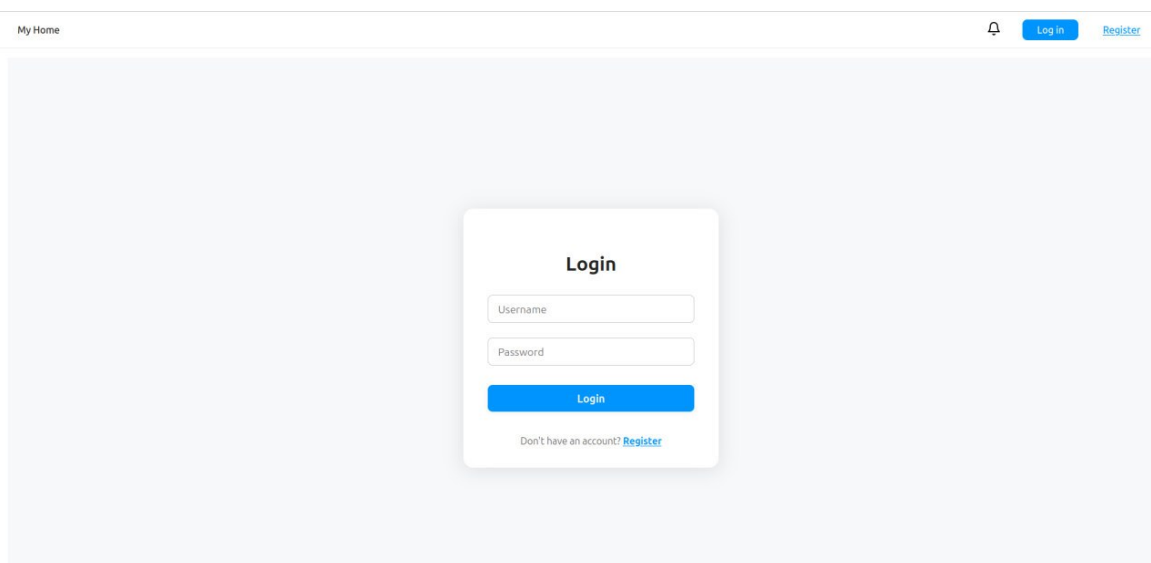


Рисунок 5.14 — Сторінка входу користувача в інформаційну систему

Також в інформаційній системі присутня сторінка перегляду товару, вона є центральним місцем, де користувач отримує аналітичну інформацію. Дана сторінка відображає детальну інформацію про обраний товар, а також історію зміни його ціни. Фактично, це основна сторінка де користувач спостерігає динаміку вартості

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		52



Кожна підписка відображається у вигляді картки, яка включає в себе назву товару, його поточну ціну та на інформацію на скільки відсотків змінилась ціна товару.

Сервіс парсингу оформлений як окремий модуль, що може запускатися незалежно; його архітектура включає підмодуль для отримання HTML модуль парсингу HTML та модуль збереження результатів у базі.

## Висновки до розділу 5

В розділі була представлена програмна реалізація інформаційної системи моніторингу динаміки цін на обраний товар. Загалом програмна реалізація системи вийшла модульною та зручною для внесення змін в архітектуру, адже система складається з модулів, які працюють незалежно один від одного. Спочатку в розділі було описано загальну структуру за якої побудована інформаційна система, потім було представлено функціональну модель, яка забезпечує відстеження цінових змін для стандартних користувачів та адміністраторів. Такий підхід дозволяє зручно відстежувати товари, прогнозувати їхні ціни та отримувати повідомлення про зміни, що робить систему ефективним інструментом для спостережень. Після опису загальної структури та функціональної моделі було детально описано усі можливості вебсерверу, який було розроблено. В цій системі він відповідає за передачу даних між базою даних та вебінтерфейсом, також окремо слід зазначити, що модуль впроваджує можливості авторизації й автентифікації користувачів для того, щоб відокремити та зробити безпечним використання системи. Також в розділі був описаний модуль парсингу та прогнозування ціни, завдяки цій розробці інформаційна система може автоматично збирати ціни та передбачувати зміни у вартості товару. Було детально описано як саме працює інтерфейс, що пропонує функціональність з відстеження товарів на будь-яких сторінках інтернет магазинів. Розроблений вебінтерфейс також був представлений в цьому розділі, він забезпечує інформативний зовнішній вигляд системи, а також дозволяє користувачу краще аналізувати дані за допомогою візуального їх представлення.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		54

## 6 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

### 6.1 Змістовна постановка задачі

Моніторинг динаміки цін на товари є важливою складовою аналізу ринку в умовах сучасної економіки. Для споживачів та бізнесу важливо вчасно виявляти тенденції зміни цін: знижки, акційні коливання, сезонні підвищення чи зниження вартості. Традиційне відстеження цін вручну або за допомогою простих засобів є неефективним, адже ціни можуть змінюватися щоденно і залежать від багатьох факторів. Проблема полягає в тому, що зміна ціни товару — це багатофакторний процес, на який впливають як внутрішні характеристики продукту (категорія, попит, життєвий цикл), так і зовнішні обставини (конкуренція, сезонність, економічні новини, день тижня тощо).

Отже, постає завдання розробити автоматизовану інформаційну систему, яка зможе одночасно опрацьовувати великі обсяги цінових даних, видаляти аномальні значення, враховувати часові закономірності, наприклад, тижневі чи сезонні цикли, та на основі цього формувати прогноз ціни на заданий проміжок часу. Така система повинна в режимі реального часу або з визначеною періодичністю оновлювати прогнози та індикатори зміни ціни, аби користувач міг швидко реагувати на ринкові зміни.

Мета задачі — забезпечити спостереження за динамікою цін вибраного товару та отримання прогнозних оцінок його вартості в майбутньому. На виході система повинна для кожного товару надавати прогноз ціни на визначений проміжок часу вперед та індикатор відсоткової зміни ціни.

Отже, користувач інформаційної системи отримає відповіді на ключові питання: чи зростає ціна на товар, чи варто очікувати зниження, якою може бути ціна через певний час. Система має працювати автоматично, враховуючи всі доступні дані та оновлюючи прогнози при надходженні нових значень, забезпечуючи тим самим актуальність і точність інформації.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		55

## 6.2 Математична постановка задачі

Нехай  $P = \{p_1, p_2, \dots, p_n\}$  — множина ідентифікаторів товарів, для яких здійснюється моніторинг цін. Для кожного товару  $p_i$  існує історичний ряд даних про ціну  $Y_i = \{(t_{i1}y_{i1}), (t_{i2}y_{i2}), \dots, (t_{ij}y_{ij})\}$ , де  $t_{ij}$  — мітка часу,  $y_{ij}$  — зафіксована ціна.

Перед проведенням аналізу часового ряду необхідно усунути пропущені значення цін та аномальні вибірки, які можуть спотворити результати моделювання. Будемо вважати викидами такі значення ціни, що значно відхиляються від середнього. Формально обчислюється середнє значення ціни  $\mu_i$  та середнє квадратичне відхилення  $\sigma_i$  для історичних даних товару  $p_i$  за допомогою формул 6.1 та 6.2.

$$\mu_i = \frac{1}{m_i} \sum_{j=1}^{m_i} y_{ij}, \quad (6.1)$$

$$\sigma_i = \sqrt{\frac{1}{m_i} \sum_{j=1}^{m_i} (y_{ij} - \mu_i)^2}. \quad (6.2)$$

Відповідно до формули 6.3, на основі правила трьох сигм формуємо підмножину очищених даних  $H'_i \subseteq Y_i$  вилучаючи цінові викиди:

$$H'_i = \{(t_{ij}y_{ij}) \in Y_i \mid \mu_i - 3\sigma_i \leq y_{ij} \leq \mu_i + 3\sigma_i\}. \quad (6.3)$$

Отриманий часовий ряд  $H'_i$  відсортовано за часом та може використовуватися для побудови моделі. Кожна точка ряду характеризується кількома параметрами, які враховуються моделлю прогнозування. В даному проєкті для кожної дати  $t_{ij}$  обрано такі ознаки:

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		56

–  $x_{ij}^{(1)}$  — порядковий номер дня відносно початку спостережень.

Обчислюється як кількість днів між датою  $t_{ij}$  та найпершою датою  $t_{i1}$  в історичних даних товару;

–  $x_{ij}^{(2)}$  — день тижня для дати  $t_{ij}$ , що враховує можливу сезонність цін;

–  $x_{ij}^{(3)}$  — авторегресійна ознака, що відображає залежність поточної ціни від попередньої;

–  $x_{ij}^{(4)}$  — значення ковзного середнього за період часу до поточного моменту.

Представляє локальний середній рівень ціни за період.

Відповідно, очищені та збагачені ознаками дані для товару  $p_i$  можна представити у вигляді набору пар  $\{(X_{ij}, y_{ij})\}$ , де  $X_{ij} = (x_{ij}^{(1)}, x_{ij}^{(2)}, x_{ij}^{(3)}, x_{ij}^{(4)})$  — вектор ознак, а  $y_{ij}$  — фактична ціна.

Після цього можна переходити до побудови моделі  $f_i(X)$  для кожного товару  $p_i \in P$ , яка набуваючи на вході значення ознак  $X_{ij}$ , пов'язаних з моментом часу  $t_{ij}$ , повертала би наближене значення ціни  $\hat{y}_{ij}$ . В рамках даної задачі як модель  $f_i$  обрано лінійну регресію, тобто лінійну функціональну залежність між ознаками та значенням ціни. Формально, прогнозне значення ціни визначається як скалярний добуток вектора ознак на вектор параметрів моделі  $\beta_i = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4)$  за формулою 6.4:

$$\hat{y}_{ij} = \beta_0 + \beta_1 x_{ij}^{(1)} + \beta_2 x_{ij}^{(2)} + \beta_3 x_{ij}^{(3)} + \beta_4 x_{ij}^{(4)}. \quad (6.4)$$

Коефіцієнти  $\beta_k$  підбираються в процесі навчання моделі на історичних даних так, щоб мінімізувати різницю між прогнозними та фактичними значеннями цін.

Результатом роботи моделі  $f_i$  є прогнозне значення ціни товару  $p_i$  на заданий проміжок часу. На основі прогнозу також обчислюється індекс зміни ціни — відсоткове відхилення прогнозованої ціни від поточної ціни. Індекс зміни ціни необхідний для того, щоб надати відповідь щодо тренду зростання або спадання ціни.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		57

### 6.3 Обґрунтування методів розв'язання

Для вирішення поставленої задачі прогнозування динаміки цін було розглянуто кілька підходів. Кожен з них відрізняється за складністю реалізації, вимогами до обчислювальних ресурсів, точністю прогнозу та здатністю адаптуватися до різних типів товарів і характеру їхніх цінових коливань. Нижче наведено три можливі підходи, їх короткий опис, переваги та недоліки, а також аналіз доцільності їх застосування в межах даного проєкту.

#### 6.3.1 Наївне прогнозування

Найбільш простий і прямолінійний спосіб передбачити майбутню ціну — це наївне прогнозування, коли в якості прогнозу приймається останнє відоме значення або середнє за останній період. По суті, цей метод взагалі не намагається вловити закономірності зміни ціни, а виходить із припущення, що у короткостроковій перспективі ціна істотно не зміниться.

Такий підхід фактично задає базовий рівень для порівняння складніших моделей. Він не враховує жодних факторів, окрім останніх значень ряду, але в деяких може давати прийнятний результат. Наприклад, якщо товар має стабільний попит і відсутні виражені тренди, тоді такий прогноз може бути недалекою від істини.

Переваги наївного підходу:

- надзвичайно проста реалізація;
- не потребує збору додаткової інформації чи налаштування параметрів;
- мінімальні вимоги до ресурсів, підходить для дуже великої кількості товарів.

Недоліки наївного підходу:

- ігнорує будь-які тренди та циклічності;
- часто дає велику похибку для товарів, ціна яких активно змінюється;
- не сигналізує про напрямок зміни.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		58

Наївний метод може слугувати базовим для оцінки, але сам по собі не задовольняє вимоги до точності та інформативності прогнозу в більшості реальних сценаріїв.

### 6.3.2 Класичні моделі часових рядів

Більш просунутий шлях прогнозування цін — використання класичних статистичних моделей аналізу часових рядів. До них належать моделі типу ARIMA — AutoRegressive Integrated Moving Average, експоненціальне згладжування — модель Хольта-Вінтерса та інші. Модель ARIMA дозволяє врахувати автокореляцію ряду, тренд та сезонність.

Переваги статистичних моделей:

– добре працюють коли прогноз залежить тільки від самого ряду цін, без зовнішніх факторів.

Недоліки статистичних моделей:

- моделі чутливі різких змін тренду, можуть викликати великі похибки;
- вимогливі до обчислювальних ресурсів при масштабуванні.

Статистичні методи є потужним інструментом, але з огляду на специфіку задачі їх застосування у чистому вигляді менш зручне.

### 6.3.3 Регресійний аналіз із ознаками

Підхід полягає у використанні методів машинного навчання, зокрема регресійного аналізу, для прогнозування цін на основі сформованих ознак. До методів, які можна застосувати, відносяться лінійні моделі, дерево рішень та ансамблі, а також більш складні алгоритми на кшталт нейронних мереж. У дипломному проєкті вибір зроблено на користь відносно простої моделі — лінійної регресії, що пояснюється достатністю її можливостей для виявлення основних трендів та сезонності, а також швидкістю й стійкістю навчання.

Переваги регресійного аналізу:

					ІС12.090БАК.005 ПЗ	Арк.
						59
Зм.	Лист	№ докум.	Підпис	Дата		

- легко інтегрувати різнорідні ознаки, що описують як внутрішню структуру ряду, так і зовнішні впливи;
- використання регуляризації запобігає перенавчанню на коротких рядах і робить модель стійкішою до шуму;
- існують зручні метрики та методи перехресної перевірки, що дозволяють об'єктивно виміряти, наскільки добре модель прогнозує на історичних даних, і підібрати параметри.

Недоліки регресійного аналізу:

- модель вважає залежність лінійною, що може не врахувати складні нелінійні ефекти. Більш гнучкі моделі могли б вловити складніші випадки, але потребують більше даних і ресурсів;
- необхідно мати достатню історію даних для навчання. Якщо товар новий і даних мало, лінійна регресія не зможе дати надійного прогнозу.

З огляду на сукупність переваг і недоліків, а також специфіку задачі, що вимагає автоматичного, швидкого та достатньо точного прогнозування цін для багатьох товарів, було обґрунтовано доцільність використання саме підходу машинного навчання на основі регресії з ознаками часу.

#### 6.4 Опис методу розв'язання

В якості основного методу автоматизованого моніторингу та прогнозування динаміки цін у даному проєкті реалізовано підхід машинного навчання з використанням лінійної регресії та декількох інформативних ознак, побудованих на основі історичного ряду цін. Алгоритм розбитий на послідовність етапів: завантаження даних, їх очищення, побудова ознак, навчання та оцінювання моделі, виконання прогнозу, збереження результатів.

На вхід алгоритм отримує історичні дані цін для одного або кількох товарів. Кожна часова точка характеризується датою і зафіксованою ціною. Після завантаження даних для конкретного товару вони очищаються: видаляються записи без ціни та відсіюються аномально великі або малі значення. Це дозволяє

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		60

прибрати помилкові дані парсингу або разові екстремальні акційні ціни, які не відображають реальної довгострокової динаміки. Далі здійснюється побудова ознак, а після їх обчислення відбувається навчання та оцінка моделі.

Алгоритм побудований так, щоб його можна було легко виконувати для великої кількості товарів автоматично. Для оптимізації процесу використовується пакетна обробка: товари обробляються групами певного розміру. Це необхідно, щоб уникнути перевантаження пам'яті при вибірці даних з бази, а також раціонально використовувати ресурси при вставці прогнозів назад в базу.

### Висновки до розділу 6

У даному розділі було сформовану мету задачі прогнозування ціни конкретного товару як задачу аналізу часових рядів із використанням методів машинного навчання. Після вирішення такої задачі користувач інформаційної системи зможе отримати відповідь на головне питання — чи буде зміна ціни товару в майбутньому. В розділі було представлено математичну постановку задачі: визначено множину вхідних даних, сформована проблема очищення даних від викидів, введено ознаки для сезонності та описано модель лінійної регресії, що буде використовуватися у відповідному модулі інформаційної системи.

Крім того було проведено огляд можливих підходів до розв'язання, серед яких розглянуто наївний метод, класичні статистичні моделі та підхід машинного навчання. Обґрунтовано вибір останнього підходу як основного, зважаючи на потребу в швидкому та відносно надійному рішенні для прогнозування цін багатьох товарів одночасно. Описано структуру розробленого алгоритму прогнозування в модулі інформаційної системи, що додатково прискорений за допомогою пакетної обробки даних.

Підсумовуючи, обраний в цьому проекті підхід забезпечує автоматизацію відстеження динаміки цін і надання прогнозної інформації, що є важливою складовою інформаційної системи моніторингу цін.

					ІС12.090БАК.005 ПЗ	Арк.
						61
Зм.	Лист	№ докум.	Підпис	Дата		

## 7 ТЕСТУВАННЯ СИСТЕМИ

### 7.1 Мета випробувань

Метою проведення випробувань є перевірка працездатності розробленої інформаційної системи моніторингу динаміки цін та відповідності її функціоналу визначеним вимогам. Необхідно впевнитися, що всі модулі системи правильно взаємодіють між собою і забезпечують реалізацію поставлених задач дипломної роботи. Ці випробування мають продемонструвати, що система працює надійно та не має в собі якихось критичних збоїв, вона правильно обробляє дані та своєчасно реагує на дії користувача згідно з визначеними сценаріями використання.

### 7.2 Загальні положення

Тестування проводилося в локальному середовищі розробника. Усі модулі системи було розгорнуто на одному комп'ютері в одній мережі, що дозволило імітувати роботу сервісів у взаємодії один з одним. Для запуску компонентів використовувалося локальне оточення: вебсервер, який працював за адресою localhost з визначеним портом; вебінтерфейс, який запускався у режимі розробки в браузері; база даних, що була розгорнута локально та модуль парсингу та прогнозування, що виконувався як окремий фоновий процес в системі. Таке тестове середовище відтворює реальну архітектуру системи, але без розгортання на зовнішньому сервері, що спрощує налагодження.

В ході випробувань було проведено одразу функціональне та інтеграційне тестування. Функціональне тестування полягало у перевірці основних функцій системи через користувацький інтерфейс та запити до вебсерверу. Інтеграційне тестування було спрямоване на перевірку того, як взаємодіють модулі системи, наприклад, відправлення запиту користувача на вебінтерфейсі до вебсерверу, збереження та отримання даних з бази та перевірка фонові роботи модулів парсингу і прогнозування. Перевірялися як типові сценарії використання, так і граничні випадки, наприклад, спроба введення некоректних даних чи відсутність

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		62

підключення до зовнішнього ресурсу для модуля парсингу, щоб упевнитися в стійкості системи.

### 7.3 Результати випробувань

Було підготовлено кілька ключових сценаріїв тестування, які охоплюють основні функціональні можливості системи. Нижче наведено результати виконання цих тестових сценаріїв, кожен з яких містить опис ситуації, очікуваний результат та фактичний результат після проведення випробування.

Сценарій 1. Додавання нового товару. Користувач через вебінтерфейс додає новий товар для моніторингу, вказуючи необхідні дані.

Очікуваний результат. Система зберігає інформацію про новий товар у базі даних і відображає його у списку підписок в інтерфейсі користувача. Користувач має бути перенаправлений на сторінку товару, що відстежується.

Спочатку було протестовано функціональність яка передбачає, що система не буде обробляти посилання на товари з інтернет магазинів, яких не підтримуються системою. Після вводу в форму посилання на подібний інтернет магазин, що не підтримує система, вебінтерфейс повідомив користувачу, що не зможе провести моніторинг ціни на товар, приклад повідомлення можна спостерігати на рисунку 7.1.

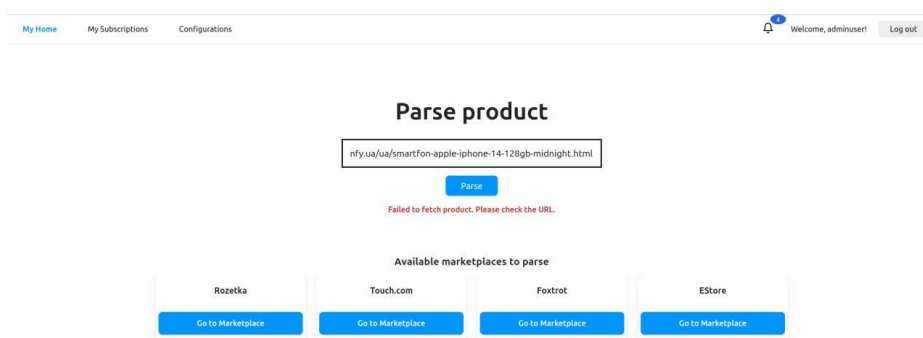


Рисунок 7.1 — Приклад повідомлення користувачу, що посилання на товар не підтримується інформаційною системою

										Арк.
										63
Зм.	Лист	№ докум.	Підпис	Дата						

Тепер у форму було введено посилання товару в інтернет магазині, який підтримує система. Після натискання кнопки “Parse” в базу даних було додано відповідний запис та користувача перенаправлено на сторінку товару, який віно запитував. Результат додавання нового запису можна побачити на рисунку 7.2.

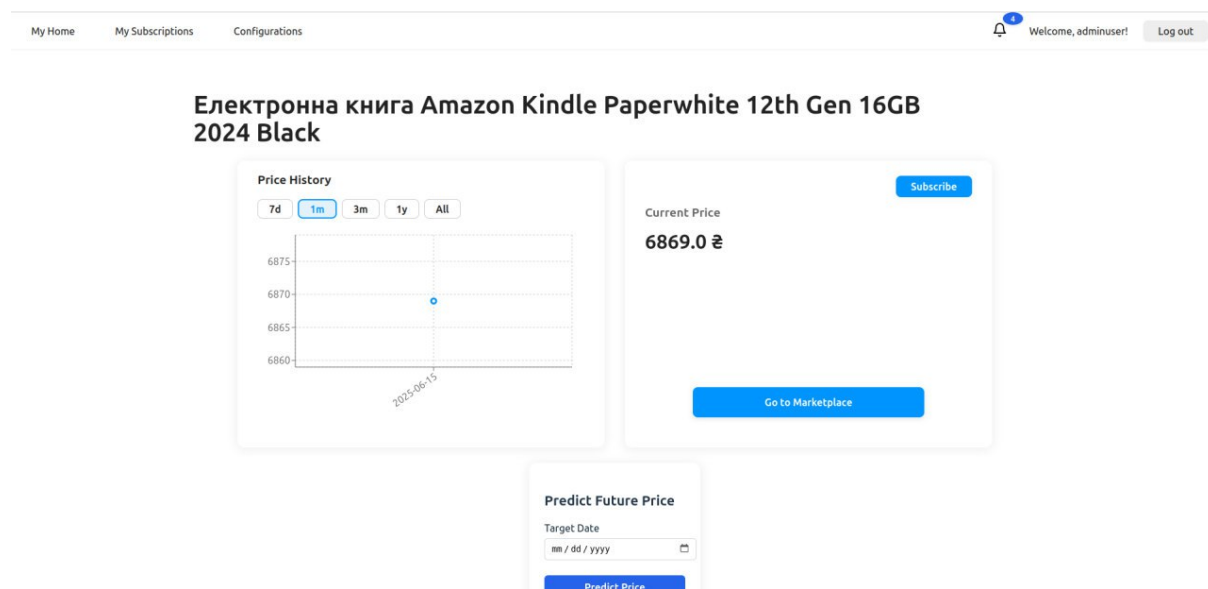


Рисунок 7.2 — Приклад сторінки щойно доданого товару

Фактичний результат. Після виконання дії товар дійсно з’явився у таблиці на сторінці, у базі даних в таблиці products створено відповідний запис. Система вивела перенаправила користувача на сторінку товару, що повністю відповідає очікуваному результату; відхилень не виявлено.

Сценарій 2. Оновлення ціни через модуль парсингу та прогнозування. Фоновий модуль парсингу запускається і отримує актуальну ціну доданого товару з інтернет магазину.

Очікуваний результат. Модуль парсингу автоматично завантажує HTML сторінку інтернет магазину, отримує поточну ціну товару, прогнозує наступну ціну та записує її у базу даних. У системному журналі повинно з’явитися повідомлення про успішне оновлення ціни. У вебінтерфейсі оновлена ціна відображається на графіку зміни ціни конкретного товару.

За допомогою інструменту Docker Compose було окремо запущено модуль парсингу та прогнозування даних, цей модуль відпрацював правильно, про що

					IC12.090BAK.005 ПЗ	Арк.
						64
Зм.	Лист	№ докум.	Підпис	Дата		



```
Body Cookies Headers (4) Test Results 401 Unauthorized
Pretty Raw Preview Visualize JSON
1 {
2   "detail": "Incorrect username or password"
3 }
```

Рисунок 7.4 — Відповідь вебсерверу, коли користувач ввів невірні дані

Якщо дані для входу були введені правильно, то генерується відповідний JWT токен, який приходить із відповіддю вебсерверу, що видно на рисунку 7.5.

```
Body Cookies (1) Headers (5) Test Results 200 OK 190 ms 552 B
Pretty Raw Preview Visualize JSON
1 {
2   "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1IjoiInR5cCI6IkpXVCJ9.eyJ1IjoiInR5cCI6IkpXVCJ9.eyJ1IjoiInR5cCI6IkpXVCJ9",
3   "token_type": "bearer"
4 }
```

Рисунок 7.5 — Відповідь вебсерверу з вкладеним JWT токеном

При спробі звернення до вебсерверу для отримання товарів, що відстежує користувач, без попередньої автентифікації повертається помилка авторизації Unauthorized, яка наведена на рисунку 7.6.

```
Body Cookies (1) Headers (5) Test Results 401 Unauthorized 4
Pretty Raw Preview Visualize JSON
1 {
2   "detail": "Not authenticated"
3 }
```

Рисунок 7.6 — Відповідь вебсерверу при невдалій спробі авторизації

Фактичний результат. Тестування показало, що механізм автентифікації працює правильно. У разі введення коректних облікових даних система здійснює вхід — користувач перенаправляється до головної сторінки, де йому стає доступна персональна сторінка товарів, які він відстежує. При введенні неправильного пароля вхід було заблоковано, на екрані з’явилося повідомлення “Неправильні дані для входу”, як і очікувалося. Запити до API без дійсного токена автентифікації були відхилені: сервер повернув помилку Unauthorized, і вебінтерфейс такі запити перенаправив на сторінку входу. Отже, автентифікація і захист даних реалізовані правильно.

## Висновки до розділу 7

У даному розділі наведено процес і результати тестування розробленої інформаційної системи моніторингу цін. Було проведено декілька результативних тестів щодо функціональних та інтеграційних можливостей системи.

Перші два сценарії допомогли протестувати один з основних процесів інформаційної системи — додавання нового товару до моніторингу та відстеження його ціни, відповідно. Ці тестові сценарії дали змогу перевірити усі наявні компоненти системи на успішне виконання поставлених задач обробки, прогнозування та відображення даних. Третій сценарій допоміг впевнитися, що система має правильно налаштований функціонал з автентифікації та авторизації користувача зі сторони вебсерверу.

Під час проходження зазначених тестових сценаріїв були протестовані усі наявні модулі інформаційної системи моніторингу динаміки цін на обраний товар. Усі тестові сценарії завершилися успішно, що свідчить про правильність архітектурних рішень та якісну реалізацію програмного коду системи. Загалом, можна сказати, що система працює стабільно в локальному середовищі, саме тому має сенс її подальше тестування на сервері під навантаженням, але це виходить за межі задач розробки інформаційної системи.

Варто зазначити, що під час виконання тестових сценаріїв система могла поводити себе не так як від неї очікують, але подальше виправлення деяких помилок в кодовій базі змінило ситуацію, тепер система може вважатися протестованою, адже усі компоненти були покриті тестовими сценаріями. Самі компоненти активно взаємодіяли один з одним під час проведення тестування, саме тому, якщо б один з них вийшов з ладу, інший компонент повідомив би про це та помилку було б виправлено.

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		67

## ВИСНОВКИ

Результатом виконання дипломного проєкту є інформаційна система моніторингу динаміки цін на обраний товар, яка також має можливість прогнозувати ціну товару за допомогою методів лінійної регресії.

У розділі було “Опис предметної області” було визначено мету, а також поставлені задачі розробки інформаційної системи.

У розділі “Аналіз існуючих рішень” було проаналізовано вже розроблені інформаційні системи, які відстежують товари в різних інтернет магазинах. В процесі аналізу стало зрозуміло, що кожна з цих систем має свої недоліки та переваги, але жодна з них на момент проведення аналізу не мала функціоналу із сповіщеннями користувача про зміну цін на товар, а також прогнозування ціни. Розроблена інформаційна система має вирішити ці недоліки.

У розділі “Формування вимог до системи” були описані основні вимоги, які має задовільнити система, та модульна архітектура системи, яка дозволить легко масштабуватись та впроваджувати новий функціонал.

У розділі “Вибір технології розробки” були обґрунтовані технології, що використовуються в проєкті. Вибір, в більшій мірі, пав на технології з якими можна працювати легко та швидко. Саме тому для створення вебсерверу та модулю парсингу та прогнозування була обрана мова програмування Python та вебфреймворк FastAPI, який вона пропонує. Для створення вебінтерфейсу був обраний JavaScript та фреймворк React, що дозволило робити інтерфейс швидко та підтримувати модульність під час розробки. Також всі ці сервіси були реалізовані модульно за допомогою технологій контейнеризації Docker.

У розділі “Розробка інформаційної системи” була представлена програмне реалізація інформаційної системи, яка була змодельована та описана в попередніх розділах. Були представлені відповідні архітектурні рішення, які дозволили підтримувати та розробляти сервіси та модулі незалежно один від одного. В самому розділі можна побачити багато компромісів на які прийшлося йти для того, щоб система працювала правильно, але загалом був створений надійний вебсервер,

					IC12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		68

оптимізований до споживання пам'яті модуль парсингу та прогнозування, а також швидкий та надійний вебінтерфейс користувача.

У розділі “Математичне забезпечення” було обгрунтовано вибір методів для рішення задачі прогнозування. Як виявилось метод прогнозування засобами лінійної регресії загалом відповідає поставленим вимогам до інформаційної системи, але через, можливо, поверхневу реалізацію без точного налаштування, може бути чутливим до різких змін трендів та відповідно показувати інформацію, що не є актуальною.

У розділі “Тестування системи” було проведено тестування основного функціоналу інформаційної системи. Були проведені тести основних модулів: вебсерверу, модулю парсингу та прогнозування, а також вебінтерфейсу. Після виконання тестових сценаріїв система працювала стабільно та поставлені задачі були виконані.

В подальшому розвивати розроблену інформаційну систему можна шляхом впровадження нового функціоналу, наприклад каталог товарів, персоналізовані рекомендації та порівняння цін між інтернет магазинами.

Підсумовуючи, можна сказати, що дипломний проєкт успішно вирішив поставлені задачі, а також було досягнуто поставлену мету проєкту.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Amazon.com. Amazon.com. URL: <https://www.amazon.com/> (дата звернення: 02.05.2025).
2. CamelCamelCamel. camelcamelcamel.com. URL: <https://camelcamelcamel.com/> (дата звернення: 02.05.2025).
3. Djinni | Hire talent or find a job: remotely & on your own. Djinni | Hire talent or find a job: remotely & on your own. URL: <https://djinni.co/> (дата звернення: 05.05.2025).
4. Docker: accelerated container application development. Docker. URL: <https://www.docker.com/> (date of access: 10.05.2025).
5. Docker compose. Docker Documentation. URL: <https://docs.docker.com/compose/> (date of access: 10.05.2025).
6. FastAPI. FastAPI. URL: <https://fastapi.tiangolo.com/> (дата звернення: 10.05.2025).
7. Hotline - порівняти ціни в інтернет-магазинах України. Hotline.ua. URL: <https://hotline.ua/> (дата звернення: 10.05.2025).
8. JavaScript | MDN. MDN Web Docs. URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата звернення: 10.05.2025).
9. PostgreSQL: documentation. PostgreSQL: The world's most advanced open source database. URL: <https://www.postgresql.org/docs/> (дата звернення: 10.05.2025).
10. PostgreSQL. PostgreSQL. URL: <https://www.postgresql.org/> (дата звернення: 10.05.2025).
11. PriceSpy Ltd. PriceSpy UK. URL: <https://pricespy.co.uk/> (дата звернення: 02.05.2025).
12. Price.ua® - порівняння цін інтернет-магазинів в Україні. Price.ua® - порівняння цін інтернет-магазинів в Україні. URL: <https://price.ua/ua> (дата звернення: 02.05.2025).

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		70

13. React – A JavaScript library for building user interfaces. React – A JavaScript library for building user interfaces. URL: <https://legacy.reactjs.org/> (дата звернення: 10.05.2025).

14. The official YAML web site. The Official YAML Web Site. URL: <https://yaml.org/> (дата звернення: 10.05.2025).

15. Welcome to python.org. Python.org. URL: <https://www.python.org/> (дата звернення: 10.05.2025).

					ІС12.090БАК.005 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		71

