

Хмельницький національний університет  
Факультет інформаційних технологій  
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Інформаційна система управління закупівлями в сфері електронної комерції

Назва теми

КВРІСТ 2101029.21.01.02 ПЗ

Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 126 «Інформаційні системи та технології»

Шифр, назва

Освітня програма «Інформаційні системи та технології»

Назва

Виконала: студентка IV курсу, група ІСТ-21-1

  
Підпис

Дарина ГАЛГАН

Ініціали, прізвище

Керівник

  
Підпис, дата

Єлизавета ГНАТЧУК

Ініціали, прізвище

Нормоконтролер

  
Підпис, дата

Тетяна КИСІЛЬ

Ініціали, прізвище

До захисту допускаю:  
Зав. кафедри комп'ютерної  
інженерії та інформаційних  
систем

  
Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

«12» червня 2025 р.

Хмельницький 2025

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 “ 01 2025 р.

## ЗАВДАННЯ

### НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Дарині ГАЛГАН

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інформаційна система управління закупівлями в сфері електронної комерції

Керівник проекту (роботи) Єлизавета ГНАТЧУК д.т.н., професор, професор кафедри

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

Теоретичні основи досліджуваної проблеми

Проектування інформаційної системи управління закупівлями

Програмна реалізація інформаційної системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) \_\_\_\_\_

Архітектура проекту \_\_\_\_\_

Архітектура інформаційної системи \_\_\_\_\_

Вигляд вікон інформаційної системи \_\_\_\_\_

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

6. Консультанти розділів дипломного проекту (роботи)

| Розділ        | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|---------------|---|----------------|------------------|
|               |   | завдання видав | завдання прийняв |
| Нормоконтроль | Тетяна КИСІЛЬ, доцент кафедри КПС         |                |                  |
| Антиплагіат   | Андрій Нічепорук, доцент кафедри КПС      |                |                  |

7. Дата видачі завдання « 10 » 01 2025 р.

**КАЛЕНДАРНИЙ ПЛАН**

| №з/п | Назва етапів (розділів) дипломного проекту (роботи)   | Термін виконання етапів проекту (роботи) | Примітка |
|------|---|--|----------|
| 1    | Вибір напряму дослідження та узгодження тематики кваліфікаційної роботи з керівником                                  | 10.01.2025                               | виконано |
| 2    | Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження | 01.02.2025                               | виконано |
| 3    | Розділ 1. Проаналізовано закупівлі в e-commerce, розглянуто рішення, визначено задачі проекту.                        | 01.03.2025                               | виконано |
| 4    | Розділ 2. Розроблено архітектуру та структуру інформаційної системи.  | 01.04.2025                               | виконано |
| 5    | Розділ 3. Реалізовано та протестовано прототип веб-застосунку для автоматизації закупівель.                           | 29.04.2025                               | виконано |
| 6    | Оформлення пояснювальної записки згідно вимог   | 25.05.2025                               | виконано |
| 7    | Попередній захист ВКР   | 26.05.2025                               | виконано |
| 8    | Захист ВКР на засіданні ЕК  | Червень 2025 року                        |          |

Студент

Підпис

Дарина ГАЛГАН  
Ініціали, прізвище

Керівник роботи

Підпис

Єлизавета ГНАТЧУК  
Ініціали, прізвище

| №<br>р<br>я<br>д<br>к<br>а | Ф<br>о<br>р<br>м<br>а<br>т | Позначення                 | Найменування                          | К<br>і<br>л<br>·<br>л<br>и<br>с<br>т<br>і<br>в | №<br>ек<br>з   | П<br>р<br>и<br>м<br>і<br>т<br>к<br>а |
|----------------------------|----------------------------|----------------------------|---------------------------------------|--|--|--------------------------------------|
|                            |                            |                            | Текстові документи                    |  |  |                                      |
| 1                          |                            | КВРІСТ 2101029.21.01.02 ПЗ | Пояснювальна записка                  | 60   |  |                                      |
|                            |                            |                            | <u>Графічні документи</u>             |  |  |                                      |
| 2                          |                            | КВРІСТ 2101029.21.01.02 Е8 | Архітектура проєкту                   | 1  |  |                                      |
| 3                          |                            | КВРІСТ 2101029.21.01.02 Е8 | Архітектура<br>інформаційної системи  | 1  |  |                                      |
| 4                          |                            | КВРІСТ 2101029.21.01.02 Е8 | Вигляд вікон<br>інформаційної системи | 1  |  |                                      |
| КВРІСТ 2101029.21.01.02 ПЗ |                            |                            |                                       |  |  |                                      |
| Зм                         | Арк                        | № докум                    | Підпис                                | Дата   | Відомість проєкту<br><br>Літера    Аркуш    Аркушів<br>у            1            60<br><br>ХНУ, ІСТ-21-1 |                                      |
| Розробив                   |                            | Галган                     |                                       |  |  |                                      |
| Перевір.                   |                            | Гнатчук                    |                                       | 12.06.20                                       |  |                                      |
| Н. контр.                  |                            | Кисіль                     |                                       | 12.06.20                                       |  |                                      |
| Затв.                      |                            | Павлова                    |                                       | 14.06.20                                       |  |                                      |

## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Інформаційна система управління закупівлями в сфері електронної комерції».

Автор роботи: Дарина ГАЛГАН

Керівник роботи: Гнатчук Єлизавета Геннадіївна

Пояснювальна записка: 60 с., 5 рис., 4 дод., 49 джерел

Графічна частина: 3 креслення

Метою даного дослідження є розробка та впровадження інформаційної системи управління закупівлями в сфері електронної комерції, що забезпечує автоматизовану взаємодію між внутрішніми процесами закупівель і платформою продажу WooCommerce. Система поєднує програмні компоненти (веб-додаток на основі Django, REST API, база даних MySQL) та, за потреби, може бути інтегрована з фізичними сенсорними пристроями для створення кіберфізичного контуру автоматичного моніторингу залишків товарів.

Об'єктом дослідження є інформаційна система управління закупівлями, орієнтована на використання в e-commerce середовищі.

Предметом дослідження є архітектура, функціональне наповнення та реалізація механізмів інтеграції між модулями управління закупівлями та платформами електронної комерції, зокрема через WooCommerce API.

Під час дослідження застосовано методи системного аналізу для оцінки існуючих програмних рішень, методи структурного та об'єктно-орієнтованого проектування для розробки архітектури системи, а також методи прототипування та модульного тестування для створення та перевірки працездатності програмної реалізації.



Підпис студента

30.05.2025

Дата

## ЗМІСТ

|  |    |
|--|----|
| <b>ВСТУП</b> .....   | 3  |
| <b>1 ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ</b> .....                  | 4  |
| 1.1 Аналіз предметної області і виявлення наявних проблем і завдань..... | 4  |
| 1.2 Порівняльний аналіз переваг та недоліків існуючих рішень .....       | 7  |
| 1.3 Підходи до вирішення задачі за темою дослідження.....                | 13 |
| 1.4 Висновки. Постановка задачі.....                                     | 17 |
| <b>2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ЗАКУПІВЛЯМИ</b> ..... | 20 |
| 2.1 Вибір архітектури системи.....                                       | 20 |
| 2.2 Інтеграція з e-commerce платформою та іншими бізнес-процесами.....   | 31 |
| 2.3 Висновки до другого розділу.....                                     | 36 |
| <b>3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ</b> .....                | 37 |
| 3.1 Розробка прототипу веб-застосунку .....                              | 37 |
| 3.2 Реалізація основних функціональних модулів.....                      | 40 |
| 3.3 Тестування системи на бізнес-сценаріях.....                          | 48 |
| 3.4 Висновки до третього розділу.....                                    | 55 |
| <b>ВИСНОВКИ</b> .....  | 57 |
| <b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ</b> .....                                     | 58 |
| <b>ДОДАТОК А</b> .....   | 63 |
| <b>ДОДАТОК Б</b> .....   | 64 |
| <b>ДОДАТОК В</b> .....   | 65 |
| <b>ДОДАТОК Г</b> .....   | 66 |

|  |             |                 |               |                |
|--|-------------|-----------------|---------------|----------------|
| <b>КВРІСТ 2101029.21.01.02 ПЗ</b>  |             |                 |               |                |
| <b>Зм.</b>   | <b>Арк.</b> | <b>№ докум.</b> | <b>Підпис</b> | <b>Дата</b>    |
| Виконав  |             | Галган Д.Л      |               |                |
| Перевір.   |             | Гнатчук С.Г     |               | 12.06.14       |
| Н.контр.   |             | Кисіль Т.М      |               | 12.06.14       |
| Затвер.  |             | Павлова         |               | 12.06.14       |
| Інформаційна система управління закупівлями в сфері електронної комерції |             |                 |               |                |
| Пояснювальна записка   |             |                 |               |                |
|  |             | <b>Літера</b>   | <b>Аркуш</b>  | <b>Аркушів</b> |
|  |             | у               | 2             | 60             |
| <b>ХНУ, ІСТ-21-1</b>   |             |                 |               |                |

## ВСТУП

Сфера електронної комерції сьогодні переживає стрімкий розвиток, який супроводжується жорсткою конкуренцією, мінливим попитом, високою залежністю від постачальників і потребою швидкого реагування на зміни ринкової ситуації. Для інтернет-магазинів, маркетплейсів та платформ онлайн-торгівлі надзвичайно важливим є ефективне управління закупівлями - процес, який має бути прозорим, структурованим і максимально автоматизованим.

У цифрову епоху все більше значення набувають інформаційні системи, здатні оптимізувати весь цикл закупівель - від аналізу потреб і прогнозування, до обліку поставчань і аналізу витрат. Проте сучасні рішення часто не враховують специфіку e-commerce або виявляються обмеженими за функціоналом, надто дорогими, складними для інтеграції або недостатньо гнучкими. Саме ці виклики формують потребу у створенні нової спеціалізованої інформаційної системи, здатної автоматизувати ключові аспекти управління закупівлями в електронній торгівлі.

Дипломна робота спрямована на вивчення теоретичних засад та практичного досвіду в сфері закупівель у цифровій торгівлі. У фокусі дослідження - розробка концептуальних рішень для побудови сучасної системи управління, визначення її архітектурних особливостей і створення прототипу веб-застосунку, здатного підтримувати основні бізнес-процеси, пов'язані із закупівлями.

У процесі дослідження розглядаються ключові аспекти предметної області, оцінюються наявні проблеми, аналізуються існуючі рішення з точки зору їх ефективності та обмежень, формуються вимоги до майбутньої системи, проектується її логічна структура та реалізується програмна частина.

У дослідженні основна увага приділяється процесу управління закупівлями в e-commerce як цілісному об'єкту, в якому ключову роль відіграють інформаційні інструменти. Предметом виступають саме ті методи і засоби, що забезпечують ефективну автоматизацію цих процесів.

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 3    |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

# 1 ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖУВАНОЇ ПРОБЛЕМИ

## 1.1 Аналіз предметної області і виявлення наявних проблем і завдань

Управління закупівлями є критичною складовою діяльності підприємств електронної комерції. Під закупівлями розуміється процес забезпечення бізнесу необхідними товарами та послугами – від визначення потреб до укладання угод з постачальниками та отримання товарів. В електронній комерції (e-commerce) закупівлі здебільшого здійснюються у цифровому середовищі, що відкриває нові можливості та одночасно створює особливі виклики [1, 2].

Розрізняють декілька моделей електронних закупівель: B2B (business-to-business) - транзакції між бізнесами (закупівля товарів та послуг одними компаніями в інших для власної діяльності чи перепродажу), та B2G (business-to-government) – участь бізнесу в публічних державних закупівлях через електронні системи. Обидві моделі є важливими сегментами сучасного ринку e-commerce і активно застосовуються для реалізації корпоративних закупівель, особливо у великих організаціях та міжнародних корпораціях.

У цифровій економіці існує багато спеціалізованих платформ, що обслуговують B2B та B2G транзакції, де підприємства можуть розміщувати заявки на закупівлю і знаходити постачальників онлайн [3, 4].

За даними Євростату, рівень впровадження електронних закупівель у країнах ЄС постійно зростає – наприклад, у 2023 році близько 86% виробничих підприємств Швеції здійснювали B2B/B2G закупівлі або продажі через інтернет-ресурси.

Це свідчить про те, що цифрові рішення для закупівель стають стандартом бізнес-практики. Процес закупівель в контексті електронної комерції охоплює кілька етапів. Спочатку визначаються потреби бізнесу – які товари і в якій кількості необхідно закупити для підтримки асортименту інтернет-магазину чи виконання замовлень. Наступним кроком є вибір постачальників: компанія може мати затверджених постачальників або шукати нових через електронні каталоги,

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 4    |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

маркетплейси чи тендери [5, 6]. Далі формується заявка на закупівлю (purchase requisition) і після внутрішнього погодження створюється замовлення постачальнику (purchase order). Постачальник підтверджує замовлення і здійснює постачання товарів, які підприємство отримує та перевіряє (етап приймання товарів) [7, 8].

Завершується цикл закупівель обробкою рахунків від постачальника та здійсненням оплати (етап обробки рахунків та оплати). Цей повний цикл часто називають procure-to-pay (від закупівлі до оплати). Інформаційна система управління закупівлями повинна підтримувати всі ці кроки, забезпечуючи прозорість, оперативність і контроль на кожному етапі [9].

Варто зазначити специфічні особливості закупівель саме в сфері електронної комерції. Інтернет-магазини та електронні платформи працюють у середовищі жорсткої конкуренції та динамічного попиту з боку споживачів. Швидкість і гнучкість постачання мають критичне значення: якщо товар відсутній на складі, клієнт швидко перейде до конкурента. Тому управління запасами та поповненням товарів повинно бути максимально ефективним.

Одним із ключових завдань є підтримання оптимального запасу: уникнути як дефіциту товарів (stockout, що призводить до втрати продажів і лояльності клієнтів), так і надлишкових запасів (overstock, що «заморожує» обігові кошти та збільшує витрати на зберігання) [10, 11].

Збалансувати це допомагає точне прогнозування попиту, але в практиці e-commerce прогнозування ускладнене швидкими змінами вподобань споживачів та сезонністю попиту.

Таким чином, неточний прогноз попиту є однією з проблем, що веде або до втрати продажів, або до перевантаження складів. Другою суттєвою проблемою є складність роботи з постачальниками та глобальність ланцюгів постачання. E-commerce бізнеси часто співпрацюють з великою кількістю постачальників, у тому числі закордонних, щоб забезпечити широкий асортимент та конкурентоспроможні ціни. Управління багатьма постачальниками (в різних часових поясах, з різними

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 5    |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

мовами і валютами розрахунків) потребує значних зусиль: треба відстежувати надійність та якість поставок, порівнювати умови, вести перемовини про ціни і контракти. Забезпечення стійких відносин з постачальниками – важливий чинник успішних закупівель, але досягти цього нелегко [12, 13].

Зокрема, бізнесу потрібно мати інструменти для оцінки роботи постачальників, контролю виконання договірних умов, своєчасного оновлення домовленостей. Логістичні та регуляторні аспекти також виступають викликами. При закупівлях за кордоном виникають питання митного оформлення, сертифікації, дотримання імпорتنих вимог. Власна логістика (транспортування товарів від постачальника на склад) потребує координації і може впливати на терміни доставки кінцевим клієнтам.

Крім того, електронна комерція часто змушена відповідати підвищеним очікуванням споживачів щодо швидкості доставки замовлень. Це ставить непрямі вимоги і до системи закупівель: вона має забезпечити наявність товарів тоді, коли вони потрібні для виконання замовлень, інакше навіть бездоганна логістика «останньої милі» не врятує ситуацію. Ще одним аспектом є технологічні виклики і безпека. Багато підприємств все ще переходять від застарілих, паперових або ручних методів закупівель до цифрових. Опір змінам та нестача компетенцій персоналу можуть гальмувати впровадження нових систем – ця проблема відома як «barrier to adoption» електронних закупівель.

Навіть впровадивши сучасну систему, компанія має подолати «людський фактор»: навчити працівників, змінити усталені процеси і культуру. Окрім того, цифровізація несе ризики кібербезпеки: розширення ІТ-інфраструктури закупівель відкриває нові потенційні вектори атак. Конфіденційні дані про контрагентів, ціни, контракти повинні бути захищені. Сучасні системи закупівель потребують надійних засобів кібербезпеки та відповідності стандартам захисту даних, адже компрометація системи може паралізувати постачання і завдати фінансових втрат.

Підсумовуючи, предметна область управління закупівлями в електронній комерції характеризується такими ключовими потребами: точне планування та

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 6    |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

прогнозування потреб, оперативний процес закупівель (мінімум затримок від виявлення потреби до отримання товару), прозорі відносини з постачальниками (контроль якості, термінів, виконання угод), інтеграція закупівель з іншими бізнес-процесами (продажами, складським обліком, фінансами), висока гнучкість і адаптивність до змін (ринку, асортименту, нормативних умов) [14].

Серед основних проблем, що потребують вирішення: неефективність ручних процесів (що призводить до довгих циклів закупівель, помилок та втрати вигод від знижок за швидку оплату), розрізненість інформації (відсутність єдиного джерела правди щодо запасів, замовлень та постачань), труднощі масштабування процесів під час росту бізнесу, а також ризики перерв поставок у турбулентних умовах ринку. Саме ці проблеми створюють запит на сучасні інформаційні системи, що автоматизують та оптимізують процеси закупівель.

## 1.2 Порівняльний аналіз переваг та недоліків існуючих рішень

Сьогодні на ринку представлено ряд програмних продуктів та платформ, призначених для автоматизації та управління закупівлями. Вони різняться за функціональністю, масштабом, вартістю та цільовою аудиторією (малим бізнесам, корпораціям, державному сектору тощо) [15, 16].

Розглянемо найбільш поширені рішення – як міжнародні, так і українські – їх сильні та слабкі сторони, особливо в контексті потреб електронної комерції. SAP Ariba. Одне з провідних світових рішень у сфері електронних закупівель – платформа SAP Ariba. Це хмарна система, яка забезпечує повний цикл source-to-pay (від пошуку постачальника до оплати) та має власну глобальну мережу постачальників Ariba Network. Функціонально SAP Ariba пропонує широкий набір модулів: управління витратами (Spend Management), модуль стратегічного sourcing (проведення тендерів, аукціонів на закупівлю), управління контрактами, управління постачальниками, модуль Procure-to-Pay для формування замовлень і обробки рахунків тощо [17].

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 7    |

Вбудовані інтелектуальні інструменти, як-от рекомендації при закупівлях (Guided Buying), аналітика даних та автоматизація робочих процесів, дозволяють оптимізувати процес закупівель і дотримуватися кращих практик.

Для великого e-commerce бізнесу Ariba приваблива можливістю уніфікувати всі процеси закупівель на одній платформі та підключитися до глобальної бази постачальників, що може сприяти встановленню нових вигідних партнерств. За даними компанії SAP, використання такої хмарної платформи допомагає прискорити процеси закупівель, зменшити кількість помилок, підвищити продуктивність та скоротити витрати за рахунок автоматизації [19, 20].

Втім, недоліки SAP Ariba також добре відомі. По-перше, ця система орієнтована на середній та великий бізнес, відповідно вартість впровадження та користування є досить високою. Ліцензійні платежі та плата за транзакції (у деяких моделях – комісія від суми контрактів) можуть бути непосильними для малого підприємства. По-друге, глибока функціональність обертається складністю: користувачі відзначають, що інтерфейс Ariba не надто інтуїтивний, а сама система має круту криву навчання.

Персоналу потрібен час і тренування, щоб ефективно працювати в Ariba, особливо якщо раніше не було досвіду з подібними системами. Також інтеграція Ariba з іншими IT-системами може бути викликом: вона найкраще інтегрується з ERP-системами SAP, тоді як з продуктами інших виробників часто вимагає додаткових зусиль та налаштувань.

До потенційних мінусів деякі користувачі відносять певну негнучкість стандартних процесів – Ariba задає кращі практики, але іноді бізнесу хочеться їх змінити під себе, що може потребувати залучення консультантів.

Odoo (модуль Закупівлі). Odoo – це популярна open-source ERP-система, яка має модульний архітектурний підхід. Серед багатьох модулів Odoo є модуль Purchase (Закупівлі), що забезпечує базову функціональність для управління процесом закупівель. Цей модуль інтегрований з модулями Склад (Inventory),

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 8    |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

Продажі, Бухгалтерія та іншими, утворюючи єдину інформаційну систему підприємства [21].

Перевага Odoo - гнучкість та економічність. Система є умовно безкоштовною (версія Community) і дуже налаштовуваною: можна допрацьовувати модулі під специфічні бізнес-вимоги, додавати нові функції через програмування або встановлення сторонніх додатків. Модуль закупівель Odoo забезпечує весь базовий цикл: від створення запиту на комерційну пропозицію (RFQ) до перетворення його в замовлення постачальнику, відстеження отримання товарів та виставлення рахунків від постачальника.

Система дозволяє вести облік постачальників, їхніх пропозицій, автоматично реєструє всі дані про покупки для подальшого аналізу, а спільна база даних дає змогу уникнути дублювання інформації.

Для e-commerce компанії середнього розміру Odoo може стати привабливим рішенням завдяки невисокій вартості впровадження, відсутності прив'язки до одного вендора і повній інтеграції закупівель з іншими операціями бізнесу (наприклад, автоматичне оновлення залишків на складі після отримання товару, синхронізація з замовленнями клієнтів тощо). Недоліки Odoo полягають в тому, що «з коробки» модуль закупівель у ньому менш функціонально насичений, ніж у спеціалізованих систем типу Ariba. Компаніям може бракувати деяких просунутих можливостей, таких як вбудовані інструменти проведення тендерів між постачальниками, розширена аналітика витрат чи автоматизовані сценарії погодження бюджету. До того ж Odoo потребує певної технічної експертизи: хоча базові налаштування під силу й непрофесіоналам, для серйозної адаптації системи під бізнес-процеси потрібні розробники або компетентні інтегратори [22, 23].

Відкритий код – це і перевага (гнучкість), і ризик, адже неправильні доопрацювання можуть вплинути на стабільність системи. Ще один можливий виклик – масштабованість: Odoo чудово працює в компаніях малого і середнього бізнесу, але при дуже великому обсязі транзакцій чи складній структурі корпорації може знадобитися більше продуктивності та контролю, ніж дає типовий розгорт

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 9    |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

Odoo. Zoho Inventory. Zoho Inventory – це хмарний програмний продукт, орієнтований на малий та середній бізнес, який здебільшого фокусується на управлінні запасами і замовленнями. Він є частиною екосистеми Zoho (що включає CRM, бухгалтерію тощо) [24]. В контексті закупівель Zoho Inventory надає базові інструменти: створення і відстеження замовлень постачальникам (Purchase Orders), управління рівнями запасів з можливістю встановлення порогів для автоматичних поповнень, облік постачальників та отриманих товарів [25, 26].

Сильні сторони Zoho Inventory – це простота використання (інтерфейс зрозумілий, система готова до роботи фактично відразу після реєстрації) та інтеграція з платформами електронної комерції. Зокрема, Zoho Inventory добре підходить онлайн-ритейлерам, що продають через власний сайт, Amazon, eBay, Shopify та інші канали, оскільки дозволяє об'єднати інформацію про продажі з різних каналів і синхронізувати залишки товарів. Сервіс є хмарним (SaaS), тому не потребує витрат на власний сервер чи складну установку; до того ж, він має мобільні додатки, що дає можливість керувати запасами і закупівлями «на ходу».

Важливо, що поріг входу для бізнесу дуже низький – існує безкоштовний тариф (з обмеженою кількістю транзакцій), який дозволяє протестувати функціонал і поступово перейти на платні плани по мірі зростання бізнесу [27].

Недоліки Zoho Inventory проявляються, коли бізнес масштабується або потребує більш спеціалізованих функцій. По-перше, обмеження функціоналу у безкоштовній та базових версіях: наприклад, ліміт на кількість замовлень (до 20 на місяць у безкоштовному плані) може швидко стати перешкодою для інтернет-магазину, що активно росте. Деякі розширені можливості – такі як управління виробництвом (BOM, складання продуктів), прогнозування попиту – відсутні або доступні лише при інтеграції з іншими продуктами Zoho чи сторонніми додатками.

Фактично Zoho Inventory не є повноцінною ERP чи глибокою системою закупівель, а радше модулем для управління запасами з мінімальними можливостями закупівель; для повного циклу може знадобитися підключення Zoho Books (для рахунків), Zoho Analytics (для глибокої звітності) тощо [28, 29].

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 10   |

Також інколи користувачі скаржаться на часткову відсутність локалізації – система більше пристосована під міжнародні ринки, а для українського бізнесу може не мати, наприклад, україномовного інтерфейсу або підтримки податкових документів за місцевими стандартами (хоча це можна обійти кастомізаціями). Підтримка Zoho надається в режимі 24/5 (будні дні), що потенційно створює незручності для бізнесів, яким потрібна допомога в вихідні чи неробочі години.

Zakupki.Prom.ua (Zakupivli.PRO). Для українського ринку значущим гравцем у сфері електронних закупівель є платформа Zakupki.Prom.ua (нині відома також як Zakupivli.pro). Це онлайн-майданчик для проведення тендерів та торгів, створений компанією EVO, яка є співзасновником системи державних закупівель Prozorro.

Платформа Zakupki.Prom.ua обслуговує як державні закупівлі (у взаємодії з центральною базою даних Prozorro), так і комерційні закупівлі приватних підприємств, які бажають здійснювати їх у прозорий та конкурентний спосіб.

Фактично, для бізнесу це можливість організувати власний тендер: підприємство-замовник оголошує закупівлю (тендер) на платформі, постачальники подають пропозиції, після чого може проводитись аукціон на зниження ціни або відбір кращої пропозиції за визначеними критеріями. Переваги Zakupki.Prom.ua – це широка база постачальників та підрядників, які зареєстровані в системі (особливо завдяки інтеграції з Prozorro).

Для українських компаній це означає доступ до десятків тисяч потенційних постачальників, у тому числі з держсектора, що вже мають досвід електронних торгів. Платформа забезпечує прозорість та чесну конкуренцію: всі кроки тендеру фіксуються, є інструменти оскарження, що підвищує довіру до процесу. Для бізнесу, який прагне отримати найвигіднішу цінову пропозицію або знайти нового постачальника, такий підхід може дати значну економію. Також платформа надає юридичний та методичний супровід: навчальні матеріали, консультації, допомогу в підготовці тендерів, що важливо, оскільки процеси можуть бути складними для новачків.

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 11   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

Тобто, компанія все одно має подбати про перенесення результатів торгів (укладених договорів, цін тощо) у свою ERP/облікову систему вручну або через доробки. Для наочності проведемо порівняння розглянутих рішень у вигляді таблиці, відобразивши їх основні характеристики, переваги та недоліки(табл. 1.1).

Таблиця 1.1 – Порівняння існуючих рішень

| Рішення         | Опис і призначення   | Переваги  | Недоліки  |
|-----------------|--|---|---|
| SAP Ariba       | Хмарна система закупівель для великих компаній, інтегрована з глобальною мережею постачальників. | Потужний функціонал, автоматизація, глобальна база. | Висока вартість, складність, залежність від SAP.                        |
| Odoo (Purchase) | Open-source модуль ERP-системи з широкою інтеграцією та можливістю налаштування.                 | Дешево, гнучко, інтегрується з іншими модулями.     | Необхідність налаштувань, базовий функціонал, обмежена масштабованість. |
| Zoho Inventory  | Хмарне рішення для малого/середнього бізнесу з фокусом на e-commerce.                            | Простота, інтеграція з платформами, доступність.    | Обмежений функціонал, потреба в інших модулях, складність локалізації.  |
| Zakupki.Prom.ua | Українська платформа для тендерних закупівель у держсекторі та бізнесі.                          | Велика база постачальників, прозорість              | Не підходить для щоденних закупівель                                    |

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |

Як видно з аналізу, кожне з рішень має своє призначення та особливості. Для невеликого інтернет-магазину, що шукає просте рішення «все в одному», може підійти Zoho Inventory або впровадження модуля закупівель в Odoo – вони забезпечать базові потреби та інтегруються з продажами. Великі ж гравці e-commerce ринку, з сотнями постачальників і складною структурою погоджень, більше виграють від впровадження потужних платформ на кшталт SAP Ariba або аналогічних (наприклад, Coupa, Oracle Procurement Cloud), адже ті дають більш глибокий контроль і аналітику. Український бізнес також має можливість використовувати локальні платформи на кшталт Zakupivli.Pro для досягнення прозорості і економії на закупівлях через конкуренцію. Однак жодне з існуючих готових рішень не є ідеальним «з коробки» саме для конкретної проблемної області нашого дослідження – управління закупівлями в сфері електронної комерції з врахуванням локальних умов. Це обумовлює необхідність пошуку або розробки підходу, що поєднає кращі сторони існуючих рішень і врахує виявлені недоліки.

### 1.3 Підходи до вирішення задачі за темою дослідження

Для вирішення проблем автоматизації закупівель в електронній комерції можна виокремити кілька підходів, що склалися в теорії та практиці. Ці підходи стосуються як вибору стратегії впровадження інформаційної системи (розробити власну або використовувати наявні платформи), так і використання сучасних технологій та методологій оптимізації процесів закупівель [30].

Один з класичних підходів – це інтеграція функцій закупівель у загальну систему управління підприємством (ERP). Як було розглянуто вище на прикладі Odoo, ERP забезпечує єдину інформаційну базу для всіх процесів. Підхід полягає у тому, щоб провести реінжиніринг процесів закупівель у компанії (проаналізувати та оптимізувати їх), а потім налаштувати ERP-модуль згідно з цими процесами. Перевагою є цілісність даних – усі відомості про товари, запаси, замовлення клієнтів та закупівлі знаходяться в одній системі, що виключає розриви і затримки в передачі інформації [31, 32].

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 13   |

У випадку електронної комерції ERP-підхід дозволяє, наприклад, автоматично створювати заявку на закупівлю, коли залишок товару падає нижче певного рівня, або резервувати на складі товари під замовлення та одразу генерувати потребу на поповнення. Багато сучасних ERP мають вбудовані засоби роботизації процесів (RPA) – тобто виконання рутинних операцій автоматично за заданими правилами. Згідно з тенденціями, що спостерігаються у світі, дедалі більше компаній впроваджують цифровізацію та автоматизацію у сфері закупівель: використання алгоритмів автоматичного створення замовлень, обробки рахунків, нагадувань про постачання тощо вже стало нормою.

Таким чином, підхід через ERP або аналогічну комплексну систему забезпечує автоматизацію наскрізного процесу procure-to-pay і його тісний зв'язок з іншими підсистемами компанії [33]. Інший підхід – це впровадження окремої системи, спеціально призначеної для функції закупівель. Сюди належать згадані рішення типу SAP Ariba, Coupa, Oracle Procurement Cloud, Ivalua тощо.

Подібні платформи зазвичай пропонують найкращі галузеві практики «вбудованими» у систему. Підхід часто реалізується у великих компаніях: виділяється закупівельна функція, впроваджується профільне ПЗ, а вже воно інтегрується з ERP та іншими системами (наприклад, передає дані про витрати у фінансову систему) [34, 35]. Перевагою є глибока функціональність: спеціалізовані системи зазвичай містять розвинуті модулі для стратегічних закупівель (аналіз витрат, управління категоріями, оцінка постачальників), а також забезпечують онлайн-взаємодію з постачальниками (через портали постачальників або мережі на кшталт Ariba Network). Таким чином, налагоджується тісна співпраця із зовнішніми контрагентами: постачальники самі можуть заходити в систему, отримувати замовлення, підтверджувати їх, завантажувати накладні тощо. Цей підхід відповідає концепції електронного узгодження (digital collaboration), що скорочує час і помилки при обміні документами з постачальниками [36].

Недоліком може бути висока вартість та складність інтеграції: компанії доводиться налаштовувати обмін даними між окремою закупівельною платформою

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 14   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

і своїми обліковими системами (нерідко через стандарти EDI – електронного обміну даними, або сучасні API). Попри це, для великих e-commerce гравців, які працюють глобально, такий підхід забезпечує найкращий контроль витрат і зниження операційних ризиків у закупівлях.

Якщо специфіка бізнесу дуже особлива, або ж жодне з ринкових рішень не задовольняє вимоги, компанії можуть обрати шлях розробки кастомної інформаційної системи управління закупівлями. Такий підхід вимагає ретельного формування вимог (що саме має робити система), залучення команди розробників та тісної співпраці між IT-фахівцями і закупівельним відділом [37, 38].

Перевага – повна відповідність потребам: система створюється під конкретні процеси компанії, інтегрується з наявними системами на рівні бази даних чи сервісів настільки глибоко, наскільки це потрібно. Також це дає гнучкість у виборі технологій – можна використати сучасні фреймворки, хмарні сервіси, мікросервісну архітектуру, закласти масштабованість «на виріст». У контексті e-commerce власне рішення може врахувати, наприклад, інтеграцію з веб-сайтом магазину, з маркетплейсами через API, з системами управління складом (WMS) тощо, створюючи єдиний «цифровий хаб» для всіх операцій. Крім того, можна впровадити алгоритми AI/ML для прогнозування попиту, оптимізації замовлень тощо прямо у логіці системи. Світовий тренд демонструє зростаюче впровадження штучного інтелекту та великої аналітики даних у закупівельні процеси – наприклад, AI-алгоритми здатні прогнозувати потребу в закупівлях та пропонувати оптимальний час і обсяг замовлення на основі даних продажів.

Проте власна розробка має і суттєві ризики: високі початкові витрати, тривалий час на створення MVP (мінімально працюючого продукту), потребу у підтримці та оновленнях. Компанія фактично стає власником IT-продукту, що не є її основним бізнесом, і це вимагає відповідних компетенцій.

Нерідко оптимальним шляхом є комбінація вищенаведених підходів. Наприклад, підприємство може спершу впровадити ERP-модуль закупівель для нагляду за базовими операціями, а для проведення періодичних великих тендерів

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 15   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

використовувати зовнішній майданчик (підключивши його через API до своєї системи). Або ж – використовувати хмарний сервіс для певної частини процесу (скажімо, тільки для збору пропозицій і аукціону між постачальниками), а решту робити у власній системі [39, 40].

Такий гібридний підхід дозволяє отримати вигоди від готових рішень там, де це можливо, і закрити унікальні потреби власними засобами. Успіх при цьому залежить від інтеграції систем – важливо, щоб дані плавно передавалися між усіма інструментами. Сучасні технології полегшують це завдання: майже всі хмарні платформи (SAP Ariba, Coupa, Zoho тощо) надають API для інтеграції, є галузеві стандарти обміну даними (наприклад, cXML – Commerce eXtensible Markup Language, що використовується для передачі замовлень, рахунків у системах на кшталт Ariba). Таким чином, компанія може побудувати своєрідну еко-систему закупівельних рішень, де різні компоненти відповідають за різні завдання: одні автоматизують рутинні операції, інші забезпечують аналітику, треті – комунікацію з постачальниками [41].

Окремо варто підкреслити підхід, який акцентує увагу на аналітичному забезпеченні процесу закупівель. Йдеться про впровадження систем бізнес-аналітики (BI) та великих даних для прийняття рішень у закупівлях. Для e-commerce підприємства, яке оперує тисячами SKU (товарних позицій) і має справу з волатильним попитом, критичною є своєчасна аналітика: які товари продаються швидше за очікування і вимагають термінового дозамовлення, де утворюються надлишки, як змінюються ціни постачальників, як коливається курс валюти і впливає на собівартість тощо [42, 43].

Підхід «Data-driven procurement» передбачає, що інформаційна система не лише автоматизує операції, але й надає інформацію для оптимізації стратегії закупівель. Це можуть бути інструменти класифікації витрат (Spend Analysis), рейтингування постачальників за різними KPI, модулі для моделювання різних сценаріїв (наприклад, «що якщо замовити більший обсяг – чи отримаємо знижку і як це вплине на витрати на зберігання?»). В еру цифрової трансформації багато

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 16   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

компаній прагнуть перейти від реактивних закупівель («закупити коли закінчується») до прогнозних та стратегічних закупівель, і тут без належної аналітичної підтримки не обійтися [44].

Таким чином, одним з підходів є впровадження надбудов або модулів до існуючих систем, що забезпечують таку розширену аналітику і використання штучного інтелекту для прийняття рішень. Отже, існує спектр підходів – від використання готових рішень до створення нових, від повної автоматизації окремо взятої компанії до побудови інтегрованих мереж з постачальниками. Вибір підходу залежить від масштабів бізнесу, наявних ресурсів, стратегічних пріоритетів та виявлених проблем [45, 46].

У нашому випадку, враховуючи, що метою є створення інформаційної системи управління закупівлями в сфері електронної комерції, доцільно орієнтуватися на інтегрований підхід. Він може поєднувати: автоматизацію операцій (щоб мінімізувати ручну працю і прискорити процес), використання аналітики і прогнозування (для прийняття обґрунтованих рішень щодо замовлень), а також забезпечення можливості розширення та інтеграції із зовнішніми платформами (наприклад, для проведення тендерів чи підключення нових каналів продажів і постачальників). У наступному підрозділі буде сформульовано конкретну задачу, яку ставить перед собою дана дипломна робота, на основі проведеного аналізу [47].

#### 1.4 Висновки. Постановка задачі

На основі проведеного аналізу предметної області, виявлення проблем та огляду існуючих рішень можна сформулювати задачу, яку належить вирішити в межах даної дипломної роботи. Задача полягає у розробці концепції та прототипу інформаційної системи управління закупівлями для підприємства електронної комерції, яка б усувала виявлені недоліки та задовольняла специфічні потреби такого підприємства. Інакше кажучи, необхідно спроектувати і реалізувати систему, що забезпечить ефективне планування та виконання закупівельного процесу,

|     |      |          |        |      |                            |            |
|-----|------|----------|--------|------|----------------------------|------------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк.<br>17 |
| Зм. | Арк. | № докум. | Підпис | Дата |                            |            |

інтегрованого зі сферою онлайн-продажів. Виходячи з теоретичних основ, сформулюємо ключові вимоги та завдання, які повинна вирішувати ця система:

– система повинна автоматично обробляти типовий цикл від заявки до оплати. Це включає автоматичне створення заявок на закупівлю на основі даних про залишки і продажі, генерацію замовлень постачальникам, відстеження статусів постачання, опрацювання рахунків. Автоматизація має зменшити час циклу закупівель і ймовірність помилок;

– інформаційна система має бути пов'язана з модулем управління продажами (інтернет-магазином). Наприклад, при отриманні великих замовлень від клієнтів система повинна перевіряти наявність товарів і за потреби ініціювати закупівлю. Інтеграція також означає єдину базу номенклатури товарів, спільний облік запасів тощо – щоб уникнути розбіжностей між тим, що продається онлайн, і тим, що замовляється у постачальників;

– система повинна вирішувати задачу надання аналітичної інформації для прийняття рішень. Це включає побудову звітів і рекомендацій: що, коли і скільки замовити. Планується використання методів прогнозування (на основі історичних даних продажів, сезонності) для рекомендацій щодо оптимальних обсягів замовлень, а також інструментів оцінки постачальників (рейтинг за ціною, надійністю, часом доставки). Такий функціонал спрямований на усунення проблеми неоптимальних запасів та реагування на зміну попиту;

– система має забезпечувати зручний механізм комунікації і обміну документами з постачальниками. Ідеально, якщо вона передбачатиме веб-портал або інтерфейс для постачальника (або принаймні інтеграцію через електронну пошту чи API), щоб постачальники могли підтверджувати замовлення, надсилати рахунки тощо;

– система повинна враховувати локальні вимоги: мову інтерфейсу (українську), можливість роботи з гривнею та іншими валютами, особливості документообігу (можливо, інтеграція з електронним документообігом, облік податку на додану вартість тощо);

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 18   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

– поставлена задача передбачає створення рішення, яке зможе масштабуватися разом із ростом бізнесу (кількості товарів, постачальників, замовлень). Архітектура системи має бути модульною, щоби надалі було можливо додавати нові підсистеми (наприклад, модуль управління контрактами, якщо знадобиться, або інтегрувати AI-модуль). Гнучкість також означає налаштуваність бізнес-правил – компанії різних сегментів e-commerce можуть мати свої нюанси у процедурах закупівель, і система повинна дозволяти це відобразити.

Необхідно не просто автоматизувати закупівлі, а зробити це в специфічному контексті електронної комерції, де все відбувається швидко, онлайн і потребує тісної інтеграції ланцюга «постачальник – склад – інтернет-магазин – клієнт». Існуючі рішення, як ми бачили, закривають окремі частини проблеми (хтось пропонує аналітику, хтось – автоматизацію, хтось – зручність для малого бізнесу), але універсального рішення, що відповідає вимогам гнучкості, інтегрованості та локалізованості, не спостерігається.

Реалізувати прототип ключових компонентів системи управління закупівлями та перевірити його роботу на тестових сценаріях, що моделюють типові ситуації електронної комерції (наприклад, різке збільшення попиту на окремий товар, затримка постачання і необхідність швидкого перемикавання на іншого постачальника тощо).

Формулюючи постановку задачі, можна сказати, що метою роботи є створення теоретично обґрунтованої та практично реалізованої моделі інформаційної системи, яка підвищить ефективність управління закупівлями в електронній комерції. Вирішення цієї задачі зробить внесок у підвищення конкурентоспроможності e-commerce підприємств через зниження витрат на закупівлі, прискорення обігу товарів і покращення взаємовідносин з постачальниками, що в кінцевому рахунку сприятиме кращому задоволенню попиту клієнтів і стійкому розвитку бізнесу.

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 19   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

## 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ЗАКУПІВЛЯМИ

### 2.1 Вибір архітектури системи

На першому етапі проєктування необхідно визначити архітектурний підхід до побудови системи закупівель. Враховуючи вимоги продуктивності, безпеки та гнучкості, розглянуто кілька альтернативних варіантів архітектури: монолітна інтеграція з платформою WooCommerce, автономний модуль (мікросервіс) закупівель, а також багат шарова веб-архітектура (трикаскадна клієнт-серверна модель). Кожен підхід має свої переваги та недоліки, які було проаналізовано перед прийняттям рішення. Одним із підходів є монолітна архітектура (інтеграція як плагін WooCommerce). Це один із варіантів реалізувати систему закупівель у вигляді плагіна до WooCommerce (WordPress). Перевагою такого підходу є простота доступу до даних магазину (продукти, замовлення, запаси) безпосередньо через базу даних WordPress та використання внутрішніх API WooCommerce. Моноліт забезпечує єдине середовище виконання, спрощує розгортання і може використовувати наявні механізми автентифікації та ролей WordPress. Проте монолітна інтеграція може ускладнити підтримку та масштабування: будь-які зміни у модулі закупівель впливатимуть на всю систему. Також потенційно знижується гнучкість вибору технологій, оскільки потрібно дотримуватися стеку WordPress (PHP, MySQL). Для вимог високої продуктивності це може стати обмеженням, а модуль закупівель, тісно пов'язаний з ядром WooCommerce, може ускладнити оновлення платформи.

Наступним модулем є автономний модуль (мікросервіс) закупівель. Тут інший підхід – виділити систему управління закупівлями в окремий сервіс, що взаємодіє з WooCommerce через інтерфейси (REST API, вебхуки).

Така мікросервісна архітектура підвищує гнучкість. Модуль закупівель може бути реалізований на оптимальному стеку технологій (наприклад, Python/Django або Node.js) незалежно від основної платформи. Це спрощує незалежне

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 20   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

масштабування – за необхідності сервіс закупівель можна розгорнути окремо на більш потужному сервері або кластері. Крім того, знижується взаємний вплив збоїв. Проблеми в модулі закупівель не зупинять роботу сайту продажів, і навпаки. Недоліком є складність реалізації інтеграції та початкового налаштування взаємодії двох систем. Доведеться забезпечити надійну синхронізацію даних (наприклад, товарних залишків, статусів замовлень постачання) між сервісами. Також мікросервісна архітектура додає накладні витрати на мережеву взаємодію та потребує продуманого керування безпекою (наприклад, захист API).

Незалежно від вибору між монолітом чи мікросервісом, внутрішня побудова системи закупівель може реалізовуватися за принципами багатошарової тривірневої архітектури. У тривірневій клієнт-серверній моделі застосунок розділено на рівень презентації (інтерфейс користувача), рівень застосунку (бізнес-логіка) та рівень даних (сховище даних).

Такий поділ сприяє масштабованості, підтримуваності та ефективності системи, оскільки кожен шар можна розвивати та оптимізувати незалежно. Обравши багатошарову структуру, ми гарантуємо чітке розмежування відповідальностей: інтерфейс веб-застосунку (UI) працює окремо від бізнес-логіки обробки закупівель, а зберігання даних здійснюється у централізованій базі даних. Цей принцип буде використано як у випадку плагіна (в межах WordPress MVC-подібної логіки), так і при реалізації окремого сервісу (де фронтенд, бекенд і база даних розділено). З огляду на вимоги до продуктивності, безпеки і гнучкості, було прийнято рішення реалізувати систему закупівель як автономний веб-застосунок (модуль) з власною базою даних, інтегрований із платформою WooCommerce через стандартизований API. Такий підхід поєднує плюси мікросервісної архітектури (автономність, масштабованість, можливість використання сучасного стеку) із чітким багатошаровим розподілом компонентів. Вибір обґрунтований тим, що WooCommerce надає повноцінний REST API для доступу до даних магазину, що дозволяє зовнішній системі створювати, читати, оновлювати і видаляти дані

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 21   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

WooCommerce (товари, запаси, замовлення тощо) через запити в форматі JSON по HTTPS.

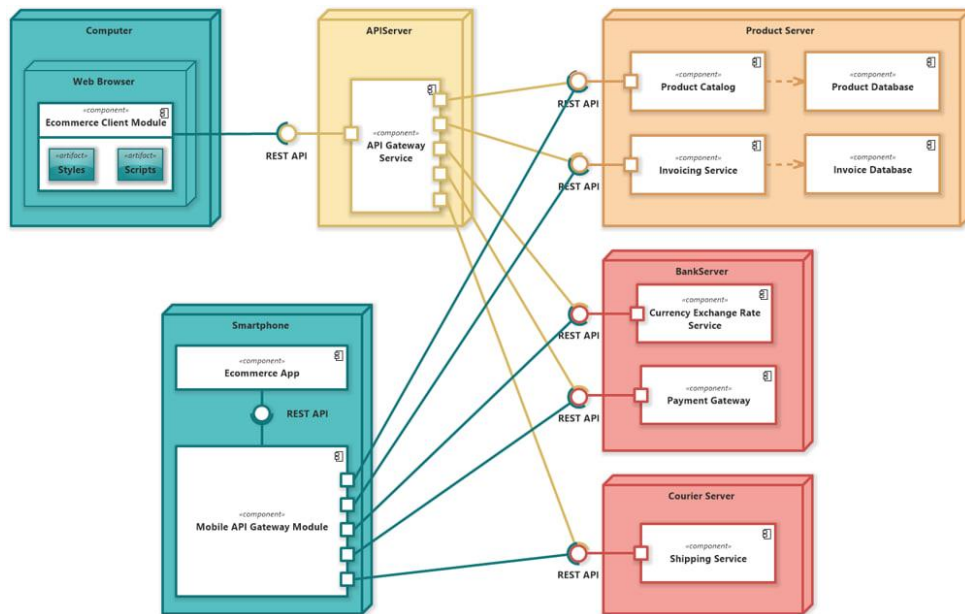


Рисунок 2.1 – Діаграма розгортання архітектури мікросервісів e-commerce

Таким чином, система закупівель може залишатися незалежною, але водночас безперешкодно обмінюватися інформацією з магазином, використовуючи офіційно підтримувані інтеграційні механізми WooCommerce.

Обрана архітектура забезпечує

- гнучкість – можна вибрати оптимальні мови програмування та фреймворки для модуля закупівель;
- продуктивність – сервіс можна розгорнути окремо й масштабувати відповідно до навантаження (наприклад, виділивши більше ресурсів для інтенсивних операцій обробки поставчань);
- безпеку – ізоляція модуля мінімізує вплив потенційних вразливостей на основний сайт, а обмін даними через API захищено стандартними методами автентифікації та шифрування.

З урахуванням обраної архітектури було проведено декомпозицію системи на окремі підсистеми та модулі, кожен із яких виконує специфічні функції у рамках процесу закупівель. Головним завданням такого підходу є забезпечення

модульності, що суттєво спрощує розробку, тестування та подальшу підтримку програмного забезпечення.

Реалізовано можливості додавання й редагування інформації про постачальників, фіксацію договірних умов, умов співпраці, а також зв'язок постачальників із конкретними товарами, які вони постачають. Модуль також зберігає актуальну базу даних постачальників та виступає джерелом інформації для інших компонентів системи, зокрема у процесі формування замовлень.

Модуль товарів та запасів, інтегрується з каталогом WooCommerce, виконує роль проміжної ланки між системою закупівель і складським обліком. Він отримує з WooCommerce перелік товарів, їхні залишки на складах, порогові значення мінімального запасу, а також іншу релевантну інформацію. На підставі цих даних модуль приймає рішення щодо необхідності поповнення запасів, реалізує бізнес-правила визначення точки замовлення та зберігає відомості про постачальників конкретних товарів і їх ціни.

Центральним елементом системи є модуль управління закупівлями, який автоматизує процес створення й обліку замовлень постачальникам. Модуль охоплює весь життєвий цикл закупівлі: від формування запиту пропозицій або заявки на закупівлю до оформлення та супроводу замовлення, його виконання та закриття. Передбачено механізми автоматичного генерування замовлень при досягненні критичних залишків товарів на складі, а також підтримку ручного створення замовлень менеджером. Додатково реалізовано вибір постачальника за визначеними критеріями, функції нумерації замовлень, збереження даних про статуси, управління номенклатурою й обсягами у межах окремих замовлень.

Підсистема управління пропозиціями та підтвердженнями (RFQ/Quote Management) забезпечує підготовку й обробку запитів на отримання комерційних пропозицій від постачальників у випадках, коли необхідне погодження цін або термінів постачання до формування фінального замовлення. Менеджер має змогу надіслати електронний запит із переліком необхідних товарів, а постачальник, у свою чергу, надати відповідь із деталізацією умов. Відстежується статус запитів,

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 23   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

дозволяється залишати коментарі та прикріплювати файли. Після погодження пропозиції дані автоматично переносяться у фінальне замовлення без повторного введення.

Модуль управління поставаннями фіксує фактичне надходження товарів, дає змогу реєструвати часткові поставки, зберігати інформацію про дати, кількість отриманих товарів, супровідні документи та відповідальних осіб. У разі повного виконання поставки система автоматично оновлює залишки на складі у WooCommerce та змінює статус замовлення. За необхідності тут також можуть фіксуватися результати перевірки якості чи відповідності продукції.

Модуль обліку оплат та розрахунків із постачальниками фіксує фінансові операції, пов'язані із закупівлями. Хоча цей компонент не є повноцінною бухгалтерською підсистемою, він дозволяє відзначати факт і спосіб оплати рахунків постачальників, а також формувати базові фінансові документи. Відстежується статус «оплачено/не оплачено» для кожного замовлення, що надає менеджерам можливість контролювати не лише фізичне, а й фінансове закриття закупівель.

Допоміжна підсистема конфігурації та довідників використовується для налаштування основних параметрів роботи системи, зокрема режиму закупівель (автоматичний або ручний), що дає змогу адаптувати систему під специфіку бізнес-процесів організації. Також тут можна задати префікси нумерації документів (наприклад, «PO» для замовлень, «RQ» для запитів пропозицій), встановити порогові рівні запасів для товарів за замовчуванням, налаштувати шаблони електронних листів постачальникам (для розсилки замовлень, запитів, повідомлень). В модулі конфігурації керуються права доступу користувачів системи (напр., менеджер закупівель, адміністратор) та локалізація інтерфейсу.

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 24   |

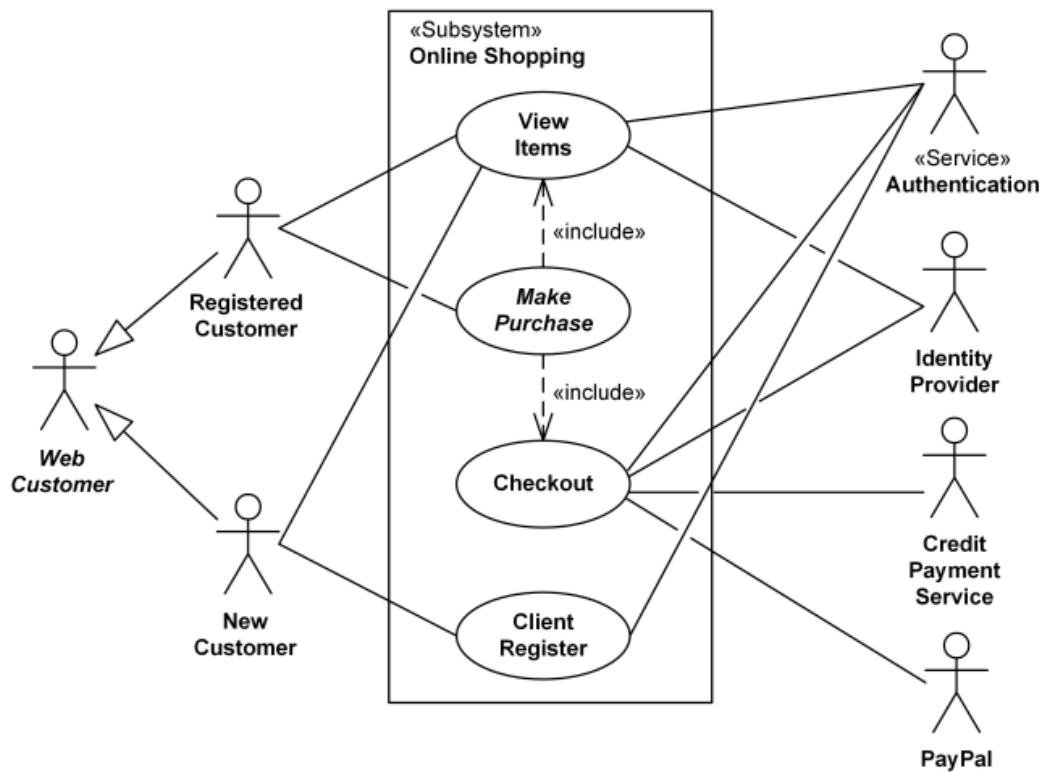


Рисунок 2.2 – UML-діаграма прецедентів використання інформаційної системи закупівель (користувачі: Клієнт/Замовник, Адміністратор, Постачальник).

Клієнт може переглядати товари, формувати і відправляти заявки (purchase requisition) та конвертувати їх у замовлення (purchase order). Адміністратор керує каталогом і користувачами. Постачальник підтверджує поставку.

Інтеграційний модуль WooCommerce API – це спеціальний компонент, що інкапсулює взаємодію з зовнішньою e-commerce платформою. Він містить функції для виклику REST API WooCommerce з метою отримання або оновлення даних. Основні операції: отримати список продуктів та їх властивості; отримати поточні запаси (stock) з WooCommerce; оновити рівень запасів для товару (після надходження поставки); можливо, отримати список замовлень клієнтів, щоб прогнозувати попит. Інтеграційний модуль відповідає за планування опитування WooCommerce або обробку вебхуків – наприклад, WooCommerce може надіслати вебхук при зміні запасів, або ми можемо налаштувати вебхук при оформленні клієнтського замовлення, щоб наш модуль отримав сигнал і перевіряв складські залишки.

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |

Інтерфейс користувача (веб-інтерфейс) забезпечує зручний доступ менеджерам закупівель до всіх перерахованих модулів. Це веб-застосунок (в адміністративному стилі, доступний лише внутрішнім користувачам), що містить відповідні розділи: «Постачальники», «Товари», «Запити на закупівлю/Пропозиції», «Замовлення постачальникам», «Поставки», «Звіти» тощо. Інтерфейс відображає дані з бази, дозволяє вводити і редагувати інформацію, має механізми валідації вводу (щоб, наприклад, не замовити від'ємну кількість товару), пошук та фільтрацію (шукати замовлення за постачальником, статусом, датою). Особлива увага приділяється інформативності. На дашборді системи можуть показуватися показники: кількість товарів з низьким запасом, замовлення в очікуванні постачання, прострочені поставки, тощо.

Формування звітів передбачає підсистема звітності та аналітики для оцінки ефективності процесу закупівель. Наприклад, звіт про закупівлі за період (скільки замовлено, на яку суму, у кого з постачальників які обсяги), звіт про своєчасність постачань (відсоток замовлень виконаних в строк), аналіз рівня запасів (скільки разів товар опускався нижче мінімуму – показник точності планування), тощо. Спершу ця підсистема може бути реалізована мінімально (декілька основних звітів у вигляді таблиць або експорту CSV), але закладається можливість її розширення. Дані для звітів беруться з основної бази (таблиць замовлень, поставок, товарів). Звітність допоможе у розділі 3.4 оцінити ефективність впровадження системи.

Опираючись на визначені модулі, формуємо загальну структуру: ядро бізнес-логіки (модулі закупівель, постачань, постачальників) взаємодіє з зовнішніми системами (WooCommerce API для отримання даних про товари та оновлення складу; e-mail сервер для відправки листів постачальникам). Веб-інтерфейс виступає клієнтом до цього ядра, через який користувачі виконують операції. Усі дані зберігаються у реляційній базі даних системи закупівель, спроектованій спеціально під ці модулі.

На етапі моделювання бази даних було визначено сутності (таблиці) та зв'язки між ними, що необхідні для збереження інформації, якою оперує система

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 26   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

закупівель. Реляційна база даних проєкту спроектована з урахуванням потреб ефективного управління постачальниками, товарами, замовленнями та поставками

Основні сутності інформаційної системи управління закупівлями включають:

– Supplier (Постачальник) – зберігає дані про постачальників. Ключові атрибути: SupplierID (первинний ключ), Name (назва компанії), ContactInfo (контактні дані: адреса, телефон, email), PaymentTerms (умови оплати – наприклад, передплата, відтермінування 30 днів), LeadTime (середній час виконання замовлення, дні), Rating (опційно, оцінка надійності постачальника). Зв'язки – один постачальник може постачати багато товарів, та мати багато закупівельних замовлень;

– Product (Товар) – інформація про товари, що потребують закупівлі. Оскільки система інтегрована з WooCommerce, є два варіанти зберігання довідника товарів: або мати локальну таблицю Product (синхронізовану з продуктами WooCommerce), або не дублювати дані і звертатися до WooCommerce при потребі. У нашому проєкті прийнято зберігати мінімальну локальну інформацію: ProductID (ідентифікатор, що може збігатися з WooCommerce Product ID), Name (назва товару), SKU (артикул), CurrentStock (поточний запас – ці дані можуть оновлюватися з WooCommerce), ReorderPoint (пороговий рівень, при досягненні якого потрібна закупівля), DefaultSupplierID (ID постачальника за замовчуванням, якщо визначено основного). Зв'язки: товар може мати декілька постачальників (альтернативні джерела). Для цього вводиться зв'язна сутність «ProductSupplier»;

– ProductSupplier – зв'язкова таблиця (багато-до-багатьох) між товарами і постачальниками, що зберігає специфічні дані: ProductID, SupplierID – разом утворюють первинний ключ (або окремий surrogate ID), SupplierProductCode (код/артикул товару у постачальника, якщо відрізняється), CostPrice (ціна закупівлі у цього постачальника, остання відома або середня), MinOrderQuantity (мінімальна партія замовлення, якщо встановлено постачальником), LeadTime (час поставки для цього товару, якщо відрізняється від загального для постачальника), Priority

|     |      |          |        |      |  |  |  |  |      |
|-----|------|----------|--------|------|--|--|--|--|------|
|     |      |          |        |      |  |  |  |  | Арк. |
|     |      |          |        |      |  |  |  |  | 27   |
| Зм. | Арк. | № докум. | Підпис | Дата |  |  |  |  |      |

(пріоритет цього постачальника для даного товару, наприклад 1 – основний, 2 – альтернативний). Ця таблиця дозволяє системі визначати, у кого замовляти товар і на яких умовах;

– PurchaseOrder (Закупівельне замовлення постачальнику) – основна транзакційна сутність, що відображає зроблене замовлення. Атрибути: PO\_ID (первинний ключ, код замовлення), PO\_Number (людсько-читаємий номер із префіксом, наприклад «PO-00045»), SupplierID (посилання на постачальника, кому відправлено замовлення), OrderDate (дата створення замовлення), Status (статус: Draft, Sent, Confirmed, Partially Received, Completed, Cancelled тощо), TotalAmount (загальна сума замовлення, розраховується з позицій), ExpectedDeliveryDate (очікувана дата доставки, розрахована або встановлена менеджером), PaymentStatus (статус оплати: Unpaid, Paid, Partial), Comments (примітки). Зв'язки: у одного замовлення – багато позицій (PurchaseOrderDetails); кожне замовлення належить одному постачальнику;

– PurchaseOrderDetail (Позиція замовлення постачальнику) – деталі замовлення, пов'язані з конкретним товаром. Атрибути: PO\_DetailID (PK), PO\_ID (FK на PurchaseOrder), ProductID (товар, FK), QuantityOrdered (замовлена кількість), UnitPrice (ціна за одиницю в замовленні, зафіксована на момент замовлення), Subtotal (розраховується: Quantity \* UnitPrice), ReceivedQuantity (скільки отримано по цій позиції на даний момент; для часткових поставок), Status (optional, статус окремої позиції: Pending, Received, Backordered). Первинний ключ може бути складеним з PO\_ID+ProductID, якщо не допускаємо дублікатів товару в одному замовленні, або окремим полем. Зв'язки: кожна позиція прив'язана до конкретного замовлення і конкретного товару;

– Shipment (Надходження, поставка) – таблиця для відстеження фактів доставки товарів за замовленнями. Оскільки одна PurchaseOrder може виконуватися декількома поставками (частинами), вводимо окрему сутність Shipment. Атрибути: ShipmentID (PK), PO\_ID (FK посилання, яке замовлення постачальника поставляється), ShipmentDate (дата отримання поставки), Status

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 28   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

(наприклад, Received, In-Progress якщо очікується, або Cancelled), ReceivedBy (користувач, що прийняв товар, можна зберігати ім'я або ID зі списку користувачів), InvoiceNumber (номер накладної/рахунку постачальника для цієї поставки), Comments. Зв'язки: поставка належить одному замовленню. Для зберігання конкретно отриманих кількостей товарів може використовуватись або детальна таблиця, або достатньо вказати, що Shipment покриває всі непоставлені товари (якщо часткова, то буде ще одна Shipment для решти).

При необхідності точної фіксації кількості товарів що прийшли в кожній поставці, можна ввести цю сутність ShipmentDetail (Деталі поставки). Атрибути: ShipmentDetailID (PK), ShipmentID (FK на поставку), PO\_DetailID (або ProductID + PO\_ID, зв'язок із позицією замовлення), QuantityReceived (кількість, отримана в цій поставці по відповідній позиції замовлення). Це дозволить відобразити, як розбивалося виконання замовлення по поставках. Якщо ж обсяг інформації не такий критичний, допускається прямо в PurchaseOrderDetail зберігати ReceivedQuantity, оновлюючи його при надходженнях, а Shipment використовувати лише як журнал з одним сумарним записом. У нашому проєкті для повноти даних передбачено ShipmentDetail, але реалізація її використання може бути спрощена на початковому етапі.

За потреби, є Payment (Оплата) окрема таблиця платежів постачальникам. Мінімально можна обмежитися полями в PurchaseOrder (PaymentStatus, PaidDate), але для розширення – PaymentID, PO\_ID, PaymentDate, Amount, PaymentMethod, Reference (номер платіжного документа). Це дає гнучкість, якщо оплата може бути в кілька траншів або одна оплата покривати кілька замовлень (тоді знадобиться зв'язна таблиця). В рамках базового функціоналу, однак, складна модель оплати не потрібна, тому ми фіксуємо оплату простим прапором і датою в самій таблиці замовлення.

Таблиця користувачів системи закупівель User (Користувач) потрібна для контролю доступу і відстеження, хто створив замовлення або прийняв поставку. Атрибути: UserID, Name, Role (Manager, Admin тощо), Email, PasswordHash (для

|     |      |          |        |      |                            |            |
|-----|------|----------|--------|------|----------------------------|------------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк.<br>29 |
| Зм. | Арк. | № докум. | Підпис | Дата |                            |            |

автентифікації, якщо не інтегровано з обліковими записами WooCommerce). У простішому випадку можна використати існуючих адміністраторів WordPress, але автономний сервіс дозволяє мати свою таблицю користувачів або інтегрувати LDAP. У нашому рішенні передбачено окрему сутність User, щоб система могла працювати незалежно.

Конфігураційні та довідкові таблиці, такі як Settings (глобальні налаштування, авто/ручна закупівля, поточний номер наступного замовлення тощо), LookupTables для статусів, типів контактів, методів оплати (наприклад, таблиця PaymentMethod, [ID, Name] – «Bank Transfer», «Credit Card», «Cash» тощо, щоб забезпечити цілісність). Ці допоміжні сутності покращують нормалізацію та підтримуваність бази.

Розроблена модель бази даних задовольняє основні правила нормалізації. Уникає надлишкового дублювання даних, забезпечує цілісність через зовнішні ключі (наприклад, неможливо створити позицію замовлення з неіснуючим товаром чи замовлення на неіснуючого постачальника). Було прийнято рішення зберігати історичні дані, такі як ціна товару у конкретному замовленні, безпосередньо в деталях замовлення (UnitPrice), щоб у майбутньому зміна ціни постачальника не спотворювала дані про минулі закупівлі. Так само зберігаються копії ключових даних на момент транзакції (наприклад, адреса постачальника в замовленні або умови оплати). За бажанням можна дублювати ці поля для архівного збереження, хоча базова модель дозволяє отримувати актуальні дані через зв'язки. Для оптимізації пошуку та відбору передбачені індекси за основними полями фільтрації: по Status в таблиці PurchaseOrder (щоб швидко вибирати відкриті/закриті замовлення), по SupplierID (для звітів по конкретному постачальнику), по ProductID в деталях (для аналізу закупівель по товару). Обсяг даних у системі на перших етапах невеликий, але враховано можливе зростання номенклатури товарів і кількості замовлень, тому індексація і грамотне використання FOREIGN KEY забезпечують масштабованість на рівні даних. В результаті моделювання було побудовано діаграму «Сутність - Зв'язок» (ERD), що

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 30   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

відображає всі основні таблиці та зв'язки між ними. Вона буде наведена в додатках (Схема архітектури БД проєкту). Модель БД узгоджується із функціональними вимогами. Наприклад, для кожного товару можна прив'язати кількох постачальників з різними умовами (таблиця ProductSupplier), система може відслідковувати виконання замовлення по частинах (Shipment/ShipmentDetail), а також дозволяє зберігати записи про транзакції закупівлі й надалі генерувати необхідні звіти.

## 2.2 Інтеграція з e-commerce платформою та іншими бізнес-процесами

Інформаційна система управління закупівлями не функціонує ізольовано – її цінність саме в тому, що вона інтегрована в загальний ланцюжок бізнес-процесів електронної комерції. У нашому випадку базовою платформою є WooCommerce (інтернет-магазин), з якою система закупівель повинна тісно взаємодіяти. Окрім того, враховуються інтеграції з процесами управління складом (інвентаризацією) та, потенційно, з фінансовим обліком і логістикою. Інтеграція з WooCommerce: Основний канал обміну даними – REST API WooCommerce. WooCommerce (починаючи з версії 2.6) має повну інтеграцію з WordPress REST API, що дозволяє через HTTP-запити у форматі JSON отримувати і змінювати дані магазину.

У межах розробки інформаційної системи управління закупівлями визначено кілька ключових аспектів інтеграції з платформою WooCommerce, що забезпечують цілісність та актуальність даних.

Першочергове значення має синхронізація каталогу товарів і запасів. Система закупівель періодично або за подієвим тригером отримує з WooCommerce перелік товарів разом із поточними залишками на складі. Це реалізується шляхом надсилання GET-запитів до REST API WooCommerce (ресурс /wp-json/wc/v3/products із відповідними параметрами, такими як stock\_status). Оновлення може відбуватися з певною періодичністю, наприклад, щогодини, або у режимі реального часу – під час відкриття менеджером сторінки із переліком

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 31   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

товарів система надсилає запит до WooCommerce для відображення актуальних залишків.

Як альтернативу або доповнення до опитування API використовується механізм вебхуків. WooCommerce надає можливість налаштування вебхуків, які спрацьовують при зміні залишків товарів, наприклад, після оформлення клієнтського замовлення. Модуль інтеграції системи закупівель підписується на такі події, після чого WooCommerce надсилає HTTP POST-запит до зовнішнього сервісу при настанні відповідної події (наприклад, `product.updated` із даними щодо оновлення запасу). Отримавши вебхук, система закупівель негайно оновлює власні дані про залишки товару і, за необхідності, ініціює процес формування нового замовлення постачальнику. Цей підхід дозволяє реагувати практично в реальному часі, забезпечуючи своєчасність прийняття рішень щодо поповнення запасів.

Після надходження товарів від постачальника та їх приймання на склад система закупівель повинна синхронізувати ці дані з обліком інтернет-магазину. Для цього використовуються PUT або POST-запити до API WooCommerce, зокрема до ресурсу `/wp-json/wc/v3/products/{id}` з оновленням значення поля `stock_quantity`. У такий спосіб залишки у магазині завжди відповідають фактичній наявності. Для уникнення конфліктів, якщо WooCommerce самостійно керує запасами, система налаштовується так, що керування запасами активоване (`Manage Stock = true`), а зовнішня система отримує дозвіл змінювати відповідне поле.

Передача довідкових даних має свої особливості. Оскільки WooCommerce не містить окремої сутності «постачальни», оновлення цієї інформації не передбачено. Якщо ж виникає потреба відобразити дані про постачальника у кабінеті WooCommerce, можна розширити функціонал платформи за допомогою відповідного плагіну: додати поле «Постачальник» до сторінки товару і забезпечити синхронізацію із зовнішньою системою. Для цього можлива як організація додаткового API, так і передача даних у мета-поля товару при зміні постачальника. Проте наразі впроваджується односторонній режим інтеграції: система закупівель читає й оновлює дані WooCommerce, але WooCommerce не

|     |      |          |        |      |  |  |  |  |      |
|-----|------|----------|--------|------|--|--|--|--|------|
|     |      |          |        |      |  |  |  |  | Арк. |
|     |      |          |        |      |  |  |  |  | 32   |
| Зм. | Арк. | № докум. | Підпис | Дата |  |  |  |  |      |

зчитує інформацію із зовнішньої системи. Це знижує складність і ризики при інтеграції.

Окремий напрямок розвитку інтеграції стосується використання даних про клієнтські замовлення для прогнозування попиту. За умови розширення функціоналу система може аналізувати історичні дані про продажі, отримані через API WooCommerce, і на основі середньоденних обсягів споживання прогнозувати необхідний обсяг поповнення запасів на майбутній період. Такий підхід дозволяє перейти від реактивної до проактивної моделі закупівель, коли рішення приймаються не лише на основі мінімальних залишків, а й із урахуванням прогнозу продажів. У базовій реалізації система орієнтована на реактивний сценарій, але архітектура дозволяє у майбутньому інтегрувати модуль аналітики та прогнозування з подальшою автоматизацією процесів планування закупівель.

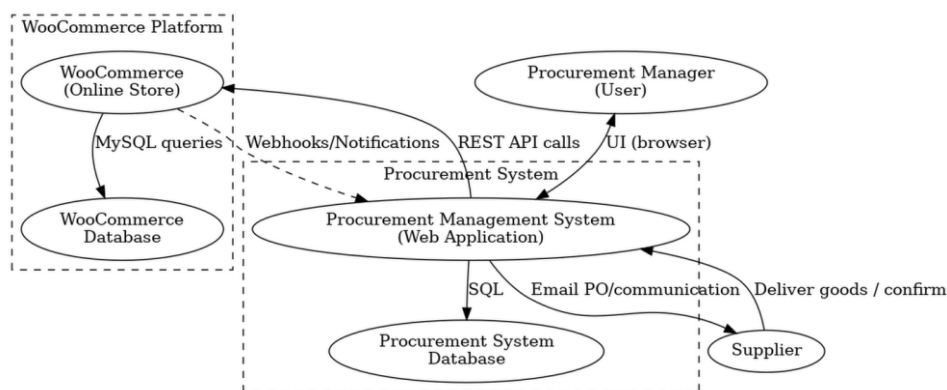


Рисунок 2.3 – Архітектура інтеграції системи закупівель з платформою WooCommerce та учасниками процесу

Система управління закупівлями (праворуч, у пунктирному блоці) – автономний веб-застосунок із власною базою даних, через REST API виконує обмін даними з WooCommerce (ліворуч). Менеджер закупівель працює через веб-інтерфейс системи. Постачальники отримують замовлення електронною поштою і постачають товари, що фіксується в системі. Взаємодія з WooCommerce двостороння: система запитує дані (товари, залишки) та відправляє оновлення

(надходження на склад). WooCommerce базується на MySQL, безпосередній доступ до своєї БД із зовнішньої системи не використовується - лише через API. Для подій (напр. зміна залишків) може використовуватися механізм вебхуків (показано штриховою стрілкою).

Інтеграція з процесами управління складом є важливим аспектом функціонування системи закупівель, особливо в контексті використання WooCommerce, який виконує базові функції складського обліку. Проте, для забезпечення повного циклу управління запасами, необхідно враховувати й фізичний вимір складської діяльності. Бізнес-процес закупівель охоплює всі етапи: закупівлю товару, його фізичне отримання, розміщення на полицях складу та відповідне оновлення даних в інформаційній системі. Таким чином, кіберфізичний вимір проявляється в тому, що всі дії у фізичному середовищі знаходять своє відображення у цифровому контурі системи.

За наявності на підприємстві окремої системи управління складом (WMS – Warehouse Management System) або ERP-системи з функціями складського обліку, доцільно організувати обмін даними між такими системами та модулем закупівель. Зокрема, після надходження товару на склад менеджер може фіксувати цю подію одночасно в системі закупівель та в ERP. У ідеальному випадку, внесення даних в одну систему має автоматично оновлювати інформацію в усіх задіяних системах. У реалізованому проекті, що орієнтується на WooCommerce, його функціонал (разом із модулем закупівель) фактично виконує роль WMS. У разі розширення передбачена можливість експорту даних про поставки та запаси у форматах, які підтримують сторонні системи, або ж синхронізація інформації через API ERP.

Сучасні тенденції розвитку автоматизації складських процесів передбачають впровадження технологій Інтернету речей (IoT). Наприклад, для автоматичного обліку залишків можуть використовуватися вагові датчики або RFID-зчитувачі, встановлені безпосередньо на полицях складу. Інтеграція таких пристроїв дозволяє системі закупівель у режимі реального часу отримувати дані про зміну залишків без участі людини. Більше того, технології RFID можуть забезпечувати

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 34   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

автоматизацію інвентаризації та навіть ініціювати процес формування замовлень безпосередньо з фізичної зони складу. Таким чином, IoT-рішення дозволяють істотно скоротити витрати часу і зменшити людський фактор у процесах обліку та закупівель.

Ще одним важливим аспектом є організація логістики та доставки. Після здійснення закупівлі виникає потреба в організації транспортування товарів до складу, а також, у разі необхідності, подальшої доставки кінцевому споживачеві. Наразі система закупівель опосередковано взаємодіє з логістичними процесами шляхом оновлення статусів поставок. Однак існує перспектива інтеграції з кур'єрськими службами чи транспортними модулями. Наприклад, під час оформлення замовлення постачальнику система може автоматично генерувати запит на транспортування або зберігати трек-номер відправлення для подальшого відстеження руху товару. Хоча ці функції наразі не є автоматизованими, система дозволяє зберігати відповідну інформацію (наприклад, у полях коментарів чи окремих полях для трекінгу), що полегшує роботу менеджера. Інтеграція з логістичними сервісами через API може стати перспективним напрямом розвитку для забезпечення автоматичного підтвердження доставки.

Необхідно також враховувати адаптацію системи закупівель до особливостей бізнес-процесів конкретної організації. У деяких компаніях перед оформленням замовлення постачальнику формується внутрішня заявка на закупівлю (purchase requisition) від інших підрозділів – наприклад, від відділу продажів або виробництва. Оскільки у запропонованому проекті система орієнтована на електронну комерцію, потреба у закупівлі визначається автоматично – на основі даних про залишки на складі або у рамках планування акцій. Відтак, окреме моделювання процесу внутрішніх заявок не передбачено: менеджер закупівель приймає рішення на підставі поточних показників запасів та політики компанії..

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 35   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

## 2.3 Висновки до другого розділу

У другому розділі дипломної роботи проведено комплексне проектування інформаційної системи управління закупівлями з орієнтацією на її інтеграцію в середовище електронної комерції. Було обґрунтовано вибір клієнт-серверної багатошарової архітектури, що передбачає поділ на рівні презентації, бізнес-логіки та зберігання даних, що забезпечує модульність, масштабованість та зручність супроводу системи.

На основі структурного аналізу сформовано логічну схему системи, визначено функціональні модулі: управління постачальниками, товарами, замовленнями, поставками та звітністю. Важливим етапом стало моделювання бази даних, у якій передбачено реляційні зв'язки між ключовими сутностями, що дозволяє забезпечити цілісність та узгодженість інформації.

Розглянуто можливості інтеграції з платформою WooCommerce через REST API - реалізація такої взаємодії дозволяє здійснювати синхронізацію даних про запаси, товари та операції між системою закупівель і e-commerce платформою. Також проаналізовано аспекти безпеки (автентифікація, контроль доступу, резервування даних) та локалізації системи для підтримки україномовного інтерфейсу й адаптації до бізнес-процесів національного ринку.

У результаті проектування створено цілісну концепцію системи, здатної автоматизувати основні процеси закупівельної діяльності та інтегруватися з цифровим торговим середовищем. Це заклало фундамент для подальшої реалізації програмного прототипу та його впровадження в умовах реального бізнесу. Вибір стеку технологій для реалізації системи закупівель базувався на критеріях продуктивності, безпеки, гнучкості, масштабованості й сумісності з WooCommerce.

Підсумково, стек включає Python 3.11, Django 4.x, MySQL 8, Bootstrap 5, Docker, Git, інтеграцію через REST API з WooCommerce.

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 36   |

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

#### 3.1 Розробка прототипу веб-застосунку

На початковому етапі розробки було створено функціональний прототип веб-застосунку, щоб перевірити основні ідеї і забезпечити базову працездатність системи. Прототип реалізовано з використанням обраного стеку технологій (Django, MySQL, Bootstrap) і включає ключові компоненти: модель даних, простий інтерфейс та інтеграцію з WooCommerce API.

Налаштування середовища розробки: Розробку розпочато в середовищі Docker. Підготовано `docker-compose.yml`, який запускає два сервіси: `app` (Python:3.11, Dockerfile якого встановлює Django та залежності) і `db` (образ MySQL 8.0). Після запуску контейнерів проведено ініціалізацію Django-проєкту командою `django-admin startproject procurement_system`. Далі, всередині проєкту, створено додаток `purchases` (`python manage.py startapp purchases`), де знаходиться основний код (моделі, `views`, `templates`). Здійснено базову конфігурацію: параметри підключення до бази MySQL (хост=`db`, ім'я БД, логін/пароль), налаштування `static files` (для Bootstrap), параметри локалізації (`LANGUAGE_CODE = 'uk'`, `TIME_ZONE = 'Europe/Kyiv'`), і додано додатки `'purchases'` та інші (для адміністративної панелі, `auth`).

Моделювання даних у кодї: Створено класи моделей Django. Наприклад, модель `Supplier` виглядає як клас, успадкований від `django.db.models.Model`, з полями: `name = models.CharField(max_length=255)`, `contact_info = models.TextField()`, `payment_terms = models.CharField(max_length=100, blank=True)`, тощо. Задано метод `__str__` для зручності відображення (назва постачальника).

Після визначення моделей, виконано міграції (`python manage.py makemigrations` та `migrate`), що створило відповідні таблиці у MySQL. Прототипна база даних наповнена тестовими даними: додано декілька постачальників, товарів і навіть один-два записів `PurchaseOrder` через Django admin (щоб перевірити чи зв'язки працюють).

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 37   |

Адміністративний інтерфейс: Django автоматично генерує admin-панель для зареєстрованих моделей. Для швидкого доступу до даних, зареєстровано основні моделі у admin.py. Це дозволило через браузер перевірити CRUD операції: додати постачальника, товар, зробити замовлення. Адмін-панель не є кінцевим інтерфейсом для користувача, але на початку розробки слугувала засобом тестування моделей і взаємодії з БД.

Інтеграція з WooCommerce (прототипна): На момент створення прототипу, для спрощення, WooCommerce API викликався на тестовому магазині. Було розгорнуто локально простий сайт WordPress+WooCommerce з декількома товарами, щоб мати ендпоінт для запитів. В Django-проєкті створено окремий модуль woocommerce\_service.py, де реалізовано клас WooCommerceAPI (обгортка над пакетом woocommerce або requests). Налаштування ключів API зберігаються у settings.py (конфіденційні дані винесені у змінні середовища і передані в контейнер). Протестовано базовий сценарій: метод fetch\_products() виконує GET /products і повертає JSON списку товарів.

Цей функціонал в прототипі використано, наприклад, щоб при відображенні сторінки «Список товарів» підтягувався актуальний залишок зі стороннього WooCommerce. Для перевірки реалізації, розроблено тимчасовий View у Django: sync\_products(request), який викликає WooCommerceAPI.fetch\_products(), порівнює зі своїм довідником Product і додає нові товари або оновлює залишки. Цей view повісили на URL /sync\_products/. У браузері, зайшовши за цією адресою (під адміном), бачили JSON або повідомлення про успішну синхронізацію. Таким чином переконалися, що доступ до WooCommerce працює, підпис валідний і дані отримуються (WooCommerce повертає масив продуктів з полями, серед яких stock\_quantity – саме його зберігали в своєму Product.current\_stock).

Список постачальників: таблиця з усіма постачальниками (ім'я, контакти, рейтинг і т.д.) з посиланнями «Редагувати»/«Видалити». Реалізація: View supplier\_list вибирає всі Supplier з БД, передає їх у шаблон. Шаблон використовує цикл for для побудови таблиці. Підключено Bootstrap для стилізації (клас .table

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 38   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

.table-striped для таблиці тощо). Також додано форму додавання нового постачальника (поле ім'я, контакт).

Сторінка товарів: відображає товари, їх мінімальні рівні запасів, поточні запаси. Тут інтегровано AJAX: біля кожного товару кнопка «Оновити залишок». Натиснувши її, скрипт JS відправляє запит /product/{id}/update\_stock/ (створено відповідний маршрут), де view Django викликає WooCommerce API для цього товару, оновлює Product.current\_stock і повертає нове значення (JSON).

Форма створення замовлення постачальнику: прототипово реалізовано упрощено: сторінка з випадаючим списком постачальників, багато рядків для вибору товару і введення кількості. Щоб не ускладнювати, використано можливість Django Formset або просто кілька полів, названих з індексами (product1, qty1, product2, qty2...). Обробник форми (create\_po view) приймає дані POST, створює PurchaseOrder об'єкт і пов'язані PurchaseOrderDetail. Перевірка полягала у тому, що після сабміту замовлення з двома позиціями в базі з'являються 1 запис в purchase\_order і 2 записи в purchase\_order\_detail, а також генерується номер (PO\_Number). Для генерації номерів зроблено простий підхід: PO\_Number = «PO-« + (<згенерований ID або послідовний глобальний лічильник>). У прототипі можна навіть було використати ID із БД, але вирішено додати окремий лічильник у Settings, щоб продемонструвати гнучкість – такий підхід узгоджується з вимогами (налаштовуваний префікс).

Список закупівельних замовлень: прототипна сторінка, що показує всі PurchaseOrder: номер, постачальник, дата, статус, суму. Натиснувши на номер, можна перейти до деталей – окремий view po\_detail, який показує склад замовлення, і якщо воно «NEW», пропонує кнопки «Надіслати постачальнику» або «Скасувати». Ці кнопки поки що просто міняли статус (для send – встановлювали «SENT» та відправляли email, для cancel – «CANCELLED»).

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 39   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

### 3.2 Реалізація основних функціональних модулів

Особливу увагу приділено модулю управління закупівлями (створення та обробка закупівельних замовлень), інтеграційному модулю WooCommerce API, та модулю поставок, оскільки вони складають серцевину системи. Реалізований відносно просто, використовуючи можливості Django CRUD. Створено відповідні представлення (views) для списку постачальників, створення нового, редагування та видалення.

Список постачальників: View `SupplierListView` (клас, успадкований від `ListView Django`) вибирає всі `Supplier` і відображає шаблон `supplier_list.html`. Таблиця містить колонки: Назва, Контактні дані, Рейтинг (якщо використовується), та дії (Редагувати, Видалити). Для оформлення використано Bootstrap-стилі, а також додано поле пошуку у верхній частині (реалізовано відфільтровування записів по введенню – через GET параметр `?q=` у URL, при обробці view фільтрує `Supplier.objects.filter(name__icontains=q)` якщо параметр задано).

Додавання/Редагування: Використано Django `ModelForm` для `Supplier` – клас форми `SupplierForm`, який базується на моделі `Supplier` і містить поля `name`, `contact_info`, `payment_terms`. У шаблоні `supplier_form.html` форма викладається з допомогою Bootstrap-класів (`form-control` для інпутів). При сабміті, `SupplierCreateView` перевіряє валідність форми і зберігає новий запис, після чого перенаправляє на список із повідомленням успіху («Постачальника додано»). Аналогічно `SupplierUpdateView` дозволяє редагувати. Поле `rating` (якщо вирішено зберігати) можна зробити, наприклад, як `ChoiceField` (А, В, С – умовна класифікація постачальників) або числове.

Як було обкатано в прототипі, реалізовано функцію оновлення запасів. У фінальній системі це оформлено як фонові задачі. Використано бібліотеку `Celery` (із `Redis` як брокер) для асинхронних задач: при отриманні вебхука або за розкладом (`Crontab`, наприклад раз на ніч) запускається задача `sync_product_stock()`, яка проходиться по всіх продуктах або отримує від `WooCommerce` список змін. Це

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 40   |

дозволило зняти навантаження з веб-запитів. Проте базові можливості (кнопка «Оновити зараз») залишено для вручну ініційованого оновлення.

Перегляд довідника товарів: `ProductListView` відображає таблицю товарів. Тут для кожного товару показано назву, постачальників (можливо, головного постачальника, або іконку якщо декілька), мінімальний залишок і поточний залишок. Якщо поточний залишок  $<$  мінімального, рядок підсвічується або додається значок попередження. Ця логіка реалізована в шаблоні: `{% if product.current_stock <= product.reorder_point %} <span class="badge bg-danger">Низький запас</span> {% endif %}`.

Додавання/Редагування товару: Форма `ProductForm` дозволяє ввести назву, SKU, мінімальний запас, вибрати один чи кількох постачальників та пріоритет/ціну. Реалізація вибору множинного постачальника – або кілька полів на формі (перший постачальник + ціна, другий + ціна), або кастомна форма. У Django, `ManyToMany` можна редагувати через `ModelMultipleChoiceField`, але тоді нема місця для додаткових полів (ціна). Тому зроблено наступне: на формі головні поля товару (назва, sku, min\_stock), а для постачальників – окремий інтерфейс після збереження товару: на сторінці деталей товару таблиця «Постачальники цього товару» з кнопкою «Додати постачальника до товару». Ця кнопка відкриває модальне вікно (через Bootstrap modal) з формою (поля: постачальник [dropdown], ціна, пріоритет, мін. партія). Після сабміту, через AJAX додавання відбувається і таблиця оновлюється. Такий UX більш зручний, ніж одразу велика форма.

У класі `Product` додано метод `get_preferred_supplier()` – він знаходить з пов'язаних через `ProductSupplier` записів той, що має найвищий пріоритет (найнижче число в полі `priority`). Це використовується, коли система автоматично формує замовлення: саме цьому постачальнику піде заявка першочергово.

В інтерфейсі ця інформація теж може відображатись (наприклад, зірочкою відмічати основного постачальника товару).

Система виявляє товар, що впав нижче порогу, вона створює чернетку замовлення (`Status = «Draft»` або «Auto-created»). Наприкінці проходження по

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 41   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

товарах, якщо `product.current_stock < product.reorder_point`, система перевіряє, чи не існує вже відкритого замовлення-чернетки для цього товару. Якщо ні – створює новий `PurchaseOrder` з постачальником = `product.get_preferred_supplier()`, і додає `PurchaseOrderDetail` з цим товаром і кількістю = наприклад,  $(reorder\_point * 2) - current\_stock$  (тобто поповнити до подвоєного мінімуму), або за фіксованою логікою. Цю кількість, звісно, можна налаштувати. Статус ставиться «Draft», щоб менеджер перевірів і підтвердив відправку.

Менеджер через UI натискає «Створити замовлення». Відкривається форма з вибором постачальника і додаванням позицій. В фінальному рішенні ми відмовились від вручну введених полів `productX`, `qtyX`, а зробили динамічне додавання рядків через JavaScript. Використано підхід: в шаблоні є прихований рядок-шаблон, який клонуються при натисканні «Додати товар». Дропдаун з товарами відфільтровано за обраним постачальником (щоб показувати тільки ті товари, які даний постачальник може постачати – або навпаки, дозволити будь-який, якщо постачальник зможе, але ми цього не знаємо). Цей момент вирішено так: коли менеджер обирає постачальника у формі, через JS генерується список товарів = `постачальник.product_set` (з таблиці `ProductSupplier`).

На сервері перевіряється, щоб не було порожнього замовлення, щоб кількості  $> 0$ . Якщо все гаразд – зберігається. Встановлюється статус «NEW» (нове, не відправлене). Номер PO генерується. У прототипі ми брали ID, але тепер вирішили вести окремий лічильник – зроблено модель `Settings` або `Sequence`, де є поле `next_po_number`. Перед збереженням PO, беремо цей номер, формуємо рядок (наприклад, `f"PO-{next_po_number:05d}"`) і оновлюємо `next_po_number += 1` для наступного. Це гарантує послідовність навіть якщо записи видаляються.

Для статусів `Draft` або `New` дозволено редагувати (наприклад, менеджер може скоригувати кількість або додати товар перед відправкою). Реалізовано окремий шаблон `po_detail.html`, де, якщо статус `Draft/New`, поля таблиці редаговані (`input` для `quantity`, або `dropdown` для товару якщо треба замінити), і кнопка «Зберегти зміни».

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 42   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

Кнопка «Надіслати» активна для New або Draft стану. Статус змінюється на «Sent»; генерується PDF-документ (комерційне замовлення) – реалізовано з використанням шаблону HTML ро\_pdf.html та бібліотеки WeasyPrint (вона добре підтримує укр. мову та HTML/CSS в PDF). У цьому PDF – логотип компанії (можна завантажити в налаштуваннях), реквізити постачальника, дата, номер, таблиця з товарами (назва, кількість, ціна, сума), підпис менеджера (опційно). Сформований PDF зберігається тимчасово і відправляється email-ом. Для email використано EmailMessage з Django:

Після успішної відправки замовлення постачальнику у системі фіксується дата відправлення шляхом заповнення поля sent\\_date у сутності Purchase Order. Важливою особливістю реалізації є налаштування логіки роботи автоматичних закупівель: автоматично створені замовлення із статусом Draft не надсилаються постачальнику до того часу, поки менеджер не перегляне їх та не виконає дію «Надіслати». Такий підхід дозволяє уникнути ситуацій, коли система може здійснити замовлення без відома відповідальної особи. Проте у параметрах конфігурації системи (Settings) передбачено опцію AUTO\\_SEND\\_ORDERS, яка дає змогу в майбутньому активувати повністю автоматичне відправлення замовлень для перевірених постачальників.

Процедура скасування замовлення також має визначену логіку. Кнопка «Скасувати» доступна доти, доки замовлення не було відправлено постачальнику, або якщо сам постачальник відмовився його виконувати. У разі активації цієї функції статус замовлення змінюється на «Cancelled». Якщо замовлення було створене автоматично, його можна скасувати, наприклад, для об'єднання з іншим замовленням. Якщо ж замовлення вже було відправлене, то скасування супроводжується додатковим бізнес-процесом: система пропонує менеджеру надіслати постачальнику електронне повідомлення про скасування на основі відповідного шаблону.

Генерація супровідних документів також інтегрована у функціонал системи. Окрім створення PDF-версії замовлення, передбачена можливість формування

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 43   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

друкованих документів для обліку отримання товарів, зокрема складських прибуткових накладних, що реалізується у відповідному модулі обліку поставок.

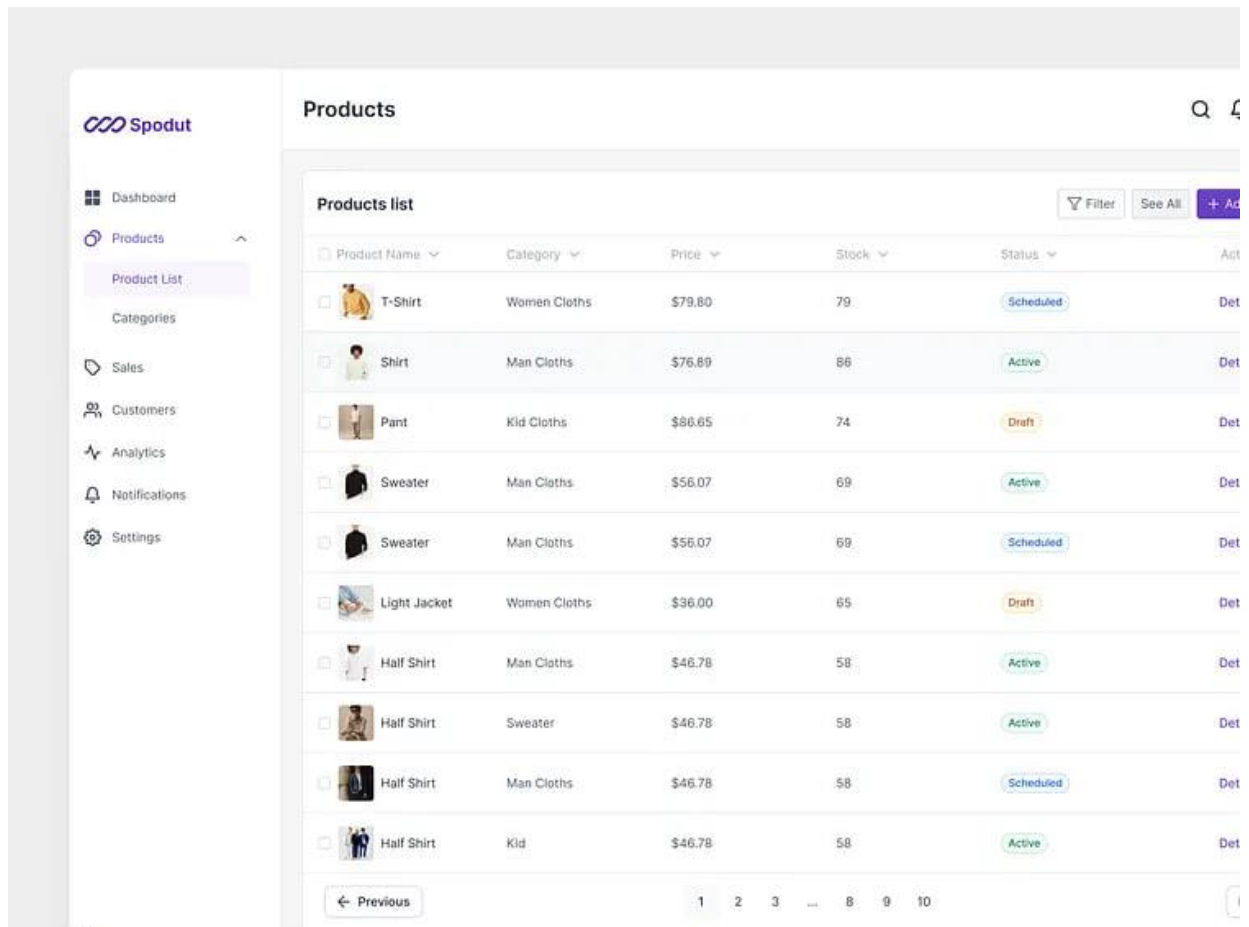


Рисунок 3.1 – Прототип інтерфейсу адміністратора системи

Зліва – бокове меню з розділами (Dashboard, Продукти, Категорії, Продажі, Користувачі, Аналітика, Налаштування). Справа – таблиця зі списком товарів: назва, категорія, ціна, кількість на складі, статус. Адмін може додавати, редагувати або видаляти товари через відповідні кнопки.

Модуль управління пропозиціями (Request for Quotation, RFQ) у системі реалізовано як опціональний компонент, який активується у разі, якщо бізнес-процеси компанії передбачають проведення тендерів або запитів цінових пропозицій. Для цього створено модель RequestForQuotation із ключовими полями: постачальник (у спрощеній реалізації кожен RFQ стосується одного постачальника, хоча можлива і розсилка кільком адресатам), перелік позицій (структурно

аналогічний до PurchaseOrderDetail), а також статус, що може набувати значень Draft, Sent, Answered, Approved.

Користувацький інтерфейс модуля організований у вигляді сторінки «Quotations», подібної до сторінки керування замовленнями, але з меншою кількістю полів і функцій. Менеджер має змогу створити RFQ, обрати постачальника, вказати товари та їх кількість. Після натискання кнопки «Send RFQ» формується електронний лист: у його тілі або у вигляді прикріпленого PDF міститься таблиця із запитом. Відповіді постачальників надходять на email, після чого менеджер вручну фіксує отримані ціни у системі (для цього у RFQDetail передбачено відповідне поле). Після надходження відповіді статус RFQ змінюється на Answered.

На наступному етапі менеджер порівнює пропозиції різних постачальників і може обрати оптимальну, після чого за допомогою функції «Create PO from this RFQ» генерується закупівельне замовлення зі статусом New, причому товари та погоджені ціни переносяться автоматично. У чинній реалізації системи функціонал RFQ підтримується частково: можна створювати й відправляти пропозиції, але не реалізовано повної підтримки багатопостачальникових тендерів. Водночас цей модуль може бути відключений у налаштуваннях (опція RFQ\_ENABLED) і не впливає на основну логіку закупівель за відсутності відповідної потреби у бізнес-процесі.

Модуль управління поставками (Incoming Shipments) забезпечує реєстрацію та контроль фактичного надходження товарів. Коли замовлення має статус Confirmed, система очікує фізичної поставки. В інтерфейсі, для замовлень зі статусами Sent або Confirmed, передбачено кнопку «Надійшло», яка відкриває форму для реєстрації отримання товару. Якщо вся партія отримана в повному обсязі, менеджеру достатньо підтвердити дату надходження та натиснути «Прийняти все». Після цього система створює запис Shipment із відповідною датою, змінює статус поставки на Completed, оновлює кількість отриманого товару

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 45   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

по кожній позиції до замовленої кількості, змінює статус замовлення на Completed, а також оновлює запаси товарів у системі.

У разі часткової поставки у формі приймання менеджер зазначає фактичні кількості отриманого по кожному товару. Після збереження інформації створюється Shipment із деталями поставки, оновлюються відповідні записи про отриману кількість, а статус замовлення змінюється на «Partially Received». При цьому система зберігає інформацію про те, скільки одиниць ще очікується (обраховується як різниця між замовленою та отриманою кількістю).

За наявності декількох часткових поставок система дозволяє переглядати історію отримань у межах кожного замовлення, а кнопка реєстрації нової поставки залишається доступною до моменту, коли всі позиції будуть повністю виконані. Після реєстрації кожної поставки менеджер може ввести номер накладної, додати скан-копію документа (Shipment містить поле для завантаження файлу), який зберігається на сервері та доступний для перегляду.

Ключовим аспектом інтеграції є оперативне оновлення залишків у WooCommerce. Після реєстрації кожної поставки система відразу коригує відповідні дані у магазині, оновлюючи поточні залишки товарів через WooCommerce API. Такий підхід забезпечує максимально точне відображення наявності товару для клієнтів.

Після виконання всіх позицій замовлення система змінює статус Purchase Order на Completed, фіксує дату завершення та переносить замовлення до архіву. Для зручності в інтерфейсі передбачено перемикання між активними, завершеними та скасованими замовленнями.

Додатково, система зберігає повний журнал дій із зазначенням часу, відповідального користувача та деталей кожної операції – як щодо поставок, так і щодо змін статусів замовлення. Це дозволяє забезпечити прозорість усіх процесів та швидко відстежувати історію будь-яких змін у разі виникнення спірних ситуацій або необхідності аудиту.

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 46   |

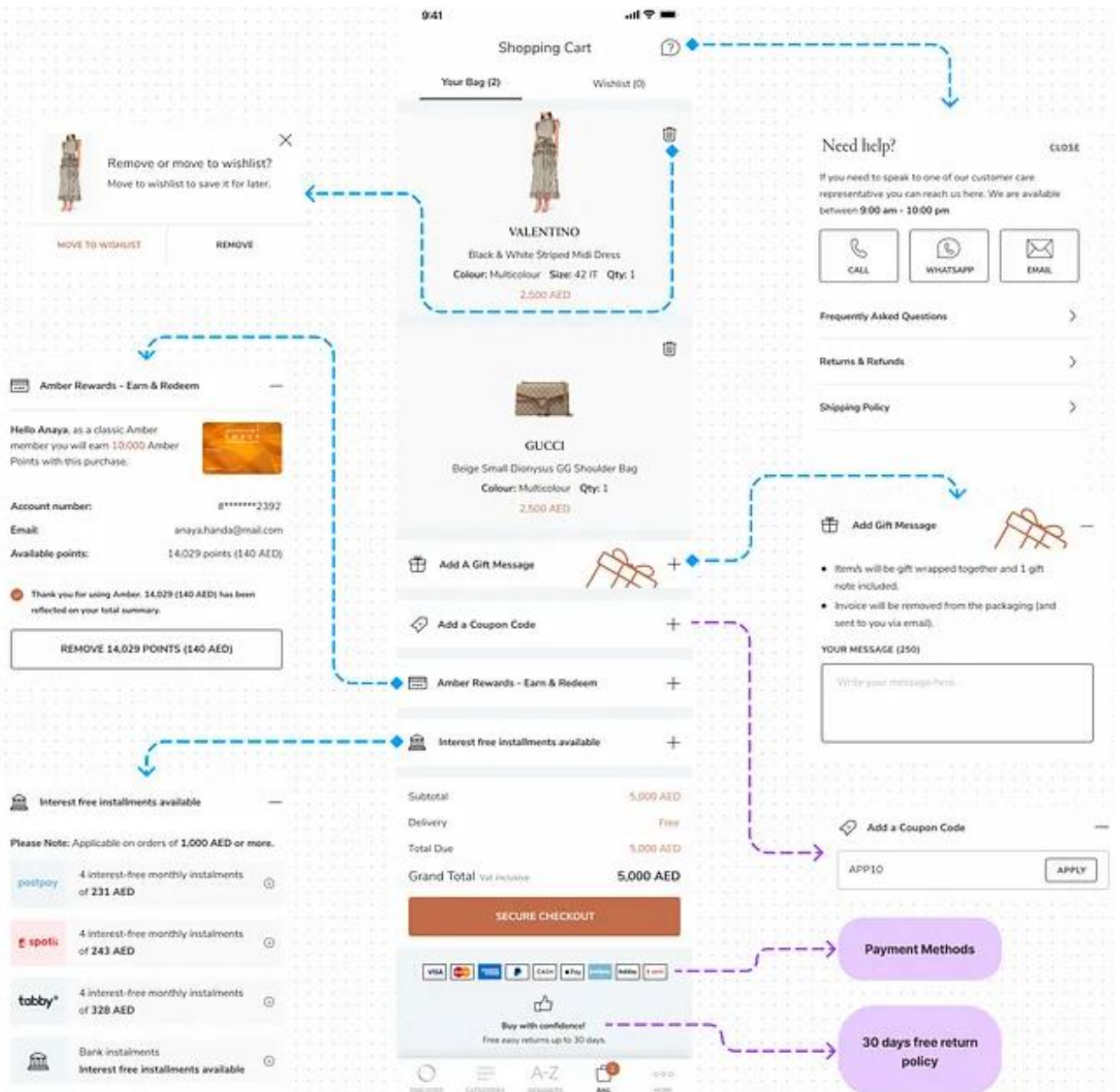


Рисунок 3.2 – UI мобільного кошика покупок (Shopping Cart) інтернет-магазину

Покупець бачить список обраних товарів (Valentino, Gucci) з можливістю змінити кількість, видалити позицію чи додати знижковий код. Також відображено підсумкову вартість (Grand Total) і кнопку «Secure Checkout» для переходу до оформлення замовлення. Нижче – інформація про способи оплати та політику повернення.

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |

Взаємодія з користувачами під час реалізації: Під час впровадження розробник тісно співпрацював з майбутніми користувачами (менеджерами закупівель), демонструючи окремі модулі і отримуючи фідбек. Це дозволило внести зміни до логіки: наприклад, вони попросили, щоб при створенні замовлення система одразу підказувала, скільки замовити.

Сумісність з WooCommerce – треба було врахувати, що WooCommerce API має ліміти швидкості: не варто 100 продуктів оновлювати за раз, краще групувати чи робити паузи. Ми знайшли, що WooCommerce API може оновлювати декілька продуктів одним запитом (batch update). Тому реалізували метод `update_products_stock(batch_data)` який шле до `/products/batch` масив об'єктів. Це значно скоротило час синхронізації при масовому оновленні.

### 3.3 Тестування системи на бізнес-сценаріях

Після завершення розробки основного функціоналу інформаційної системи управління закупівлями було проведено ретельне тестування в умовах, максимально наближених до реальної експлуатації. Метою тестування було не лише виявлення можливих помилок у коді чи логіці, але й перевірка коректності роботи системи в різноманітних бізнес-сценаріях, оцінка узгодженості даних із зовнішньою платформою WooCommerce, а також аналіз зручності інтерфейсу для кінцевих користувачів, включаючи менеджерів із закупівель і адміністраторів. Тестування проводилося в два основні етапи: автоматизоване модульне та інтеграційне тестування, яке використовувало спеціалізовані інструменти для перевірки окремих компонентів і їх взаємодії, а також ручне тестування, яке імітувало реальні бізнес-процеси за участю користувачів. Окрім цього, було проведено додаткові перевірки продуктивності, безпеки та зручності інтерфейсу, щоб переконатися, що система відповідає всім вимогам і готова до дослідного впровадження. У цьому розділі детально описано процес тестування, виявлені проблеми та їх вирішення, результати взаємодії з користувачами, а також оцінку

|     |      |          |        |      |  |                            |      |
|-----|------|----------|--------|------|--|----------------------------|------|
|     |      |          |        |      |  | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |  |                            | 48   |

ефективності системи на основі кількісних і якісних показників, отриманих під час тестового використання та моделювання.

Автоматизоване модульне тестування було першим етапом, спрямованим на перевірку окремих компонентів системи на рівні коду. Для цього використовувалася бібліотека Django TestCase, яка дозволяла створювати ізольовану тестову базу даних у пам'яті, що значно прискорювало процес тестування та забезпечувало незалежність від реальних даних. Наприклад, було протестовано метод `Product.get_preferred_supplier()`, який мав визначати постачальника з найвищим пріоритетом на основі заданих умов, таких як наявність товару, ціна чи терміни поставки. Тести підтвердили, що метод коректно обирає постачальника навіть у складних випадках, коли кілька постачальників мають однакові пріоритети, але відрізняються за іншими параметрами. Аналогічно, метод `PurchaseOrder.is_fully_received()` перевірявся для забезпечення правильної логіки: він мав повертати `True` лише тоді, коли всі отримані кількості товарів у замовленні відповідали замовленню. Для цього створювалися тестові замовлення з різними комбінаціями отриманих і замовлених кількостей, включаючи часткові поставки. Окремо тестувалася генерація унікальних номерів замовлень на закупівлю (PO). Цей функціонал мав забезпечувати створення номерів у заданому форматі (наприклад, PO-YYYYMMDD-NNN) і гарантувати їх унікальність навіть при одночасних викликах кількома користувачами. Тести показали, що генерація працює коректно, а формат зберігається за будь-яких умов.

Інтеграційне тестування з WooCommerce API вимагало особливого підходу, оскільки пряме використання реального API в тестовому середовищі було неможливим через обмеження доступу та потенційні витрати. Для цього застосовувався принцип мокінгу з використанням бібліотек `responses` та `unittest.mock`, які дозволяли імітувати відповіді API без реальних HTTP-запитів. Наприклад, було створено сценарій, у якому API повертає дані про три продукти з різними значеннями `stock_quantity` (кількість на складі). Функція `sync_product_stock()` мала коректно оновлювати відповідні записи в полі

|     |      |          |        |      |  |                            |      |
|-----|------|----------|--------|------|--|----------------------------|------|
|     |      |          |        |      |  | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |  |                            | 49   |

Product.current\_stock у базі даних системи. Тести підтвердили, що оновлення відбувається правильно, а запаси в системі синхронізуються з даними, отриманими від API. Окремо перевірялася обробка помилок: якщо API повертає код 500 (внутрішня помилка сервера) або виникає таймаут, система мала зафіксувати проблему в логах і продовжити роботу без збою. Для цього було реалізовано механізм обробки винятків із можливістю повторних спроб через певний інтервал часу. Тести показали, що система коректно логує помилки, наприклад, записуючи повідомлення типу «Failed to connect to WooCommerce API» із зазначенням часу та деталей, і не припиняє роботу навіть при тривалих перебоях зв'язку. Додатково було протестовано сценарії з частковими відповідями API, коли, наприклад, повертаються дані лише про частину продуктів. У таких випадках система оновлювала лише доступні записи, а для відсутніх даних створювала попередження для адміністратора.

Тестування форм і валідації проводилося за допомогою вбудованого HTTP-клієнта Django (Client), який симулював дії користувача через HTTP-запити. Наприклад, перевірялася форма створення замовлення на закупівлю: при спробі відправити форму без деталей (наприклад, без вказання товарів чи кількості) система повертала повідомлення про помилку з чітким поясненням, яке поле заповнено некоректно. Якщо користувач вказував нульову кількість товару, форма також видавала помилку у відповідному полі, не дозволяючи зберегти невалідні дані. У разі коректного заповнення форми після POST-запиту відбувався редирект на сторінку підтвердження, а в базі даних створювалося нове замовлення з усіма очікуваними даними, включаючи номер, дату та статус. Окремо тестувалася форма отримання товару: якщо користувач помилково вказував кількість, що перевищує замовлену (наприклад, 120 одиниць при замовленні 100), система видавала повідомлення «Отримана кількість не може перевищувати замовлену», що було реалізовано у відповідній view. Ця перевірка виявилася важливою, оскільки користувачі під час ручного тестування зазначили, що такі помилки можуть виникати через неуважність. Тести також охоплювали сценарії з частковим

|     |      |          |        |      |  |                            |      |
|-----|------|----------|--------|------|--|----------------------------|------|
|     |      |          |        |      |  | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |  |                            | 50   |

отриманням товару, коли, наприклад, лише частина замовлення була доставлена, і система мала коректно оновлювати статус замовлення та кількість на складі.

Безпекові тести були спрямовані на перевірку захисту системи від несанкціонованого доступу та неправильного використання. Наприклад, спроба доступу до URL /orders/ без авторизації автоматично перенаправляла користувача на сторінку /login/, що відповідало вимогам безпеки. Також перевірялася поведінка системи при спробі користувача з обмеженими правами (наприклад, менеджера без адміністративного доступу) виконати дії, доступні лише адміністратору, такі як видалення замовлення. У поточній реалізації системи всі авторизовані користувачі мали однакові права, тому ці тести були відносно простими, але підтвердили коректність базового захисту. Додатково перевірялися сценарії з некоректними URL-запитами, наприклад, введення неіснуючого ідентифікатора замовлення, що призводило до повернення сторінки 404 без витоку даних. Ці тести допомогли переконатися, що система стійка до типових спроб несанкціонованого доступу.

Результати автоматизованих тестів показали, що більшість перевірок пройдено успішно з першого разу, хоча було виявлено кілька незначних проблем, які потребували доопрацювання. Наприклад, при автоматичному створенні замовлення поле PurchaseOrder.status помилково встановлювалося як NEW замість Draft, що могло заплутати користувачів. Проблему усунули, додавши параметр, який явно вказує статус Draft при створенні. Інша проблема стосувалася методу is\_fully\_received(): якщо замовлення не містило деталей (ситуація, яка теоретично не мала виникати, але могла статися через помилку), метод повертав True, оскільки сума отриманого дорівнювала нулю. Це вважалося неправильним, оскільки замовлення без деталей не може вважатися виконаним. Логіку змінили так, щоб у таких випадках метод повертав False, а порожні замовлення автоматично позначалися для видалення. Ще одна проблема була пов'язана з валідацією форми постачальника: система дозволяла створювати двох постачальників з однаковими назвами, що могло викликати плутанину. Користувачі під час тестування зазначили, що хотіли б отримувати попередження про неунікальні назви. У форму

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 51   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

SupplierForm додали перевірку унікальності через метод `clean_name`, який порівнює назву з уже існуючими записами в базі даних.

Ручне тестування бізнес-сценаріїв проводилося спільно з групою користувачів, які імітували реальні процеси, такі як створення нових замовлень, їх редагування, скасування чи відстеження статусу поставок. Особлива увага приділялася інтеграції з WooCommerce: на копії реального магазину перевірялася синхронізація даних. Наприклад, після отримання товару його статус у фронт-енді магазину оновлювався до «в наявності» із затримкою приблизно 2 секунди, що було викликано часом виконання HTTP-запиту до WooCommerce. Користувачі визнали цю затримку прийнятною, оскільки вона не впливала на операційну роботу. Тестування продуктивності інтерфейсу показало хороші результати: сторінка зі списком 100 товарів завантажувалася за 0.2 секунди, а сторінка з 50 замовленнями - за 0.3 секунди, що відповідало очікуванням навіть на тестовому ноутбучі з обмеженими ресурсами. Okремо перевірялася стійкість системи до некоректного вводу та перебоїв зв'язку з WooCommerce. Наприклад, при відключенні API на хвилину система видавала чітке попередження для користувача, записувала детальну інформацію про помилку в лог («Failed to connect to WooCommerce API, timestamp: ...») і автоматично планувала повторну спробу синхронізації через заданий інтервал. Ці тести підтвердили надійність системи в нестандартних ситуаціях.

Усі виявлені під час тестування проблеми були незначними і швидко виправлені. Наприклад, окрім згаданих виправлень статусу замовлення, логіки `is_fully_received()` і валідації назв постачальників, було доопрацьовано повідомлення про помилки у формах, щоб вони були більш інформативними для користувачів. Наприклад, замість загального «Некоректні дані» форма отримання товару тепер чітко вказує, яке поле викликає проблему, наприклад, «Кількість перевищує замовлену на N одиниць». Критичних помилок чи збоїв під час сценарного тестування не виявлено, що свідчить про міцну основу, закладену модульними тестами. Система коректно генерує пропозиції щодо закупівель при

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 52   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

зниженні запасів, дозволяє менеджерам гнучко керувати замовленнями, включаючи їх редагування, об'єднання чи скасування, а також відстежує поставки та оновлює запаси як у власній базі даних, так і на платформі WooCommerce. Інтерфейс виявився зрозумілим: користувачі зазначили, що більшість дій відповідає їхнім очікуванням, а освоєння системи зайняло лише один-два дні завдяки знайомим термінам і логічній структурі.

Оцінка ефективності системи проводилася за кількісними та якісними показниками на основі результатів тестового використання та моделювання бізнес-процесів. Одним із ключових кількісних показників було скорочення часу реагування на низькі запаси. Раніше менеджери виявляли нестачу товару із затримкою, оскільки перевіряли залишки вручну, зазвичай раз на день. У тестовому сценарії, коли запаси товару падали нижче порогу о 14:00 після кількох продажів, система автоматично генерувала пропозицію замовлення о 14:05, а менеджер відправляв його постачальнику до 15:00. Без системи проблему могли помітити лише наступного ранку, що вказує на скорочення часу реагування з годин чи навіть днів до хвилин чи годин – потенційне покращення на 80-90%. Це значно знижує ризик тривалої відсутності товару в продажі, що могло б призвести до втрати клієнтів.

Інший важливий показник – зменшення випадків відсутності товару (stockout). Моделювання на основі історичних даних для п'яти товарів показало, що періоди відсутності скорочуються на 30-40% завдяки своєчасному ініціюванню поповнення. Наприклад, товар, який раніше був недоступний 10 днів на місяць через затримки в реагуванні, із системою прогнозовано буде відсутній лише 6-7 днів, оскільки затримка на етапі виявлення усунута, хоча тривалість поставки від постачальника залишається обмежуючим фактором. Система також сприяє оптимізації запасів, зменшуючи надлишки, які раніше виникали через замовлення «із запасом». Моделювання показало, що система дозволяє підтримувати менший середній запас: наприклад, замість коливань від 100 до 0 одиниць (при замовленні раз на місяць) система тримає приблизно 30 одиниць, поповнюючи їх частіше. Це

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 53   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

зменшує середній запас на 20-30%, що вивільняє оборотні кошти, які раніше були заморожені в надлишкових запасах.

Продуктивність праці менеджерів також значно зросла. Раніше два менеджери витрачали кілька годин на тиждень на рутинні задачі, такі як перевірка залишків, оформлення документів у Excel чи Word і відстеження статусів поставок у окремих журналах. Система автоматизувала ці процеси: перевірка залишків тепер займає близько години на тиждень, оскільки менеджери лише переглядають автоматично згенеровані пропозиції; оформлення документів економить приблизно 30 хвилин на кожне замовлення завдяки автоматичній генерації PDF із правильним форматуванням; відстеження поставок спрощено, оскільки система сама оновлює статуси та відображає їх у централізованому інтерфейсі. Загалом це зменшує трудовитрати на операційні закупівлі приблизно на 30%, дозволяючи менеджерам приділяти більше часу стратегічним задачам, таким як аналіз ринку чи переговори з новими постачальниками.

Якісні переваги системи виявилися не менш значущими. Прозорість і контроль закупівель значно покращилися: раніше інформація про замовлення зберігалася в листуванні, особистих нотатках чи таблицях менеджерів, що ускладнювало швидкий доступ до даних. Тепер керівництво може в будь-який момент переглянути стан закупівель, включаючи активні замовлення, затримки чи проблеми з постачальниками. Це полегшує прийняття рішень, наприклад, про зміну постачальника, якщо система показує, що він тричі поспіль затримував поставки. Система також забезпечує повне трасування та аудит: кожне замовлення має історію дій, включаючи інформацію про те, хто і коли його створив, відредагував чи відправив постачальнику. Це знижує ризики непорозумінь: наприклад, якщо постачальник стверджує, що не отримав замовлення, можна перевірити лог відправки email або ідентифікатор повідомлення. Гнучкість системи дозволяє масштабувати бізнес: вона підтримує більший асортимент і обсяг продажів без пропорційного збільшення штату, що є важливою перевагою для компаній, які планують розширення.

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 54   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

Користувачі, які брали участь у тестуванні, високо оцінили зручність інтерфейсу. Вони зазначили, що всі ключові функції доступні «під рукою», а необхідність відкривати кілька файлів Excel для аналізу залишків відпала. Особливо позитивно було оцінено автоматичне створення PDF-документів для замовлень, яке усуває помилки форматування, що раніше траплялися при ручному заповненні. Освоєння системи зайняло один-два дні, що свідчить про вдалий дизайн інтерфейсу, який використовує знайомі терміни та логіку, схожу на попередні процеси користувачів. Водночас користувачі висловили побажання щодо додаткових функцій: наприклад, розробки мобільної версії чи додатка для зручного доступу до системи в дорозі. Поточний веб-інтерфейс, побудований на основі Bootstrap, уже є адаптивним і дозволяє працювати зі смартфона, хоча таблиці іноді доводиться скролити. Також було запропоновано додати календар, який відображав би планові поставки на тиждень чи місяць, що полегшило б планування. Ці ідеї були зафіксовані для можливого доопрацювання в майбутніх версіях.

Тестування підтвердило, що система повністю відповідає вимогам, забезпечуючи коректне управління закупівлями, надійну синхронізацію з WooCommerce, стійкість до помилок і зручність для користувачів. Вона своєчасно реагує на зниження запасів, дозволяє гнучко керувати замовленнями, відстежує поставки та оновлює дані в реальному часі. Відсутність критичних помилок і швидке виправлення незначних проблем свідчать про готовність системи до дослідного впровадження.

### 3.4 Висновки до третього розділу

У даному розділі було здійснено безпосередню реалізацію інформаційної системи управління закупівлями відповідно до розробленої архітектури та функціональних вимог. Створено повнофункціональний веб-застосунок на базі фреймворку Django, що включає модулі керування постачальниками, товарами,

|     |      |          |        |      |                            |  |  |  |      |
|-----|------|----------|--------|------|----------------------------|--|--|--|------|
|     |      |          |        |      |                            |  |  |  | Арк. |
|     |      |          |        |      |                            |  |  |  | 55   |
| Зм. | Арк. | № докум. | Підпис | Дата | КВРІСТ 2101029.21.01.02 ПЗ |  |  |  |      |

замовленнями на закупівлю, поставками та інтеграцію з платформою електронної комерції WooCommerce.

Було розроблено базу даних із реляційною структурою, що дозволяє зберігати ключову інформацію про закупівельні процеси та забезпечує узгодженість між модулями системи. Реалізовано REST API-зв'язок з WooCommerce, який дозволяє оновлювати рівень запасів товарів у режимі реального часу на основі внутрішніх процесів постачання. Крім того, впроваджено базові сценарії тестування системи, що підтвердили її функціональність і здатність реагувати на типові бізнес-процеси.

У межах реалізації прототипу також було передбачено можливість подальшого розширення системи – зокрема, інтеграцію з сенсорними пристроями для побудови кіберфізичної моделі управління запасами. Результати практичної реалізації засвідчили доцільність обраної архітектури, зручність інтерфейсу та ефективність розподілу логіки між підсистемами.

Таким чином, створений програмний прототип є придатною основою для впровадження повноцінної інформаційної системи в середовищі електронної комерції з можливістю масштабування та подальшого розвитку.

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 56   |

## ВИСНОВКИ

У дипломній роботі «Інформаційна система управління закупівлями в сфері електронної комерції» комплексно вирішено задачу створення сучасного програмного рішення для автоматизації процесів постачання в умовах динамічного онлайн-бізнесу. У першому розділі проведено системний аналіз предметної області: досліджено особливості закупівель у сфері e-commerce, виявлено ключові проблеми – нестабільність попиту, необхідність оперативної реакції на зміни запасів, складність інтеграції з платформами онлайн-продажів.

У другому розділі розроблено архітектуру системи, орієнтовану на автономний модуль з інтеграцією через REST API з платформою WooCommerce. Запропоновано багатoshарову трирівневу архітектуру веб-додатку, що забезпечує гнучкість, масштабованість і безпеку. Сформовано структуру бази даних, описано ключові модулі системи: управління постачальниками, товарами, закупівельними замовленнями, поставками, аналітикою. Вибрано оптимальний стек технологій на основі Python (Django), MySQL, Bootstrap та WooCommerce API.

У третьому розділі реалізовано функціональний прототип системи: створено веб-застосунок із базовим інтерфейсом для менеджера закупівель, впроваджено механізми створення замовлень, обліку поставок та синхронізації запасів з WooCommerce. Проведено тестування на бізнес-сценаріях, яке підтвердило працездатність системи, її здатність скорочувати час реагування, знижувати ризики «out of stock» та оптимізувати взаємодію з постачальниками. У межах реалізації також закладено потенціал масштабування та доповнення IoT-компонентами для побудови кіберфізичного контролю запасів.

Таким чином, розроблена інформаційна система є ефективним інструментом для підвищення продуктивності закупівельної діяльності в e-commerce, сприяє прозорості, контролю та цифровій трансформації бізнес-процесів, що в майбутньому дозволить покращити сервіс для кінцевих клієнтів і зміцнити конкурентні переваги компанії.

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 57   |

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Chan A. P. C., Owusu E. K. Evolution of Electronic Procurement: Contemporary Review of Adoption and Implementation Strategies. *Buildings*. 2022. Vol. 12(2). P. 198.
2. Dudić Ž., Vrhovac V., Vulanović S., Dakić D., Erdeji I., Perović V. A Risk-Aware Approach to Digital Procurement Transformation. *Sustainability*. 2024. Vol. 16(3). P. 1283.
3. Nicoletti B. Procurement 4.0 and the Fourth Industrial Revolution: The Opportunities and Challenges of a Digital World. Berlin: Springer, 2020. 288 p.
4. Nicoletti B. Agile Procurement. Vol. II: Designing and Implementing a Digital Transformation. Cham: Springer, 2018. 281 p.
5. Glas A. H., Kleemann F. C. The Impact of Industry 4.0 on Procurement and Supply Management: A Conceptual and Qualitative Analysis. *International Journal of Business and Management Invention*. 2016. Vol. 5. P. 55-66.
6. Yevu S. K., Yu A. T. W., Darko A. Digitalization of Construction Supply Chain and Procurement in the Built Environment: Emerging Technologies and Opportunities for Sustainable Processes. *Journal of Cleaner Production*. 2021. Vol. 322. P. 129093.
7. Bauer W., Dorn J. System Design of E-Markets for IPSS Purchasing. *Procedia CIRP*. 2017. Vol. 64. P. 211-216.
8. Brandon-Jones A., Kauppi K. Examining the antecedents of the technology acceptance model within e-procurement. *International Journal of Operations & Production Management*. 2018. Vol. 38(1). P. 22-42.
9. Croom S., Johnston R. E-service: Enhancing internal customer service through e-procurement. *International Journal of Service Industry Management*. 2013. Vol. 14(5). P. 539-555.
10. Croom S., Brandon-Jones A. Impact of e-procurement: Experiences from implementation in the UK public sector. *Journal of Purchasing and Supply Management*. 2017. Vol. 13(4). P. 294-303.

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 58   |

11. Gunasekaran A., Ngai E. W. T. Adoption of e-procurement in Hong Kong: An empirical research. *International Journal of Production Economics*. 2018. Vol. 113(1). P. 159-175.
12. Tanner C., Wölflé R. Current Trends and Challenges in Electronic Procurement: An Empirical Study. *Electronic Markets*. 2018. Vol. 18(6). P. 6-18.
13. Presutti W. D. Supply management and e-procurement: creating value added in the supply chain. *Industrial Marketing Management*. 2013. Vol. 32(3). P. 219-226.
14. Davila A., Gupta M., Palmer R. Moving procurement systems to the Internet: The adoption and use of e-procurement technology models. *European Management Journal*. 2013. Vol. 21(1). P. 11-23.
15. Vaidya K., Sajeev A. S. M., Callender G. Critical factors that influence e-procurement adoption success in the public sector. *Journal of Public Procurement*. 2016. Vol. 6(1). P. 70-99.
16. Kim J., Shunk D. Matching indirect procurement process with different B2B e-procurement systems. *Computers in Industry*. 2014. Vol. 53(2). P. 153-164.
17. Muffatto M., Payaro A. Implementation of e-procurement and e-fulfillment processes: A comparison of cases in the motorcycle industry. *International Journal of Production Economics*. 2014. Vol. 89(3). P. 339-351.
18. Puschmann T., Alt R. Successful use of e-procurement in supply chains. *Supply Chain Management*. 2015. Vol. 10(2). P. 122-133.
19. Teo T. S. H., Lin S., Lai K. Adopters and non-adopters of e-procurement in Singapore: An empirical study. *Omega*. 2019. Vol. 37(5). P. 972-987.
20. Knudsen D. Aligning corporate strategy, procurement strategy and e-procurement tools. *International Journal of Physical Distribution & Logistics Management*. 2013. Vol. 33(8). P. 720-734.
21. Neupane A., Soar J., Vaidya K., Yong J. An empirical evaluation of the potential of public e-procurement to reduce corruption. *International Journal of Procurement Management*. 2014. Vol. 7(1). P. 94-111.

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 59   |

22. Panayiotou N. A., Gayialis S. P., Tatsiopoulou I. P. An e-procurement system for governmental purchasing. *International Journal of Production Economics*. 2014. Vol. 90(1). P. 79-102.

23. Kishor W., Teng J. T. C. Facilitators, inhibitors, and the transition to electronic procurement systems. *Communications of the ACM*. 2016. Vol. 49(12). P. 41-46.

24. Aboelmaged M. G. Predicting e-procurement adoption in a developing country: an empirical integration of TAM and TPB. *Industrial Management & Data Systems*. 2020. Vol. 110(3). P. 392-414.

25. Wyld D. C. The cloud-based procurement service model: a new era of procurement. *Journal of Enterprise Information Management*. 2021. Vol. 24(3). P. 237-250.

26. Hawking P., Stein A., Wyld D., Foster S. E-procurement: Is the ugly duckling actually a swan? *Asia Pacific Journal of Marketing and Logistics*. 2014. Vol. 16(1). P. 41-61.

27. Dhillon G., Margiono A. The role of information systems in procurement process innovation. *Information Systems Frontiers*. 2019. Vol. 21(4). P. 921-939.

28. Bof F., Previtali P. Organizational pre-conditions for e-procurement in governments: the Italian experience. *Electronic Government, an International Journal*. 2017. Vol. 4(4). P. 369-382.

29. Somasundaram R., Damsgaard J. Policy recommendations for electronic public procurement. *Electronic Journal of e-Government*. 2015. Vol. 3(3). P. 147-156.

30. Carter P. L., Kaufmann L., Beall S., Carter C. R., Hendrick T., Petersen K. J. eProcurement: Myth or reality? *Journal of Supply Chain Management*. 2014. Vol. 40(2). P. 22-28.

31. de Boer L., Harink J., Heijboer G. A conceptual model for assessing the impact of electronic procurement. *European Journal of Purchasing and Supply Management*. 2012. Vol. 8(1). P. 25-33.

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 60   |

32. Allal-Chérif O., Simón-Moya V., Cuenca Ballester A. C. Intelligent purchasing: how artificial intelligence can redefine the purchasing function. *Journal of Business Research*. 2021. Vol. 124. P. 69–76.

33. Arvidsson A. P., Jonsson P., Kaipia R. Big data in purchasing and supply management: a research agenda. *International Journal of Procurement Management*. 2021. Vol. 14, No. 2. P. 185–212.

34. Bodendorf F., Lutz M., Michelberger S., Franke J. An empirical investigation into intelligent cost analysis in purchasing. *Supply Chain Management: An International Journal*. 2021. Vol. 26, No. 6. P. 785–808.

35. Bienhaus F., Haddud A. Procurement 4.0: factors influencing the digitisation of procurement and supply chains. *Business Process Management Journal*. 2018. Vol. 24, No. 4. P. 965–984.

36. Herold S., Heller J., Rozemeijer F., Mahr D. Dynamic capabilities for digital procurement transformation: a systematic literature review. *International Journal of Physical Distribution & Logistics Management*. 2022. No. 4. P. 424–447.

37. Lysons K., Farrington B. *Purchasing and Supply Chain Management*. 8th ed. Harlow: Pearson, 2016. P. 520.

38. Prier E., McCue C. Purchasing in the 21st century: e-procurement and beyond. *The Public Manager*. 2017. Vol. 36(1). P. 70-76.

39. Basl J., Doucek P. Digital Transformation and Industry 4.0 in Purchasing and Supply Management. В кн.: *IDIMT-2019 Digitalized Economy and Society: Proceedings*. Linz: Trauner Verlag, 2019. С. 85-92.

40. Arvidsson A. P., Jonsson P., Kaipia R. Big data in purchasing and supply management: a research agenda. *International Journal of Procurement Management*. 2021. Vol. 14, No. 2. P. 185-212.

41. Bag S., Dhamija P., Gupta S., Sivarajah U. Examining the role of procurement 4.0 towards remanufacturing operations and circular economy. *Production Planning & Control*. 2021. Vol. 32, No. 16. P. 1368-1383.

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 61   |

42. Bienhaus F., Haddud A. Procurement 4.0: factors influencing the digitisation of procurement and supply chains. *Business Process Management Journal*. 2018. Vol. 24, No. 4. P. 965-984.

43. Herold S., Heller J., Rozemeijer F., Mahr D. Dynamic capabilities for digital procurement transformation: a systematic literature review. *International Journal of Physical Distribution & Logistics Management*. 2022. No. 4. P. 424-447.

44. Bals L., Schulze H., Kelly S. J. H., Stek K. Purchasing and supply management (PSM) competencies: current and future requirements. *Journal of Purchasing and Supply Management*. 2019. Vol. 25, No. 5. P. 100572.

45. Bodendorf F., Lutz M., Michelberger S., Franke J. An empirical investigation into intelligent cost analysis in purchasing. *Supply Chain Management: An International Journal*. 2021. Vol. 26, No. 6. P. 785-808.

46. Hallikas J., Immonen M., Brax S. Digitalizing procurement: the impact of data analytics on supply chain performance. *Supply Chain Management: An International Journal*. 2021. Vol. 26, No. 5. P. 629-646.

47. Kosmol T., Reimann F., Kaufmann L. You'll never walk alone: why we need a supply chain practice view on digital procurement. *Journal of Purchasing and Supply Management*. 2019. Vol. 25, No. 4. P. 100553.

48. Křenková E., Rieser K., Sato A. How software robots can facilitate the procurement process? A case study of Siemens in the Czech Republic. *Entrepreneurial Business and Economics Review*. 2021. Vol. 9, No. 3. P. 191-203.

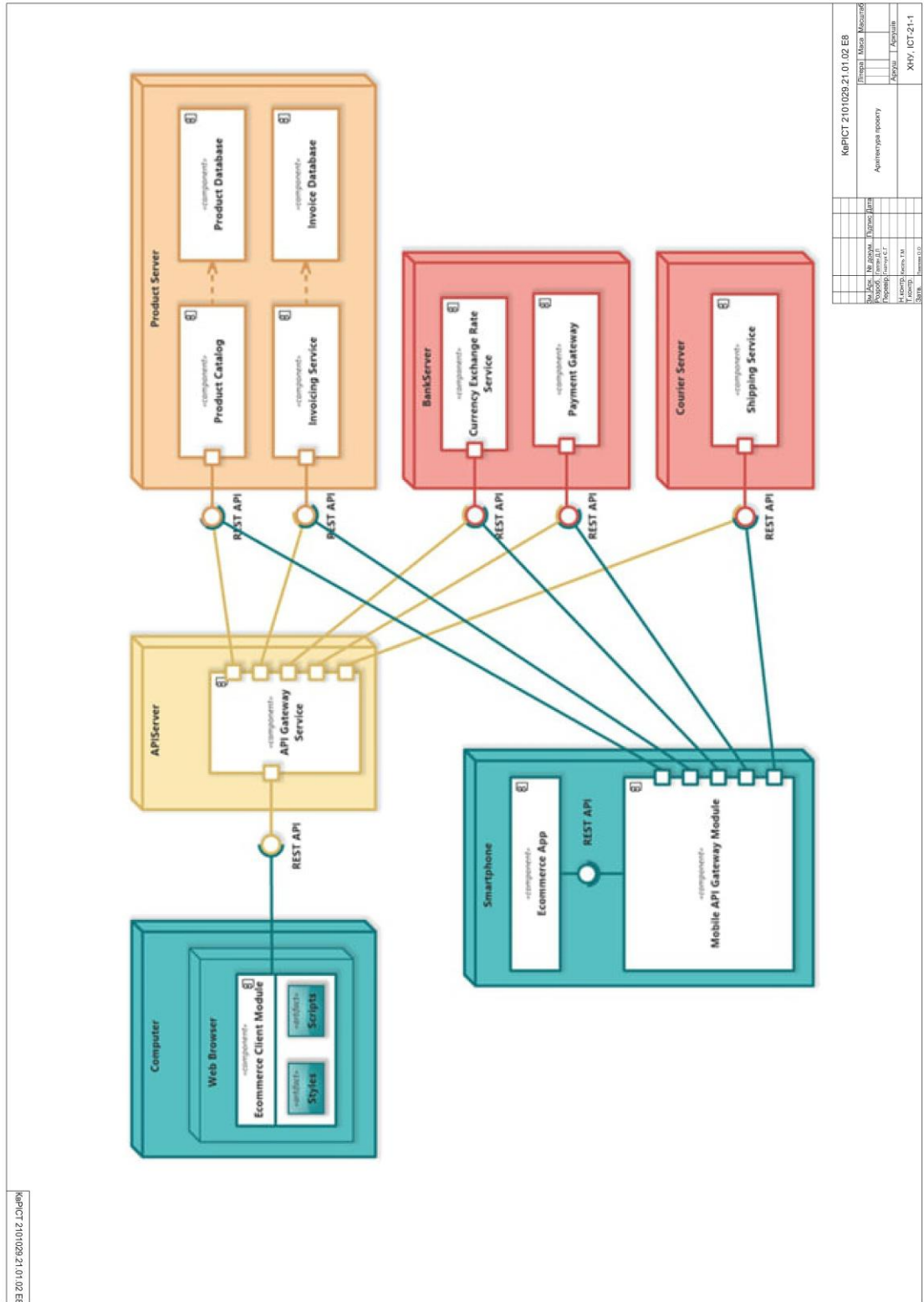
49. Legenvre H., Henke M., Ruile H. Making sense of the impact of the Internet of Things on purchasing and supply management: a tension perspective. *Journal of Purchasing and Supply Management*. 2020. Vol. 26, No. 4. P. 100596.

|     |      |          |        |      |                            |      |
|-----|------|----------|--------|------|----------------------------|------|
|     |      |          |        |      | КВРІСТ 2101029.21.01.02 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                            | 62   |

# ДОДАТОК А

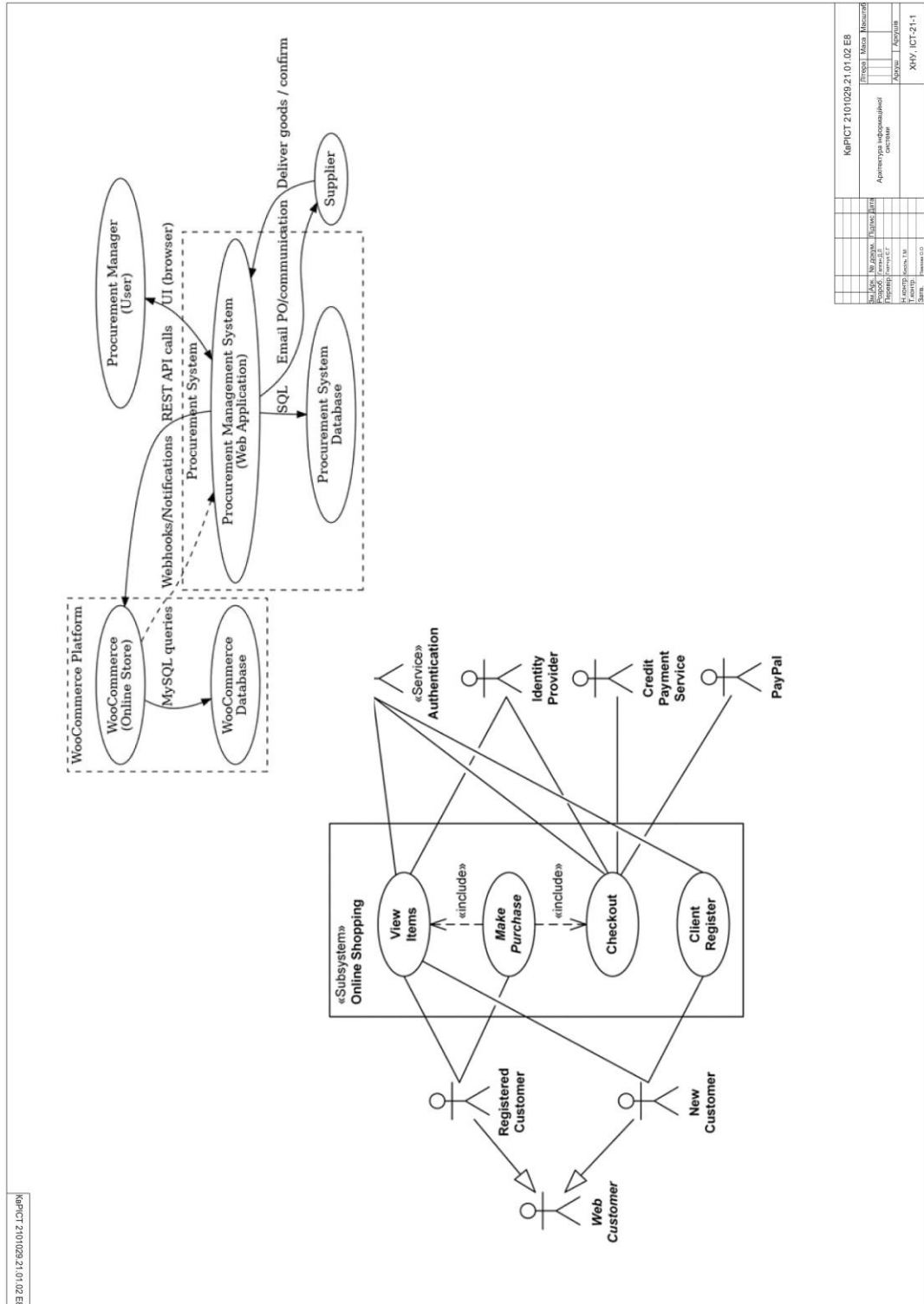
## (обов'язковий)

### КОПІЯ КРЕСЛЕННЯ «АРХІТЕКТУРА ПРОЄКТУ»



## ДОДАТОК Б (обов'язковий)

### КОПІЯ КРЕСЛЕННЯ «АРХІТЕКТУРА ІНФОРМАЦІЙНОЇ СИСТЕМИ»



|                             |                                   |
|-----------------------------|-----------------------------------|
| Картрич 2101029.21.01.02.ЕВ |                                   |
| Ім'я:                       | Мико                              |
| Прізвище:                   | Мико                              |
| Посада:                     | Менеджер                          |
| Підприємство:               | Архітектурна інформаційна система |
| Місто:                      | Київ                              |
| Держава:                    | Україна                           |
| Сторінка:                   | 1 з 1                             |
| Дата:                       | 2019.10.12                        |

# ДОДАТОК В (обов'язковий)

## КОПІЯ КРЕСЛЕННЯ «ВИГЛЯД ВІКОН ІНФОРМАЦІЙНОЇ СИСТЕМИ»

8E32010126201012.L0406X

The image displays two screenshots of the Spodut e-commerce system. The left screenshot shows a 'Products list' page with a table of items. The right screenshot shows a detailed product view for 'VALVINDO' with a flow diagram illustrating the user journey from product selection to payment.

| Product Name | Category     | Price   | Stock | Status    |
|--------------|--------------|---------|-------|-----------|
| T-Shirt      | Women Cloths | \$78.80 | 79    | Scheduled |
| Shirt        | Men Cloths   | \$78.89 | 86    | Active    |
| Pant         | Kid Cloths   | \$86.65 | 74    | Draft     |
| Sweater      | Men Cloths   | \$54.07 | 69    | Active    |
| Sweater      | Men Cloths   | \$58.07 | 89    | Scheduled |
| Light Jacket | Women Cloths | \$36.00 | 65    | Draft     |
| Half Shirt   | Men Cloths   | \$45.78 | 58    | Active    |
| Half Shirt   | Sweater      | \$45.78 | 59    | Active    |
| Half Shirt   | Men Cloths   | \$45.78 | 58    | Scheduled |
| Half Shirt   | Kid          | \$45.78 | 58    | Active    |

КРПСТ 2101029.21.01.02.ЕВ

Триггер: Маса

Місяць: \_\_\_\_\_

Варіанти відображення:  По дати,  По даним,  По назві,  По статусу

Інформаційна система: \_\_\_\_\_

Актуальні: \_\_\_\_\_

ХНУ, ІСТ-21-1

## ДОДАТОК Г

(обов'язковий)

### МОДУЛЬ «ОБЛІК ПОСТАЧАЛЬНИКІВ І ПОСТАВОК»

```
# === Модуль 1: Запуск Django-проєкту ===
# === manage.py ===
#!/usr/bin/env python

import os
import sys

def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE',
'procurement_system.settings')

    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError("Couldn't import Django.") from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()

# === Модуль 2: Конфігурація Django-проєкту ===
# === procurement_system/settings.py ===
SECRET_KEY = 'example-key'

DEBUG = True

ALLOWED_HOSTS = []

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
```

```

'django.contrib.staticfiles',
'purchases',
]
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'procurement_db',
        'USER': 'root',
        'PASSWORD': "",
        'HOST': 'localhost',
        'PORT': '3306',
    }
}
LANGUAGE_CODE = 'uk'
TIME_ZONE = 'Europe/Kyiv'
USE_I18N = True
USE_TZ = True
STATIC_URL = '/static/'
# === Модуль 3: Головний маршрутизатор проекту ===
# === procurement_system/urls.py ===
from django.contrib import admin
from django.urls import path, include
urlpatterns = [
    path('admin/', admin.site.urls),
    path("", include('purchases.urls')),
]
# === Модуль 4: Моделі даних ===
# === purchases/models.py ===
from django.db import models

```

```

class Supplier(models.Model):
    name = models.CharField(max_length=255)
    contact_info = models.TextField()
    def __str__(self):
        return self.name

class Product(models.Model):
    name = models.CharField(max_length=255)
    reorder_point = models.IntegerField()
    current_stock = models.IntegerField(default=0)
    def __str__(self):
        return self.name

class PurchaseOrder(models.Model):
    supplier = models.ForeignKey(Supplier, on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True)
    status = models.CharField(max_length=50, default='NEW')
    def __str__(self):
        return f"PO-{self.id}"

class PurchaseOrderDetail(models.Model):
    po = models.ForeignKey(PurchaseOrder, related_name='details',
on_delete=models.CASCADE)
    product = models.ForeignKey(Product, on_delete=models.CASCADE)
    quantity = models.IntegerField()

# === Модуль 5: Формы введения данных ===
# === purchases/forms.py ===
from django import forms
from .models import Supplier

class SupplierForm(forms.ModelForm):
    class Meta:
        model = Supplier

```

```

    fields = ['name', 'contact_info']
# === Модуль 6: Представлення та логіка обробки запитів ===
# === purchases/views.py ===
from django.shortcuts import render, get_object_or_404, redirect
from .models import Supplier, Product, PurchaseOrder, PurchaseOrderDetail
from .forms import SupplierForm
from django.contrib import messages
def supplier_list(request):
    suppliers = Supplier.objects.all()
    return render(request, 'supplier_list.html', {'suppliers': suppliers})
def receive_shipment(request, po_id):
    po = get_object_or_404(PurchaseOrder, id=po_id)
    if request.method == "POST":
        for detail in po.details.all():
            qty = int(request.POST.get(f'received_{detail.id}', 0))
            detail.product.current_stock += qty
            detail.product.save()
        po.status = 'COMPLETED'
        po.save()
        messages.success(request, "Поставка зареєстрована")
        return redirect('supplier_list')
    return render(request, 'receive_shipment.html', {'po': po})
# === Модуль 7: Маршрутизація модуля purchases ===
# === purchases/urls.py ===
from django.urls import path
from . import views
urlpatterns = [
    path('suppliers/', views.supplier_list, name='supplier_list'),
    path('receive/<int:po_id>/', views.receive_shipment, name='receive_shipment'),

```

```

]
# === Модуль 8: Інтеграція з WooCommerce API ===
# === purchases/woocommerce_service.py ===
from woocommerce import API
from django.conf import settings

wcapl = API(
    url='https://example.com',
    consumer_key='ck_xxx',
    consumer_secret='cs_xxx',
    version='wc/v3'
)

def update_product_stock(product):
    data = {"stock_quantity": product.current_stock}
    wcapl.put(f'products/{product.id}', data)
# === Модуль 9: Шаблон списку постачальників ===
# === purchases/templates/supplier_list.html ===
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Список постачальників</title>
</head>
<body>
    <h1>Постачальники</h1>
    <ul>
        {% for supplier in suppliers %}
            <li>{{ supplier.name }} - {{ supplier.contact_info }}</li>
        {% endfor %}

```

```

    </ul>
</body>
</html>
# === Модуль 10: Шаблон форми отримання поставки ===
# === purchases/templates/receive_shipment.html ===
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Отримання поставки</title>
</head>
<body>
    <h1>Отримати поставку для {{ po }}</h1>
    <form method="post">
        {% csrf_token %}
        <ul>
            {% for detail in po.details.all %}
                <li>{{ detail.product.name }}: <input type="number" name="received_{{
detail.id }}" min="0"></li>
            {% endfor %}
        </ul>
        <button type="submit">Зберегти</button>
    </form>
</body>
</html>

```

## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Дарина ГАЛГАН

Співавтор:

Назва: Галган\_Інформаційна система управління закупівлями в сфері електронної комерції

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:6.1%

Коефіцієнт подібності 2:1.8%

Мікропробіли: 76

Заміна букв: 54

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-11 03:24:52.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2025-06-11

Дата



Доцент Андрій Нічепорук

експерт

# Anti-Plagiarism (UA) v-15.281 Educational

**The maximum coincidence with one document 1.0%**

Dictionary check: en\_US, ru\_RU, ua\_UA. **Errors in the documents: 10%**

|  |          |         |                           |         |
|--|----------|---------|---------------------------|---------|
| ID: 244909<br>Title: БКР Інформаційна система управління закупівлями в сфері електронної комерції<br>Added in a DB: 2025-06-11<br>Authors: Дарина ГАЛГАН<br>Heads: Єлизавета ГНАТЧУК<br>Consultants:<br>Opponents: | Document |         | Sum coincidence on the DB |         |
|  | Symbols  | Lexemes | Symbols                   | Lexemes |
|  | 110858   | 799     | 2030 (2%)                 | 28 (4%) |

## Plagiarism sources

| ID | Description | Plagiarism presence in the document |         |
|----|-------------|-------------------------------------|---------|
|    |             | Symbols                             | Lexemes |
|    |             |                                     |         |

## РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломниця: Галган Дарина Леонідівна

Тема: Інформаційна система управління закупівлями в сфері електронної комерції

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг кваліфікаційної роботи:

Кількість листів креслень   3   Кількість сторінок записки   60  

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є розробка та впровадження інформаційної системи управління закупівлями для електронної комерції, що автоматизує взаємодію між внутрішніми закупівельними процесами та платформою продажу WooCommerce. У роботі реалізовано веб-застосунок на базі Django з REST API та MySQL, який може бути інтегрований із сенсорними пристроями для автоматичного моніторингу залишків. Висвітлено особливості предметної області, проведено аналіз сучасних рішень, розроблено архітектуру та реалізовано прототип системи.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено комплексний аналіз предметної області, виявлено проблеми та завдання управління закупівлями в e-commerce, виконано порівняльний аналіз існуючих рішень та обґрунтовано вибір підходу до автоматизації.

У другому розділі розроблено архітектуру інформаційної системи, детально описано модульну структуру, способи інтеграції з платформою WooCommerce та розглянуто питання безпеки і масштабованості.

У третьому розділі наведено опис реалізації програмного прототипу, розглянуто тестування системи на типових бізнес-сценаріях, проаналізовано результати

впровадження. Робота демонструє застосування сучасних технологій (Django, REST API, інтеграція з e-commerce платформами) і методів модульного проектування.

4. Позитивні сторони роботи: Актуальність теми для сучасного бізнесу, комплексний підхід до вирішення поставленої задачі, високий рівень реалізації прототипу, практична значущість результатів, використання сучасних інструментів та технологій, наявність аналітики і підтримка інтеграції з реальними платформами.

5. Негативні сторони роботи: Недостатньо уваги приділено розширенню функціоналу для масштабування під великі підприємства та питанням довгострокової підтримки і супроводу системи.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному технічному рівні.

8. Інші зауваження: \_\_\_\_\_

9. Оцінка дипломної роботи: задовільно (D 3.50)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_

*Омишко Олександр Григорович, в. пер. каф.,  
доцент каф. ІІЗ, УКУ*

“ \_\_\_\_\_ ” \_\_\_\_\_ 2025 р.

 (підпис)

Завідувачу кафедри КІПС  
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Дарини ГАЛГАН

---

ІІБ здобувача вищої освіти

ФІТ, 4 курсу, групи ІСТ-21-1

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

10.06. 2025 року



**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА  
ІНФОРМАЦІЙНИХ СИСТЕМ**

**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Назва кваліфікаційної роботи Інформаційна система управління закупівлями в сфері електронної комерції

Автор Дарина ГАЛГАН

Освітня програма Інформаційні системи та технології

Рівень вищої освіти перший (бакалаврський) рівень

Спеціальність 126– Інформаційні системи та технології

Науковий керівник: Єлизавета ГНАТЧУК д.т.н., професор, професор кафедри

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

| №   | Висновок  | Позначка про відповідність |
|-----|---|----------------------------|
| 1   | Ознаки академічного плагіату  |                            |
| 1.1 | Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.   | Відповідає                 |
| 1.2 | Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.   |                            |
| 1.3 | Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат. |                            |
| 1.4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.  |                            |
| 2   | Інші види порушень академічної доброчесності  | Не виявлено                |

**Підтвердження:**

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;

2) усі запозичення фрагментарні, або мають належним чином оформленні посилання.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 6.1% і адресується до 18 першоджерела; та системою Anti-Plagiarism складає 1%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Єлизавета ГНАТЧУК

Єлизавета ГНАТЧУК

Ольга ПАВЛОВА