

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ГРИГОРЕНКО МАРИНА СЕРГІЇВНА

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій
канд. техн. наук, доцент
_____ О. В. Зелінська
« ____ » _____ 20__ р.

ВЕБДОДАТОК УПРАВЛІННЯ СКЛАДОМ

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (бакалаврська) робота

Керівник:

Т. В. Січко, доцент кафедри
інформаційних технологій,
к. т. н., доцент

Оцінка: _____ / _____ / _____
(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____

Вінниця - 2025

АНОТАЦІЯ

Григоренко М.С. Вебдодаток управління складом. Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця 2025.

У кваліфікаційній (бакалаврській) роботі досліджено та проаналізовано поняття управління складом, виділені основні поняття баз даних та вебдодатків. За допомогою платформи Oracle APEX було розроблено вебдодаток, що дозволяє ефективно керувати складом одягу.

Ключові слова: база даних, вебдодаток, управління, склад, Oracle APEX.

69 ст. 19 рис., 49 джерел.

ABSTRACT

Hryhorenko M. Warehouse Management Web Application. Specialty 122 «Computer Science», educational program «Computer Science». Vasyl Stus Donetsk National University, Vinnytsia 2025.

In the qualification (bachelor's) work the concept of warehouse management was studied and analyzed, and the basic principles of databases and web applications were outlined. Using the Oracle APEX platform, a web application was developed to enable efficient management of a clothing warehouse.

Keywords: database, web application, management, warehouse, Oracle APEX.

69 p., 19 fig., 49 references.

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРОЦЕСУ УПРАВЛІННЯ СКЛАДОМ	8
1.1. Основні задачі управління складом.....	8
1.2. Організація баз даних у системах обліку складу	12
1.3. Принципи роботи вебдодатків управління складом.....	18
1.4. Постановка завдання	25
РОЗДІЛ 2. АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ТА ІНСТРУМЕНТІВ	28
2.1. Аналіз технологій для розробки вебдодатків управління складом	28
2.2. Обґрунтування використання Oracle APEX для розробки вебдодатку управління складом.....	38
РОЗДІЛ 3. РОЗРОБКА ВЕБДОДАТКУ ДЛЯ УПРАВЛІННЯ СКЛАДОМ.....	45
3.1. Проектування бази даних для складу одягу за допомогою Oracle Database	45
3.2. Розробка вебдодатку за допомогою Oracle APEX.....	48
3.3. Функціональні можливості вебдодатку управління складом	51
ВИСНОВКИ.....	61
СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ	63

ВСТУП

Як відомо, однією з важливих передумов успіху для будь-якого бізнесу є ефективне управління складом. Особливо це стосується виробництва, логістики та торгівлі. Основною вимогою для успіху в цих галузях є швидка та точна обробка даних, яку людина самотійно без допомоги додаткових засобів не в змозі забезпечити. Зробити це без інструментів, які чітко контролюють основні процеси, що відбуваються на складі – фактично неможливо. А це такі важливі операції, як прийом та видача товарів, формування звітностей, контроль усіх взаєморозрахунків і так далі.

Часто можна спостерігати ситуації, коли підприємство з певної причини не має чіткої системи контролю товарів. В цих випадках працівники складів змушені витрачати багато часу на пошук необхідних товарів або інформації про них тому що відсутні цифрові бази даних. У таких умовах виникають постійні проблеми із товарами і процесами – щось може бути втраченим, щось можуть недорахувати, щось пораховано двічі та інші випадки. В кінцевому результаті це може призвести до фінансових втрат, юридичних проблем, недотримання термінів поставки товару та інших питань.

Окрім того, слід враховувати людський фактор, а саме неухважність, необережність, недбалість, втомлюваність тощо. Працівники складу можуть робити помилки при введенні даних вручну, що створює подальші неточності в обліку. Без централізованої системи управління складом підприємство може мати певні складнощі у взаємодії між різними відділами. Керівники, менеджери, бухгалтерія, фінансисти, логісти, обліковці та інші працівники не мають оперативного доступу до необхідної для них актуальної інформації і це уповільнює прийняття важливих рішень.

Підприємства малого та середнього бізнесу, а також і деякі крупні фірми, мають проблеми, пов'язані із неактуальними методами керування складськими процесами. Деякі малі підприємства та ФОП досі покладаються на паперове ведення документації, а це значно ускладнює контроль за залишками товарів,

особливо в випадках, коли зростає кількість артикулів чи номенклатури товарів, і це в свою чергу вимагає досить великої витрати часу на обробку даних і ймовірність помилок зростає. При такому підході ведення складської логістики слід врахувати можливість втрати важливих документів, складність обліку та пошуку необхідних даних та необхідність дублювання інформації у різний спосіб. Деякі компанії користуються електронними таблицями для ведення обліку, проте такий підхід також має суттєві недоліки: часто дані вносяться працівником вручну, що може призвести до численних невідповідностей чи неточностей в даних, а відсутність централізованого або спільного доступу ускладнює роботу в команді.

На даний час уже є розроблені багаторівневі системи управління складською логістикою. Ці системи дають можливість повної автоматизації усіх процесів, але їх доволі висока вартість та складність у впровадженні, а також у навчанні персоналу роблять їх недоступними для більшості. Так для малого та середнього бізнесу додаткові витрати часто є непосильними, оскільки суттєво зменшують рентабельність бізнесу або ведуть до підняття ціни на продукцію.

Для багатьох виробничих або торгівельних компаній бажаним варіантом було б мати можливість не витрачати значні кошти на закупівлю та обслуговування дорогих систем управління, а знайти ефективний і доступний інструмент, який дозволить оптимізувати роботу складу без надмірних зусиль та великих коштів.

Таким чином, перед підприємствами наразі постає задача віднайти баланс між вартістю та ефективністю впровадження автоматизованої системи.

Один із сучасних інструментів, що відповідає таким вимогам і є на даний час одним з найперспективніших, - це Oracle APEX (Application Express) – потужна low-code платформа, яка дозволяє швидко створювати вебдодатки для роботи з базами даних. Вона має зручний інтерфейс, має широкі можливості для оптимізації складських процесів, а також підтримує високий рівень продуктивності та безпеки. Використовуючи Oracle APEX для розробки вебдодатків для управління складом, можна уникнути проблем, притаманних

традиційним методам обліку, і водночас зберегти простоту у впровадженні та використанні.

Актуальність теми полягає в тому, що автоматизація складських процесів є важливим фактором для підвищення продуктивності бізнесу, зменшення витрат та мінімізації ризику помилок. Використання сучасних вебдодатків для управління складом дозволяє значно покращити контроль за рухом товарів, спростити взаємодію між співробітниками та забезпечити точність у веденні обліку.

Об'єктом дослідження роботи є процес оптимізації обліку та управління товарними запасами на складі.

Предмет дослідження – метод реалізації вебдодатку управління складом за допомогою low-code платформи Oracle APEX.

Метою дослідження є розробка вебдодатку на базі платформи Oracle APEX, спрямованого на автоматизацію управління складськими операціями. Передбачається досягнення оптимізації шляхом автоматизації процесів обліку товарів, мінімізації ручного введення даних, забезпечення достовірності та актуальності інформації про залишки, а також створення зручного і доступного інтерфейсу для персоналу. Очікуваний ефект оптимізації полягає у скороченні часу на виконання щоденних операцій, зниженні ймовірності помилок, підвищенні прозорості внутрішніх процесів та покращенні якості управлінських рішень.

Для досягнення поставленої мети потрібно виконати такі завдання:

1. Розглянути теоретичні основи вебдодатків, баз даних та управління процесами складу.
2. Проаналізувати існуючі системи управління складом.
3. Визначити вимоги до функціональності вебдодатку.
4. Розробити структуру бази даних вебдодатку, яка забезпечує зберігання та обробку інформації про товари, накладні, взаєморозрахунки та інвентаризаційні дані.

5. Реалізувати функціональність основних складських операцій, механізми відстеження взаєморозрахунків і створити підґрунтя для прийняття управлінських рішень.
6. Перевірити розроблений додаток на предмет коректності роботи та зручності використання.



РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ПРОЦЕСУ УПРАВЛІННЯ СКЛАДОМ

1.1. Основні задачі управління складом

Керування складом - це комплекс заходів організаційно-логістичного та облікового спрямування, націлених на забезпечення ефективності функціонування складу на підприємстві. Ці заходи охоплюють весь процес обліку перебування запасів на складах - від моменту надходження матеріалів до кінцевого їх списання з обліку. Керування складом включає не лише облік матеріальних цінностей, а й постійний контроль їх кількості та проведення інвентаризації.

Цілісність і точність інформаційних даних є ключовими для ефективного управління складом, тому що саме на них базуються прийняття рішень у сферах закупівель, логістики та фінансового планування. Ведення обліку дозволяє не лише реалізовувати операційну документацію, а й формувати аналітичний фундамент для виявлення проблем і знаходження шляхів їх вирішення. Це має особливе значення сьогодні, коли склад - це не лише місце фізичного збереження продукції, а важлива ланка в структурі управління ресурсами підприємства, що тісно взаємодіє з іншими структурами. Сучасне управління складом включає такі ключові задачі:

1. Облік надходження товарів

Потрібно забезпечити точну реєстрацію всіх товарів, які надходять на склад. Це передбачає збереження такої інформації як дата отримання, джерела постачання, дані постачальника, кількість, вартість, а також інших супровідних даних. Облік надходжень допомагає відстежувати джерела поповнення запасів та вести історію взаємодії з постачальниками і покупцями.

2. Фіксація вибуття та списання товарів

Важливою є своєчасна фіксація всіх випадків вибуття товару зі складу: реалізація, передача до інших підрозділів, повернення, утилізація чи

списання з тих чи інших причин. Це дозволяє відслідковувати актуальні залишки та здійснювати контроль за обігом товару, при цьому уникати випадки, коли приймається до уваги та продукція, якої вже немає на складі, але про яку забули оновити дані.

3. Проведення інвентаризацій

Періодична перевірка наявності товарів і співпадиння цієї кількості із поточною інформацією, є ключовим завданням для запобігання розбіжностям між обліковими та реальними даними. Інвентаризація допомагає виявити нестачі або надлишки, та можливі зловживання.

4. Ведення взаєморозрахунків з постачальниками

Управління фінансовими взаємовідносинами передбачає облік проплат, заборгованостей, авансів та уточнення умов постачання. Це важлива складова обліку, яка дозволяє оцінити умови співпраці з кожним контрагентом та планувати майбутні поставки.

5. Аналіз складських процесів та підтримка прийняття рішень

Збір і узагальнення даних щодо руху товарів, залишків, обсягів поставок і динаміки обігу дозволяє здійснювати аналітичну оцінку роботи складу. На основі цієї інформації приймаються рішення щодо оптимізації запасів, корекції графіків постачання, тощо.

Отже, діяльність складу має бути прозорою, гнучкою та інтегрованою з іншими підрозділами підприємства, включаючи фінансовий, логістичний, комерційний та аналітичний, тому що склад завжди є основою в ширшому ланцюжку бізнес-процесів. І саме тому, будь-яка система, яка автоматизує складські процеси, повинна враховувати потреби інших підрозділів — не тільки технічно, а й логічно.

У першу чергу, це стосується бухгалтерії, якій необхідно мати доступ до первинних документів: накладних, актів списання, звітів про рух товарів. Бухгалтери використовують ці дані для формування фінансової звітності, аналізу витрат, податкового обліку. Якщо система управління складом не дозволяє оперативно передавати дані в бухгалтерські програми або формувати звичні для

них документи, виникають дублювання й помилки — а отже, втрачається весь ефект автоматизації.

Ще один критично важливий підрозділ — відділ продажів або комерційний відділ. Фахівці з продажу повинні мати доступ до актуальної інформації про залишки товарів, терміни придатності, партії, зарезервовані позиції, тощо — без цього вони не можуть формувати реальні комерційні пропозиції або виставляти рахунки. Якщо така інформація недоступна в режимі реального часу, або потребує додаткових запитів до складу, це сповільнює продаж, погіршує клієнтський сервіс і створює додаткове навантаження на персонал.

Також важливо враховувати логістичний або закупівельний відділ. Логісти мають потребу в даних про надходження, терміни доставки, залишки на складах. Працівники, що займаються закупками — у відстеженні обсягів попередніх поставок, історії постачальників, частоти замовлень.

Отже, як можна бачити з описаного вище, різні співробітники потребують конкретної інформації, пов'язаної з їх обов'язками. Наприклад, комірник потребує лише інформації безпосередньо про накладні та стан складу. Саме тому важливо в системах передбачати розмежовування доступу, навіть щоб зайва інформація не заважала роботі. У системах, які не дозволяють підтримувати гнучкий розподіл прав доступу всі співробітники мають доступ до всього, що є шкідливою практикою навіть просто для безпеки підприємства.

В загальному, існує чотири основні методи автоматизації складу: базова автоматизація, автоматизація системи, механізована та розширена автоматизація, кожен з них має свої особливості [1].

Важливо розуміти, що побудова логіки складських процесів і підрозділів, а також вибор методу автоматизації складу, безпосередньо залежать від специфіки підприємства та характеру його діяльності. Кожна компанія, незалежно від розміру чи бюджету, має потребу в обліку надходжень, вибуття, зберігання та контролю товарних запасів. Навіть якщо організація не має окремих спеціалізованих підрозділів, усі функції все ж таки виконуються відповідними працівниками. Надійне інформаційне середовище, яке забезпечує

достовірні дані про стан запасів, постачальників, рух товарів та фінансові взаєморозрахунки, стає важливою основою стабільної роботи всіх бізнес-процесів.

База даних — це структуроване сховище інформації, яке дозволяє зберігати, впорядковувати, змінювати та швидко отримувати потрібні дані. У контексті складського господарства база даних дозволяє фіксувати всі ключові операції, забезпечуючи контроль та актуальність кожного з процесів. З технічної точки зору, база даних — це набір взаємопов'язаної інформації, яку зберігають в електронному вигляді та керують за допомогою спеціальних програм — системами управління базами даних (СУБД).

Сучасні інформаційні системи складно уявити без баз даних — вони буквально в основі всього, що працює автоматично: документообіг, фінанси і навіть простий запис на прийом до лікаря. Фактично, кожна дія користувача в інтернеті або в інформаційній системі так чи інакше стосується запису, зчитування чи оновлення даних у певній базі. Навіть проста дія, як-от лайк у соцмережі чи оформлення замовлення в онлайн-магазині, передбачає роботу з базою даних.

Бази даних забезпечують надійність та достовірність інформації, дозволяючи здійснювати обробку даних ефективно та безпечно. У межах підприємства це може бути централізована база товарів, у якій містяться детальні відомості про залишки, надходження, ціни, постачальників, штрих-коди, партійні номери, терміни зберігання та історію змін. Така структура дає змогу не лише оперативно здійснювати пошук і облік, а й формувати підґрунтя для подальшої аналітики.

Наявність єдиного джерела усуває потребу дублювати дані в різних таблицях або системах, а отже, дозволяє мінімізувати ризики розбіжностей і неточностей. Такий підхід пришвидшує перевірку інформації, підвищує довіру до аналітичних звітів і значно економить час на комунікацію між підрозділами. Це особливо критично для компаній, які працюють із товарами які швидко

псуються, ведуть юридично значимий облік або обслуговують велику кількість замовлень і клієнтів одночасно.

Крім того, сучасні бази даних підтримують розмежування прав доступу та журнали змін, що дозволяє не тільки зберігати актуальні дані, але й відстежувати, хто і коли вніс ту чи іншу зміну. Це особливо важливо у корпоративному середовищі, де одночасно з базою працює велика кількість користувачів з різними ролями — від оператора до адміністратора системи.

Усі описані задачі управління складом можуть бути реалізовані лише за умови наявності точної, структурованої та завжди доступної інформації, яку забезпечує належним чином організована база даних - вона забезпечує не лише збереження даних, а й створює умови для їх ефективного використання — із гарантією безпеки, прозорості, швидкого доступу та узгодженості між підрозділами. У контексті розробки вебдодатків, у тому числі систем для управління складом, саме правильно структурована база даних дозволяє забезпечити цілісний облік усіх операцій. Без надійної бази неможливо досягти аналітичної точності, необхідної для прийняття обґрунтованих управлінських рішень і прогнозування подальших дій у логістиці, фінансах чи закупівлях.

Окрім побудови якісної бази даних, важливим завданням є також налаштування механізмів моніторингу і підтримки цілісності інформації в режимі реального часу. Це включає автоматичний контроль допустимих меж залишків, відстеження термінів зберігання продукції, а також виявлення аномалій у русі товарів, тощо. Наявність таких функцій дозволяє своєчасно реагувати на помилки, запобігати втратам і підтримувати високий рівень облікової дисципліни на складі.

1.2. Організація баз даних у системах обліку складу

У контексті систем управління, зокрема складських, реляційна база даних є ядром усього бізнес-процесу. Реляційна база даних забезпечує стабільну й надійну роботу системи, оскільки її функціонування відповідає вимогам ACID (Atomicity, Consistency, Isolation, Durability — атомарність, несуперечливість,

ізолюваність, довговічність) [5], що допомагає уникнути помилок і втрат даних навіть під час збоїв або одночасної роботи декількох користувачів. Тому у межах даної роботи буде розроблятися та застосовуватися саме реляційна база даних.

Основними складовими реляційної БД є:

- таблиці — основні структури, що слугують для зберігання інформації;
- записи — рядки у таблиці, які представляють окремі об'єкти та інформацію про них (наприклад, один товар);
- поля — стовпці, які містять властивості записів (наприклад, назва, ціна, кількість);
- ключі — для ідентифікації та зв'язку між таблицями (первинний та зовнішній ключ).

Щоб отримати доступ до даних, у реляційній базі використовують мову SQL (Structured Query Language). Вона дозволяє формувати запити на вибірку, зміну, додавання чи видалення даних тощо.

Окрім базових запитів, SQL надає можливість виконувати складніші операції: об'єднання декількох таблиць, групування різних даних, використання підзапитів, створення представлень та збережених процедур, і ще велику кількість можливостей. Це дозволяє збудувати потужну логіку взаємодії з даними безпосередньо на рівні СУБД, що надзвичайно корисно в тих системах, де є велика кількість операцій та користувачів.

Під час роботи з базами даних необхідно, щоб було враховано питання цілісності даних. Для цього використовуються транзакції – послідовність операцій над БД, розглянутих СУБД як єдине ціле [8]. Використання такої послідовності дає гарантію збереження консистентного стану бази навіть при наявності помилок чи збоїв. Крім того, сучасні СУБД підтримують обмеження на рівні полів, які дозволяють автоматично перевіряти, чи не перевищено значення допустимого діапазону, чи заповнені обов'язкові поля, чи значення ключів є унікальним. Забезпечення логічної цілісності даних надає можливість зміни логічного представлення бази даних без необхідності зміни фізичних структур зберігання даних [17].

У світі в даний час розроблені і застосовуються сотні різних систем управління базами даних, які можна класифікувати по виду використовуваної програми, характеру і моделі даних [4]. Найбільш поширеними сьогодні є Oracle Database, MySQL та PostgreSQL.

MySQL — це відкрита СУБД, яка часто використовується у невеликих або середніх веб-проектах. Вона проста у встановленні, має базовий набір функцій для роботи з реляційними таблицями та активно підтримується спільнотою. Проте, її можливості масштабування, безпеки та керування транзакціями обмежені у порівнянні з корпоративними рішеннями.

PostgreSQL — ще одна потужна система з відкритим кодом, яка орієнтована на розширюваність і точність. Вона підтримує складні типи даних, має більш розвинуту підтримку транзакцій, але для того, щоб досягти максимальної продуктивності, вимагає глибшого налаштування. PostgreSQL часто використовується у наукових та фінансових проектах, де важливою є точність розрахунків.

Oracle Database відрізняється від конкурентів високою продуктивністю, функціональністю та стабільністю. Ця платформа є комерційною, вона досить активно використовується у великих корпоративних системах, критично важливих бізнес-процесах, банківській сфері та державних установах. Вона підтримує розподілені транзакції, масштабованість, вбудовану реплікацію, розширені засоби безпеки, а також автоматичне резервне копіювання.

У багатьох сучасних компаніях бази даних є невід'ємною частиною внутрішніх інформаційних систем. Саме завдяки їм відбувається централізований доступ до важливої інформації, узгодження даних між різними відділами, обробки транзакцій і збереження історії змін. База даних не лише зберігає інформацію, але і забезпечує логіку її структури, тобто встановлює зв'язки між даними, що дозволяє системі функціонувати ефективно, навіть при великій кількості користувачів та даних.

Окрім того, з розвитком інформаційних технологій, зростаючою є потреба у масштабованості баз даних. Це означає, що база має гарантувати швидку

обробку запитів, попри той факт, що обсяг інформації постійно зростає. Для цього використовують різні методи кешування, індексації та оптимізації запитів. Такі методи зменшують навантаження на сервери та покращують загальну продуктивність системи, оптимізують її роботу.

Хоча використання баз даних відіграє ключову роль у роботі складських облікових систем, на практиці досить часто трапляються помилки на етапі створення її структури. Ці помилки не лише ускладнюють подальший розвиток системи, а й призводять до неточностей у даних, збоїв у роботі та загального падіння довіри до автоматизованого обліку. Однією з найпоширеніших проблем є відсутність нормалізації. Якщо база спроектована без урахування принципів нормалізації, дані дублюються в декількох таблицях або навіть у межах однієї. Це призводить до невідповідностей: змінивши інформацію в одному місці, її часто забувають оновити в іншому. У результаті — різні звіти на основі однієї й тієї ж інформації дають різні значення. Наприклад, якщо назва постачальника змінена в таблиці поставок, але не оновлена в таблиці звітів — виникає розбіжність, яка є неочевидною для користувача, проте є критичною для обліку.

Ще однією критичною помилкою можна вважати відсутність первинних ключів або, наприклад, використання таких полів для ідентифікації записів, що не є унікальними і є нерелевантними. Без надійної системи ключів неможливою є реалізація зв'язків між таблицями. Це в свою чергу призводить до дублювання, втрати зв'язків, проблем під час спроб фільтрувати, з'єднувати або оновлювати записи. Наприклад, у таблиці товарів відсутній первинний ключ, і тоді це дозволяє вносити декілька однакових записів одного й того ж товару. Під час інвентаризації, та й не лише в цьому випадку, такі дублікати спричиняють плутанину: неможливо визначити, який саме товар повинен бути списаним або проданим, і чи залишок товару є повністю облікованим.

Часто поширеною є така ситуація, коли модель бази даних занадто ускладнена. Великі схеми можуть нараховувати десятки таблиць, ці таблиці в свою чергу можуть містити численні обмеження та зв'язки і технічно вони можуть бути правильними, проте зовсім нечитабельними та фактично

непридатними для використання персоналом, який не має досвіду роботи з базами. У наведеній ситуації навіть прості дії — такі як додавання нового надходження чи перевірка залишку товару — може потребувати проходження через декілька вкладених таблиць та запитів, і це вимагає відповідної високої кваліфікації працівників і при відсутності такої часто призводить до помилок через людський фактор.

З іншого боку, надмірне спрощення моделі також може створювати проблеми. Якщо розробник вирішує зекономити ресурси, час чи зусилля, і вносить всю інформацію до декількох універсальних таблиць без структури — така система втрачає свою гнучкість. У майбутньому буде складно розширити її функціональність, створювати і реалізовувати нові типи звітів або додавати нові бізнес-процеси. До прикладу, у таблиці з операціями може не бути поділу на вхідні та видаткові накладні — усі записи в такому випадку зберігаються разом, без типізації. Це робить неможливим точний аналіз даних та ускладнює контроль за ними.

Таким чином, важливим є забезпечення прозорості та читабельності структури бази даних, без надмірних ускладнень і спрощень. Назви таблиць, полів та ключів повинні бути однозначними, логічними та такими, що відповідають правилам найменування, які є прийнятими у проєкті. Наявність чіткої документації структури бази забезпечує простоту в підтримці системи, полегшує інтеграцію між іншими сервісами та робить ризики помилок при розширенні функціональності мінімальними.

Таким чином, професійне, логічне та грамотне проєктування бази даних — це не лише технічні питання, а основа надійності в роботі усієї складської системи. Помилки на цьому етапі можуть спричинити значні труднощі вже на етапі щоденного використання: від неможливості обліку реальних залишків до неможливості створити звітність. Тому нормалізація, використання ключів, баланс між складністю та простотою моделі є обов'язковою умовою для успішного оцифрування складського процесу.

Також важливим аспектом є забезпечення надійності та безпеки бази даних. Цього можна досягнути за допомогою реалізації механізмів резервного копіювання, можливості відновити дані після збою, а також розмежувати права доступу для користувачів. У сучасних СУБД наявною є можливість створення окремих ролей, до прикладу, «читач», «редактор», «адміністратор», тощо і ці ролі визначають, що і для кого може бути видимим або доступним для редагування в системі. Подібний підхід зменшує ризики випадкових чи помилкових змін, а також запобігає витоку інформації, яка може бути конфіденційною і доступ до неї мають лише визначені користувачі.

З урахуванням різних еволюційних особливостей з'явилася і існує велика кількість дуже різних засобів забезпечення безпеки, що в підсумку призвело до відсутності комплексного розуміння безпеки даних [2]. Хоча передбачення всіх можливих загроз для безпеки баз даних є неможливим, розробник всерівно повинен намагатися зробити все можливе для розробки та впровадження надійних засобів захисту, таких, що відповідають сучасним стандартам, можуть витримати сучасні вимоги та навантаження, а також сприяють зменшенню ризиків втрати даних чи несанкціонованого доступу до них.

Ще одним з важливих аспектів є організація перевірок та контролю даних безпосередньо на рівні бази. Використання механізмів перевірки заданих обмежень дозволяє автоматизувати запобігання внесення некоректних, невідповідних даних. Наведу такий приклад, що обмеження унікальності (UNIQUE) дає можливість забезпечити відсутність повторення записів у критичних полях, таких як коди товарів чи партійні номери. Використання обов'язкових полів (NOT NULL) дає гарантію, що в базі не будуть залишеними незаповнені важливі атрибути.

Окрім статичних обмежень, є ще таке поняття як тригери, які широко застосовуються у реляційних базах даних. Тригери — це спеціальні автоматизовані дії, що запускаються у відповідь на зміну даних у таблицях. Наприклад, під час додавання нового запису про отримання товару тригер має можливість автоматичного оновлення залишків на складі або створення запису в

журналі змін, також є можливість передати інформацію в іншу таблицю, переробити певну необхідну інформацію – отже загалом може виконувати багато різноманітних функцій. При цьому суттєво зменшується ризик помилок, які людина може зробити при виконанні рутинних операцій, та і просто спрощує та пришвидшує виконання роботи, звільняє від необхідності робити щось додатково.

Разом з тим інформаційних технологій еволюціонували, і сприяли появі нових, інших підходів до роботи з базою даних. Зокрема, з огляду на зростання попиту на хмарні технології, усе більшої популярності набуває формат DBaaS (Database as a Service). Цей формат надає можливість розміщувати бази даних у хмарних середовищах та отримувати до них доступ через інтернет, без необхідності самостійного налаштування серверів, оновлення чи захисту. Такі сервіси, як Oracle Cloud Infrastructure, Amazon RDS, дають змогу швидко розгорнути масштабовану і гнучку систему, що особливо корисно та зручно для проєктів із динамічно змінюваними потребами.

1.3. Принципи роботи вебдодатків управління складом

Оскільки для більшості задач складського обліку потребою є зручний доступ до даних, взаємодія персоналу з системою в реальному часі та можливість керувати інформацією з різних пристроїв, все більшого значення набувають вебдодатки. Вебдодаток — це така прикладна програма, що запускається через браузер і доступ до неї є з будь-якого пристрою, який має інтернет підключення. Цю прикладну програму не потрібно встановлювати на комп'ютері або смартфоні, і це робить її особливо зручною для користувачів. Усі необхідні оновлення, зміни у функціоналі чи виправлення помилок відбуваються на сервері, тому користувач завжди має можливість працювати з актуальною версією програми.

При розширенні діяльності підприємства зростає обсяг товарообігу та збільшується кількість персоналу і тоді виникає потреба в масштабуванні облікової системи — як за кількістю користувачів, так і за кількістю об'єктів

обліку. Без використання вебдодатків та централізованої системи це стає дедалі складнішим: з'являється ризик того, що дані можуть бути продубльовані. Також втрачається узгодженість між відділами, а під час доступу до актуальної інформації виникають певні труднощі. Кожен новий пристрій, на якому виконується робота, потребує окремого налаштування передачі даних або дублювання функціональності, і це вимагає додаткового часу, ресурсів та зусиль. Натомість вебдодаток дозволяє легко масштабувати систему без суттєвих витрат ресурсів та часу— достатньо буде лише надати доступ до системи для нового користувача, аби він чи склад міг працювати з єдиною інформаційною платформою в реальному часі.

Ще однією важливою перевагою є незалежність від операційної системи, встановленої на робочому пристрої користувача. Оскільки програма працює у браузері, вона є однаково доступною з Windows, macOS, Linux, Android чи iOS. І це дозволяє значно зменшувати витрати на впровадження і підтримку процесів обліку, тому що не потрібно створювати окремі версії програми для кожної платформи. Окрім того вебдодатки спрощують централізоване керування доступом: адміністратор має права призначати різні ролі, обмежувати або розширювати можливості для конкретних користувачів, вести контроль журналу подій. Завдяки цьому безпека підвищується і ймовірність недозволених змін у системі зменшується.

Вебдодатки, які призначаються для управління складом, виконують функції моніторингу, обліку а також аналізу руху товарів на складах. Їх головна мета — забезпечити швидкий доступ до актуальної інформації про стан товарів на складах, спростити щоденну роботу персоналу та звести до мінімуму ризик помилок, які можуть виникнути в результаті ручного введення даних.

Уявімо підприємство, де щодня проходить до 100 операцій надходження та списання товарів. За відсутності вебдодатку працівники витрачають багато часу для пошуку фактичних залишків. Необхідність заповнювати форми вручну, пошук можливості пересилання інформації між відділами також не сприяє процесам оптимізації. Як наслідок з'являються затримки при відвантаженні

товарів, обслуговуванні покупців, помилки в документах, а керівництво не має оперативного доступу до звітності. Вебдодаток спрощує та покращує ці процеси, дозволяючи скоротити час обробки даних і підвищує якість обліку.

Основний принцип роботи таких додатків полягає в тому, що всі складські операції реєструються у системі в реальному часі. Кожна дія, виконана користувачем, миттєво фіксується в базі даних, що забезпечує постійну актуальність інформації. Усі дані оновлюються динамічно, тож керівництво, бухгалтерія, менеджери завжди мають доступ до актуального стану складу, актуальних залишків товарів, обсягів руху, історії поставок та іншої аналітичної інформації, яка допомагає приймати обґрунтовані управлінські рішення.

Користувач під час роботи з інтерфейсом вебдодатку взаємодіє з базою даних через набір форм, кнопок, таблиць і звітів. Кожна дія призводить до виконання певного запиту до бази: збереження, оновлення або отримання інформації.

Системи управління складом зазвичай реалізують такі ключові функції:

- облік взаєморозрахунків із постачальниками або клієнтами;
- облік залишків товарів;
- ведення історії надходжень і витрат;
- контроль інвентаризацій;
- реєстрація поставок і продажів;
- облік взаєморозрахунків із постачальниками або клієнтами;
- ведення довідників товарів, контрагентів, складів;
- контроль за залишками у різних одиницях виміру.

Облік залишків товарів дозволяє в реальному часі бачити, скільки одиниць кожної позиції знаходиться на складі, і це мінімізує ризик помилок при видачі або списанні. Ведення історії надходжень і витрат забезпечує можливість відстежувати усі операції і дозволяє аналізувати рух товарів у розрізі дат, контрагентів, оплат.

Реєструвати поставки та продажі є важливим завданням для взаємодії зі сторонніми організаціями, наприклад постачальниками та клієнтами. Кожна

така операція фіксується з прив'язкою до конкретної дати, особи та документа. Контроль інвентаризацій — це інструмент, який дозволяє зробити звірку фактичних залишків товару з даними обліку та виявити відповідно наявність нестач або надлишків.

Окрему увагу потрібно приділити довідникам. Це централізовані списки товарів, складів, контрагентів, одиниць виміру, які використовуються для заповнення документів. Наявність єдиної структури довідників забезпечує цілісність даних і запобігає дублюванню записів.

З технічної точки зору, веб-додаток управління складом реалізується за допомогою:

1. Інтерфейсу користувача — доступного через браузер, де працівники можуть вводити або переглядати дані.
2. Серверної частини — вона обробляє логіку програми: перевіряє вірність введених даних, обчислює підсумки, формує звіти.
3. Бази даних — у ній зберігається фактично вся складська інформація, подана у вигляді таблиць: "товари", "накладні", "контрагенти", "операції" тощо.

Коли користувач працює з формами, кнопками чи запитамі на сайті, інформація передається на сервер, обробляється та записується в базу даних або витягується з неї.

Сценарій взаємодії виглядає так:

1. Користувач вводить дані (наприклад, додає товар).
2. Клієнт надсилає запит до серверу.
3. Сервер обробляє інформацію.
4. База даних зберігає або повертає потрібні дані.
5. Користувач бачить результат на екрані.

Залежно від обраної технології та типу веб-додатку, ці складові можуть реалізовуватись у різних середовищах — окремо або в межах інтегрованих платформ. У випадку використання таких платформ, як правило, розробник має

доступ до готових інструментів для створення форм, таблиць, звітів і керування логікою, що значно скорочує час розробки.

Існують різні типи вебдодатків:

- статичні — прості сторінки, що залишаються сталими, тобто не змінюються незалежно від дій користувача;
- динамічні — контент, тобто їх наповнення, генерується у відповідь на дії користувача або зміни у даних;
- SPA (Single Page Application) — усі дії відбуваються в межах однієї сторінки без перезавантаження;
- PWA (Progressive Web App) — мають функціональність, подібну до мобільних додатків, можуть працювати офлайн.

Кожен із наведених типів має свої переваги та обмеження. Наприклад, SPA-додатки забезпечують швидку взаємодію без перезавантаження сторінки, але можуть бути досить складними у реалізації без досвіду JavaScript. Натомість класичні динамічні додатки простіші в розгортанні й чудово підходять для облікових систем, де стабільність і надійність обробки даних є набагато важливішими за швидкість оновлення інтерфейсу.

У проектах, які пов'язані з обліком, наявність правильно спроектованої логіки додатку є критично необхідною, оскільки саме від структури даних залежить наскільки всі система буде зручною, надійною та швидкою. Якісно продуманий додаток у поєднанні з логічно правильно розробленою базою даних дають змогу уникнути дублювання інформації, зберігати зв'язки між об'єктами та забезпечувати коректну роботу звітності й пошуку.

Разом із тим, створення таких рішень традиційними методами часто потребує значних ресурсів: часу, зусиль розробників та глибоких технічних знань. Це стає особливо помітним в умовах, при яких виникає необхідність швидкої реалізації прикладної системи з конкретними функціями обліку, формування звітів та зручною взаємодією з користувачами.

У відповідь на ці виклики, у сучасних умовах все ширшого поширення набувають low-code платформи — інструменти, що дозволяють створювати веб-

додатки з мінімальним використанням традиційного програмування або взагалі без нього [7]. Вони є орієнтованими на максимально швидке створення і впровадження прикладних рішень, які можна легко адаптувати до необхідних бізнес-процесів або до специфіки у внутрішньому середовищі підприємства. Завдяки наявним візуальним конструкторам, готовим модулям і автоматизованим механізмам підключення до баз даних, low-code платформи значно знижують бар'єр входу і пришвидшують розробку.

Інтегровані платформи дозволяють значно пришвидшити процес створення додатків завдяки використанню вбудованих інструментів — шаблонів форм, генераторів звітів, автоматичного зв'язування з таблицями бази даних. Такий підхід знижує вимоги до знань мов розробки та пришвидшує впровадження.

На окрему увагу заслуговує той факт, що low-code рішення є оптимальним рішенням у випадках, коли підприємства зіштовхуються з такими перешкодами, як обмежений бюджет, відсутність кваліфікованих програмістів, низька цифрова грамотність персоналу та з низкою інших факторів. Завдяки простому графічному інтерфейсу та наявності візуальних шаблонів, розробку додатку можуть здійснювати не лише професійні IT-фахівці, а й фахівці з предметної області — логісти, аналітики чи менеджери складу, і це, в свою чергу, значно пришвидшує впровадження цифрових рішень без необхідності масштабного навчання персоналу. Тому low-code платформи є ефективними, особливо для малого та середнього бізнесу, де важливо швидко отримати результат з мінімальними витратами ресурсів.

Окрім основних облікових функцій, сучасні складські веб-додатки можуть надавати можливості для автоматичного формування супровідної документації, наприклад, товарних накладних або актів списання. Це дозволяє значно скоротити час, який був би витрачений на паперову роботу та зменшити кількість помилок при оформленні документації. У середовищі, де є багато користувачів, це дозволяє ефективніше контролювати роботу персоналу та уникати зловживань.

Важливим фактором у розробці систем управління складом є простота використання інтерфейсу. Вебдодаток має бути інтуїтивно зрозумілим навіть для працівників, які не мають спеціальної ІТ-освіти або досвіду роботи з комплексними інформаційними системами. Надмірно складні або перевантажені інтерфейси значно підвищують ризик помилок, уповільнюють виконання операцій і можуть викликати відторгнення користувачів. Саме тому під час проєктування таких рішень велика увага приділяється логічності розміщення елементів, зрозумілій навігації та скороченню кількості кроків, які можуть бути необхідними для виконання типових операцій та дій.

Зокрема, в основному повинні бути передбаченні наступні фактори:

- чітка структура меню та й в цілому додатку, де основні операції (реєстрація надходжень, видача товарів, перегляд залишків, проведення інвентаризації) можуть бути доступними в один-два кліки;
- автоматичні розрахунки, такі як підсумки кількості товарів або вартості партій, щоб знизити ризик арифметичних помилок користувачів.

Завдяки такому підходу підвищується ще й якість даних, що вносяться до системи, адже мінімізуються випадкові помилки, забуті поля чи дублювання записів.

Одним із важливих аспектів сучасного управління складом є також інтеграція вебдодатку з мобільними пристроями. Практика показує, що працівникам складу часто необхідно оперативно вносити дані про надходження або переміщення товару, переглядати залишки, перевіряти партії або здійснювати інвентаризацію безпосередньо на місці, не повертаючись до стаціонарного комп'ютера.

Тому система має забезпечувати адаптивний інтерфейс, який коректно може бути відображеним на екранах смартфонів і планшетів без втрати функціональності;

Впровадження мобільної підтримки дозволяє значно оптимізувати внутрішні процеси складу, скоротити час обробки товарів, знизити кількість затримок у обробці даних і забезпечити вищий рівень прозорості усіх операцій в

реальному часі. Таким чином, мобільна інтеграція є не просто додатковою функціональністю, а важливою перевагою коли постає питання, яку систему обрати серед наявної різноманітності складських інформаційних систем.

Крім цього, важливою є можливість резервного копіювання даних. Втрата інформації може мати серйозні наслідки для підприємства, тому сучасні веб-додатки повинні мати автоматизовані механізми створення резервних копій з можливістю швидкого відновлення. Така функція підвищує надійність системи та забезпечує безперервність бізнес-процесів навіть у разі технічних збоїв.

Отже, веб-додаток для управління складом — це не лише технічний інструмент, а й організаційне рішення, яке дозволяє зробити роботу складу більш прозорою, контрольованою та ефективною. Його впровадження позитивно впливає на швидкість обробки операцій, зменшує витрати часу та ресурсів, а також забезпечує більш якісний облік товарів, що особливо важливо для підприємств з великою номенклатурою продукції або інтенсивним товарообігом.

1.4. Постановка завдання

У межах дослідження передбачається розробка вебдодатку для управління складом із використанням платформи Oracle APEX — інструменту, що поєднує гнучкість технологій low-code із потужністю реляційних баз даних. Основна мета полягає у створенні інтегрованої системи, яка забезпечить надійний, інтуїтивний та функціональний інтерфейс для ведення складських процесів, сприятиме процесу цифровізації обліку товарів на підприємствах малого та середнього бізнесу а також дозволить швидко адаптуватися до змін у бізнес-процесах.

Вебдодаток має на меті:

- підвищити точність обліку товарів, мінімізуючи ризики помилок, що виникають під час ручного введення даних;
- забезпечувати доступ до актуальної інформації щодо залишків, руху товарів і фінансових розрахунків;
- знизити навантаження персоналу завдяки автоматизації рутинних операцій та стандартних бізнес-процедур;

- покращити якість управлінських рішень, надаючи доступ до структурованих аналітичних звітів та інструментів моніторингу.

Основні процеси, які має підтримувати система, включають:

- реєстрацію надходження товарів від різних постачальників із фіксацією всіх супровідних даних (дата, партія, вартість, кількість, характеристики);
- облік відвантаження товарів за видатковими накладними з урахуванням кількісного балансу;
- проведення планових і вибіркового інвентаризацій із можливістю порівняння фактичних і облікових залишків;
- ведення взаєморозрахунків із контрагентами: реєстрація оплат, заборгованостей;
- формування оперативних та управлінських звітів про стан запасів, товарообіг, ефективність роботи складу.

У рамках роботи буде розроблено базу даних та вебдодаток для управління складом одягу, а також проведено його тестування на прикладах. Предметна область охоплює повний життєвий цикл товарних запасів: від надходження продукції на склад до її вибуття в результаті продажу, передачі або списання.

Ключові вимоги до системи включають:

- забезпечення актуальності даних у будь-який момент;
- підтримку розмежування доступу відповідно до ролей користувачів (оператори складу, менеджери, бухгалтери, адміністратори);
- централізоване зберігання інформації із гарантією цілісності та безпеки даних;
- підтримку мобільної адаптивності, що дозволить користувачам працювати із системою через смартфони та планшети;
- зручність інтеграції із зовнішніми системами або розширення функціоналу в майбутньому.

Вибір платформи Oracle APEX обумовлений її перевагами:

- можливістю швидкої розробки без залучення великих команд розробників;
- наявністю готових шаблонів для побудови форм, звітів, графіків та діаграм;

- низькими вимогами до серверної інфраструктури та можливістю використання хмарного середовища Oracle Cloud Infrastructure;
- підтримкою стандартних механізмів безпеки баз даних Oracle, включаючи контроль доступу, резервне копіювання та захист від збоїв.

Таким чином, розробка запропонованого вебдодатку спрямована на досягнення таких основних цілей:

- скорочення часу на виконання складських операцій за допомогою автоматизації введення та обробки даних;
- зменшення кількості помилок при введенні інформації за рахунок застосування перевірок і обмежень на рівні бази даних;
- підвищення доступності та прозорості облікової інформації для різних підрозділів підприємства;
- підвищення контрольованості процесів управління складом та полегшення аналізу діяльності;

Очікуваним результатом стане повноцінний вебдодаток, який буде придатним для реального впровадження у сфері обліку товарів на складі одягу, із можливістю масштабування та подальшого розширення функціоналу відповідно до змін бізнес-процесів підприємства.

РОЗДІЛ 2

АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ТА ІНСТРУМЕНТІВ

2.1. Аналіз технологій для розробки вебдодатків управління складом

В наш час підприємства прагнуть, щоб управління складами було автоматизованим і зіштовхуються з задачею вибору поміж багатьох технологічних рішень. В залежності від мети, величини бізнесу, доступного бюджету та ІТ-компетенцій персоналу, підприємства мають змогу обирати серед створення власного ПЗ, використання low-code платформ або впровадження готових систем. Кожен із цих варіантів має як свої переваги так і обмеження, які необхідно детально проаналізувати.

Перший варіант — повна розробка вебдодатку від самого початку, тобто з нуля. Він передбачає написання коду для клієнтської та серверної частин, проєктування баз даних, реалізацію логіки обробки даних та створення інтерфейсу для зручного використання для користувачів. Такий підхід забезпечує найвищу гнучкість: розробники мають можливість реалізувати специфічні функції, що є повністю відповідними до бізнес-логіки конкретної організації. Особливо це стає актуальним, якщо склад має нестандартні процеси або складну структуру взаємодії з іншими відділами чи підрозділами. Водночас, власна розробка вимагає значних ресурсів — від висококваліфікованої команди до часу на тестування, супровід і подальшу технічну підтримку. Навіть за умови залучення аутсорсингових фахівців вартість такого проєкту може перевищити допустимий бюджет підприємства. Окрім того, реалізація звичайних облікових функцій займає багато часу через необхідність писати логіку з нуля, що не завжди виправдано в умовах обмежених термінів.

Другий варіант — це використовувати low-code або no-code конструктори, про які вже було згадано раніше. Останнім часом цей підхід стає все більш популярним. Завдяки наявності візуального інтерфейсу розробки, шаблонів сторінок, форм для збору даних і автоматизованих інструментів налаштування звітів, створення вебдодатків стає доступним навіть для користувачів без

програмістського досвіду. Найбільшого поширення low-code підходи набули у сфері автоматизації бізнес-процесів, зокрема складського обліку, оскільки дозволяють швидко протестувати і впровадити систему без необхідності проходження тривалого циклу розробки. До переваг відносять також низький поріг входу, підтримку мобільних інтерфейсів, централізоване керування доступом, а також можливість гнучкого масштабування функціоналу. Однак навіть такі платформи мають певні обмеження: часто вони не дозволяють реалізувати складні інтеграції або унікальну бізнес-логіку без залучення додаткового програмування. Крім того, залежність від провайдера платформи (ліцензії, тарифи, хмарна інфраструктура) створює ризики у довготривалій перспективі.

Третій варіант — це впровадження готових систем обліку, які вже містять необхідні модулі для роботи зі складом. Серед них можна також згадати модулі складу у складі ERP-систем, CRM-рішень або інших спеціалізованих логістичних платформ. Такі рішення часто інтегровані з бухгалтерськими модулями, дають змогу вести фінансовий облік, контролювати залишки, генерувати стандартні звіти. Вони можуть підійти середньому або великому бізнесу, де вже налагоджена цифрова інфраструктура. Водночас для малого бізнесу такі продукти часто є надмірно складними, з великою кількістю зайвих функцій. Крім того, ці рішення в дуже небагатьох випадках дозволяють адаптувати логіку під конкретні процеси без участі ІТ-фахівців або модифікацій системного коду, який часто є закритий, а модифікація його взагалі недоступна. Вартість ліцензії, обслуговування і навчання персоналу також є значним бар'єром та викликом.

На практиці більшість компаній використовують поєднання підходів. Наприклад, часто починають із готового шаблону чи low-code рішення, а згодом переходять до кастомної розробки, коли вимоги ускладнюються і конкретизуються під потреби замовника. Інші, навпаки, обирають кастомізоване ПЗ одразу, передбачаючи його подальше масштабування. Усе залежить від рівня

технічної зрілості підприємства, ресурсів і швидкості, з якою потрібно отримати готове робоче рішення.

При цьому вибір інструменту має враховувати не лише технічні параметри, а й практичні аспекти:

- скільки часу знадобиться на впровадження системи;
- яка вартість повної реалізації циклу рішення (від створення до обслуговування);
- які витрати на відповідне навчання персоналу та технічну підтримку;
- наскільки просто вносити зміни та масштабувати систему.

На жаль, багато готових рішень, особливо серед no-code платформ, не розраховані на багаторівневу корпоративну структуру. Вони не дозволяють:

- налаштувати гнучку модель доступу за ролями;
- розмежовувати права користувачів для перегляду, редагування, додавання чи затвердження документів;
- створювати окремі інтерфейси для різних категорій користувачів (наприклад, склад — одна панель, бухгалтерія — інша);
- робити можливою інтеграцію із зовнішніми системами (наприклад, з бухгалтерськими програмами або CRM).

В результаті підприємства змушені або дублювати частину обліку в інших системах, або вручну передавати дані між відділами, що знову ж таки повертає до паперової тяганини, можливих помилок та постійних затримок.

Ефективна система управління складом — це така, що не обмежується рамками фізичного приміщення складу. Вона має бути частиною єдиного цифрового простору підприємства, де всі учасники бізнес-процесів працюють з актуальною інформацією відповідно до своєї ролі. І якщо платформа цього не дозволяє, це суттєво знижує її практичну придатність.

Платформи, що широко застосовуються малими та середніми бізнесами, мають різний ступінь придатності до роботи з обліковими задачами. Наприклад:

Zoho Creator позиціонується як потужна low-code платформа, яка дозволяє будувати кастомні бізнес-додатки. У неї є візуальний редактор, також вона

підтримує автоматизацію бізнес-логіки, дозволяє створювати звіти й контролювати доступ користувачів.

Втім, при використанні Zoho Creator варто враховувати:

- **складність масштабування:** з часом, коли структура даних ускладнюється, система починає вимагати все більше кастомної логіки, яка пишеться на внутрішній мові Deluge. Це фактично нівелює «простоту» low-code платформи;
- **залежність від екосистеми Zoho:** для повноцінної роботи часто виникає необхідність підключати інші продукти (CRM, Books, Analytics), а це також збільшує витрати;
- **цінова політика:** базовий функціонал обмежений, а комерційні тарифи можуть нести досить високі і часто непередбачувані витрати при зростанні кількості користувачів або обсягів даних.

Airtable — інструмент, який поєднує простоту таблиць з можливостями реляційних баз. Його часто використовують стартапи та невеликі компанії для побудови MVP або швидких облікових систем.

Проте:

- **проблеми з гнучкістю:** Airtable не дозволяє створювати повноцінні бізнес-процеси з валідацією (тобто з перевіркою), умовною логікою чи інтеграцією між модулями без сторонніх сервісів;
- **відсутність нормальної обробки транзакцій:** платформа не підходить для випадків, коли важливою є цілісність даних (наприклад, синхронізована зміна залишків і накладних);
- **погана масштабованість:** безкоштовна версія досить швидко вичерпується, а варіант переходу на платну версію не вирішує проблему складних зв'язків і зростаючого обсягу записів.

AppSheet — ще один no-code інструмент, який дозволяє створювати прості мобільні чи вебдодатки на базі Google Таблиць або SQL.

Його обмеження:

- **низька гнучкість інтерфейсу:** користувацький інтерфейс практично не налаштовується. Це критично для додатків, де потрібно реалізувати зручну навігацію для роботи з великим обсягом даних;
- **обмежений контроль над структурою даних:** досить складно реалізовувати складні структури таблиць, організовувати багаторівневі зв'язки, каскадні оновлення чи унікальні обмеження;
- **залежність від Google-екосистеми:** важко масштабувати рішення поза межі внутрішнього користування компанії, особливо у випадках, коли є необхідною інтеграція з іншими ERP або CRM.

Окрім платформ для створення вебдодатків, на ринку також представлено низку готових програмних рішень для складського обліку, які позиціонуються як універсальні або спеціалізовані інструменти для малого та середнього бізнесу. Ці системи передбачають використання вже готової інфраструктури — інтерфейсів, форм, звітів, логіки обліку — без потреби розробки власного функціоналу.

Варто зазначити, що раніше в Україні доволі широко використовувалися рішення на базі 1С для обліку складських процесів, завдяки їхній доступності та великій кількості модулів, що створювали ілюзію універсальності. Насправді така структура часто ускладнювала систему і обмежувала її гнучкість. З початком повномасштабного вторгнення Росії у 2022 році питання безпеки даних і недопущення залежності від російських програмних продуктів стало критично важливим. Це спричинило активний пошук і подальший перехід підприємств на альтернативні рішення українського або міжнародного походження.

Попри очевидні переваги використання вже розроблених систем, на практиці такий підхід часто виявляється обмеженим через низку технічних або організаційних факторів.

Odoo — потужна ERP-система з відкритим кодом, яка містить модулі для складу, закупівель, продажів, обліку та іншого. Її часто називають універсальним конструктором бізнес-логіки.

Але:

- **високий поріг входу:** незважаючи на відкритість коду, налаштування Odoo вимагає від працівників значного технічного досвіду. Самостійне впровадження змін за допомогою цієї системи — це довгий і складний процес;
- **вартість впровадження та підтримки:** безкоштовна версія обмежена в функціоналі. Повноцінне використання з підтримкою всіх модулів вимагає ліцензій, технічного обслуговування;
- **перевантаженість функціями:** для малого бізнесу Odoo може виявитися «надто великим» — з зайвим функціоналом, з важким інтерфейсом та складною адаптацією до простих складських задач.

ERPNext — безкоштовна система планування ресурсів підприємства з відкритим кодом, яка містить базові модулі управління підприємством, включаючи склад, продажі, закупівлі, бухгалтерію тощо.

- **переваги:** підтримка кастомізації (можливість налаштовувати чи редагувати під власні потреби), можливість локального розгортання, наявність активної спільноти;
- **недоліки:** висока складність процесів налаштування — система вимагає наявності глибоких знань Linux, Docker і навичок використання принципів серверної архітектури. Відсутня повна українізація, що ускладнює використання у вітчизняних компаніях. Інтерфейс недостатньо інтуїтивний, містить проблеми з навігацією та не передбачає гнучких шаблонів. Окрім того, дизайн програми застарілий, а мобільна підтримка є реалізованою лише частково.

Megaventory — хмарне рішення для управління запасами, орієнтоване на компанії з розподіленою, віддаленою мережею складів.

- **переваги:** підтримка багатоскладових конфігурацій, ведення замовлень, інтеграція з Shopify та іншими платформами;
- **недоліки:** інтерфейс погано пристосований до масових операцій, що ускладнює роботу з великою номенклатурою. Кількість користувачів для базового тарифу є обмеженою. Система не дозволяє налаштовувати

індивідуальну логіку поведінки документів або форм, а витрати на підписку стрімко зростають зі збільшенням складів чи обсягу даних.

inFlow Inventory — гібридна система (локальна + хмарна), яка спеціалізується на обліку товарів, клієнтів і замовлень.

- перевагами є: простий старт, підтримка інтеграцій із CRM, можливість ведення клієнтських баз.
- недоліки: орієнтована здебільшого на англomовний ринок, без повноцінної локалізації. Низький рівень автоматизації — більшість дій виконується вручну. Відсутня чітка ролева модель для управління доступом, а базові тарифи не містять всього необхідного функціоналу.

myWarehouse — легка мобільна система, призначена для малого бізнесу та тимчасових складських об'єктів.

- переваги: швидкий запуск, простота у використанні, адаптація для смартфонів;
- недоліки: занадто обмежений функціонал — відсутні партійні номери, історія руху, облік по групах товарів. Система не підтримує створення нових типів документів або їх кастомізацію. Немає можливості підключення до зовнішніх аналітичних панелей. Також відсутній офлайн-режим, що робить систему критично залежною від доступу до Інтернету.

Також на ринку складських програм існує чимало рішень, зокрема українських розробників, які позиціонуються готовими або універсальними для автоматизації обліку. До таких належать Торгсофт, УкрСклад, Zirru CRM. Попри свою доступність і певну популярність серед малого бізнесу, ці системи мають низку обмежень, які стають критичними вже на етапі активного використання.

Наприклад, Торгсофт — це українська програма з акцентом на торгівлю, яка не дає можливості користувачам робити зміни в структурі даних або адаптувати логіку обліку під специфіку конкретного підприємства. Всі операції в даній системі прив'язані до шаблонів жорстко, тобто без можливості гнучких

змін, а розширення можливе лише через використання окремих додаткових платних модулів.

УкрСклад — ще одне українська розробка, яка хоч і забезпечує базову роботу зі складом, але суттєво відстає в частині інтерфейсу, не підтримує хмарні сервіси, а також не дозволяє ефективно працювати з правами доступу або вести аналітику без сторонніх засобів.

Ще приклад — Zipru CRM, українська система, яка деякий час пропонувала автоматизацію процесів, але наразі фактично не підтримується, а її функціональність обмежена CRM-складовою, без можливості реалізації повноцінного товарного обліку чи інвентаризації.

Доречно буде згадати ще один доволі популярний сервіс CRM Appointer. Це українська хмарна система, яка орієнтована на автоматизацію бізнес-процесів у сфері послуг, таких як салони краси, клініки, стоматології, СТО, автомийки та інші. Система, поміж інших функцій, також дозволяє організовувати інтеграцію з IP-телефонією, можливість онлайн-запису через сайт або соціальні мережі. Цей сервіс також підтримує мобільні додатки. Проте, не зважаючи на ці переваги, як вже згадувалось вище, Appointer має обмеження, в першу чергу, пов'язані з вузькою спеціалізацією - система переважно заточена під сферу послуг, а також має закриту архітектуру, що робить її негнучкою. Також є ще один великий мінус — ця система є більш орієнтованою на обслуговування фізичних осіб, а чіткої підтримки B2B сегменту вона немає.

Також не можна не згадати і про традиційні офісні інструменти, які і досі все ще активно використовуються для складського обліку в багатьох компаніях — мова про Microsoft Excel, Google Sheets, LibreOffice Calc та інші подібні табличні редактори.

Незважаючи на те, що електронні таблиці є дуже поширеними, їх використання як основи складської системи має наступні критичні обмеження:

- відсутня можливість контролювати структуру даних: користувачі можуть випадково видалити або внести зміни у важливу інформацію, що в свою чергу призводить до втрати цілісності бази;

- немає можливості для транзакційності: декілька користувачів не мають змоги одночасно працювати з таблицею без ризику виникнення конфліктів і затирання змін;
- існує висока ймовірність помилок: формули легко зламати або змінити, переписавши їх; зміст комірок часто змінюється без сліду історії змін;
- немає ролей і обмежень доступу: кожен має повний доступ до всього — це небезпечно й неприпустимо для корпоративного середовища;
- відсутність журналів змін: неможливо відслідкувати, хто і коли вніс зміни в документ;
- складність автоматизації: обробка накладних, створення звітів, облік нових надходжень, переміщень чи залишків — усе це працівникам доводиться робити вручну або за допомогою нестійких макросів.

Такі інструменти можуть бути тимчасовим рішенням для початкового обліку, але не відповідають вимогам масштабованості, безпеки, багатокористувацької роботи та автоматизації, які висувуються до сучасних складських систем. Використання їх в якості основної платформи несе в собі великий ризик для бізнесу, особливо якщо мова йде про розгалужену мережу складів, інтеграцію з бухгалтерією або необхідність формування оперативної звітності.

Під час вибору технології для розробки вебдодатків управління складом важливим є врахування не лише функціональних можливостей, а й гнучкості у впровадженні, подальшій підтримці та адаптації до змін у бізнес-процесах. Технології, які обмежуються фіксованим набором функцій без можливості гнучкої модифікації, швидко втрачають актуальність у динамічному середовищі малого та середнього бізнесу. Натомість ті платформи, які дозволяють створювати більш адаптивні інтерфейси, можуть підтримувати умовну логіку, мають налаштовуванні сценарії обробки даних а також можливості централізованого керування ролями користувачів, значно спрощують реалізацію системи при реальному використанні на практиці.

Рішення, що володіють особливою перевагою, це рішення, що забезпечують автономність — тобто не потребують складної інтеграції зі сторонніми API, модулями чи сервісами для реалізації базового функціоналу. Наприклад, вбудована підтримка генерації PDF-документів, побудови звітних графіків, експорту в таблиці Excel, надсилання повідомлень користувачам або формування динамічних звітів дозволяє зменшити залежність від сторонніх сервісів, знизити витрати та уникнути ризиків, пов'язаних зі змінами в API чи політиці сторонніх провайдерів.

Окрім того, також важливим є питання масштабованості — здатності системи рости разом із підприємством, тобто здатності підлаштовуватись під зростаючі потреби підприємства. На ранніх етапах бізнесу часто достатнім є обмежений функціонал, але зі зростанням обсягів даних, кількості користувачів та складності процесів природньо виникає потреба у розширенні логіки, впровадженні ролей доступу, контролі за діями користувачів, створенні умовно залежних полів, багаторівневих погоджень та інших можливостей. Технології, що дозволяють реалізовувати ці функції без переписування основної архітектури додатку, мають стратегічну перевагу. Це зменшує витрати на доопрацювання і дозволяє швидко вносити правки до змін в бізнес-моделі.

Не менш критично важливим аспектом при виборі технології є технічна підтримка платформи: чи є наявною офіційна документація, приклади, спільнота користувачів, а також вбудовані журнали змін, логування дій та засоби резервного копіювання. Системи, які передбачають самостійне адміністрування, значно підвищують рівень незалежності підприємства від зовнішніх розробників. Це особливо важливим є для малих компаній, які не мають власної IT-команди або не можуть постійно залучати фахівців для внесення змін у код.

Роблячи підсумок аналізу різних підходів до створення вебдодатків, можна впевнено стверджувати, що єдиного, універсального рішення не існує — кожен із підходів має свою відповідну нішу та доцільність залежно від багатьох факторів, наприклад масштабу проєкту, наявності доступних ресурсів та логіки внутрішньої організації підприємства. Власна розробка, створена під конкретний

запит, надає ширшу свободу, але вимагає значних зусиль; готові системи — швидкі у впровадженні, проте часто обмежені у налаштуваннях; low-code платформи стали компромісом між швидкістю та кастомізацією. У цьому контексті доцільним є питання оцінювання наявності технології не лише за набором функцій, а з погляду їхньої стійкості до зростання, зрозумілості для кінцевого користувача та гнучкості в адаптації до складських бізнес-процесів.

У сучасних умовах, коли підприємства прагнуть оптимізувати витрати часу, ресурсів та технічного обслуговування, особливо важливим стає вибір інструменту, який дозволяє швидко адаптувати систему до змін у бізнес-середовищі. Це стосується не лише гнучкості інтерфейсу чи зручності для користувача, а й можливостей масштабування, підтримки аналітики, простоти обслуговування та надійності на рівні бази даних. Технологічна платформа, яка відповідає цим критеріям, має стати не тимчасовим рішенням, а довгостроковою основою цифрової трансформації підприємства.

Саме тому наступний підпункт присвячено обґрунтуванню вибору конкретної технологічної платформи, яка дозволяє реалізувати вебдодаток управління складом з урахуванням усіх практичних вимог, визначених під час аналізу.

2.2. Обґрунтування використання Oracle APEX для розробки вебдодатку управління складом

Як було показано в попередньому підпункті, більшість популярних платформ, які позиціонуються як готові або low-code рішення для автоматизації складського обліку, на практиці виявляються далеко не універсальними. Попри простоту та швидкість впровадження, які їх розробники декларують, такі інструменти досить часто мають приховані технічні та організаційні обмеження, що стають очевидними вже після запуску системи в реальному виробничому середовищі. Серед найбільш поширених проблем — обмежена гнучкість структури даних, яка не дозволяє адаптувати систему до специфіки окремого

складу; залежність від сторонніх сервісів, без яких реалізація базових функцій стає неможливою; низька масштабованість, що ускладнює подальший розвиток системи; а також складнощі з технічною підтримкою у разі необхідності внесення змін або розширення функціональності.

Особливо критичними такі обмеження стають у випадках, коли мова йде про бізнес, який прагне цифровізувати процеси, але не має великої ІТ-команди чи бюджету на складні інтеграції. У багатьох випадках компанії стикаються з необхідністю обирати між функціональністю та доступністю, між гнучкістю і простотою. У результаті впроваджуються компромісні рішення, які частково закривають поточні потреби, але не дозволяють системі рости разом із підприємством.

У такому контексті постає необхідність у більш універсальному, але водночас доступному і керованому інструменті, який дозволяє поєднати переваги швидкої розробки з потужним функціоналом і високим рівнем контролю над даними. Йдеться про платформу, яка забезпечує стійкість, гнучкість, інтуїтивність для кінцевого користувача й водночас — глибоку функціональність і стабільну архітектуру для розробника або адміністратора системи. Саме таким рішенням є Oracle APEX (Application Express) — інструмент, який об'єднує переваги класичної розробки та low-code підходу, дозволяючи створювати професійні, масштабовані вебдодатки з мінімальними витратами часу й ресурсів.

Платформа Oracle APEX є універсальним інструментом для розробки складських систем різного профілю, однак саме у випадку складу одягу її переваги проявляються особливо чітко. Номенклатура одягу зазвичай характеризується великою кількістю варіацій за такими ознаками, як категорія товару, стать, сезон, розмір. Облік таких даних потребує підтримки складних реляційних зв'язків і гнучкої роботи з довідниками — завдань, які Oracle APEX вирішує ефективно завдяки розвиненій підтримці баз даних та широким можливостям побудови інтерфейсів. Крім того, система дозволяє швидко формувати інтерактивні звіти, аналітичні панелі та діаграми, що полегшує аналіз залишків, визначення популярних моделей та моніторинг сезонних тенденцій. Це

робить Oracle APEX особливо зручним рішенням для обліку й аналітики у сфері торгівлі одягом та взуттям.

Якщо говорити в загальному, Oracle Application Express (APEX) — це сучасна low-code платформа для створення вебдодатків на основі реляційної бази даних Oracle. Вона розроблена спеціально для швидкої побудови прикладних систем, які працюють безпосередньо з даними. У середовищі APEX реалізовано повний цикл створення бізнес-додатку — від проєктування структури бази до формування інтерактивного інтерфейсу користувача, побудови звітів, графіків, управління правами доступу та публікації готового рішення в вебформаті.

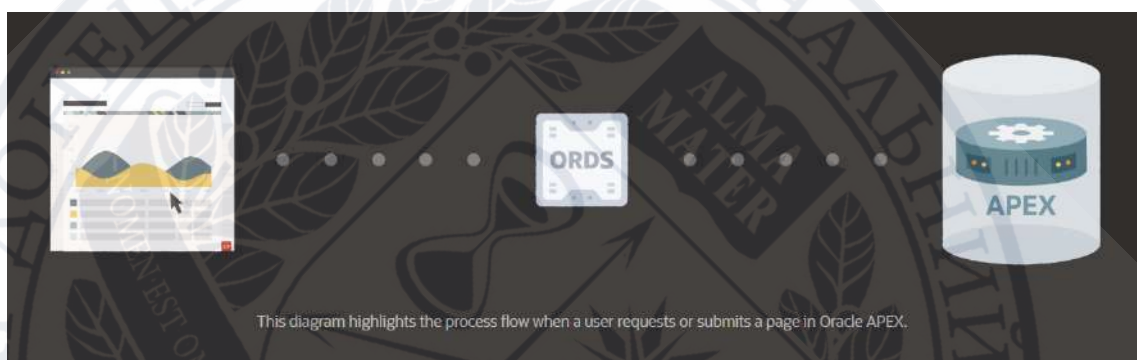


Рисунок 2.1 - Процес взаємодії користувача з вебдодатком Oracle APEX[22]

На рисунку 2.1 наочно показано, як саме влаштована робота вебдодатків платформи: запит із браузера надходить через ORDS до APEX Engine, який зчитує структуру додатку і формує сторінку для повернення у браузер.

Такий підхід дозволяє уникнути розробки окремо інтерфейсу і серверної частини, оскільки вся бізнес-логіка, інтерфейс і робота з даними виконуються безпосередньо в межах бази даних. Це суттєво полегшує підтримку та прискорює розробку, тому що розробник працює у єдиному середовищі, використовуючи візуальні інструменти та PL/SQL, що знижує технічний поріг входу та економить час, полегшує підтримку.

Ще однією суттєвою перевагою платформи Oracle APEX є її повна сумісність із хмарними інфраструктурами, зокрема з екосистемою Oracle Cloud. Це розширює можливості розробника та підприємства, дозволяючи обрати оптимальну модель розгортання — локальну, хмарну або гібридну. Усе це робить

APEX не лише засобом швидкої low-code розробки, а й інструментом, здатним інтегруватись у сучасну IT-архітектуру, де гнучкість, доступність і масштабованість є ключовими вимогами до складських систем.

Як уже зазначалося в підпункті 1.1, моделі Database as a Service (DBaaS) набувають все більшої популярності завдячуючи своїй гнучкості, економічності та простоті масштабування. У цьому контексті Oracle Cloud Infrastructure (OCI) виступає одним із провідних рішень для розміщення додатків Oracle APEX. Працюючи через OCI, користувачі отримують у розпорядження повністю автономну базу даних із попередньо налаштованим середовищем APEX. Це дозволяє уникнути необхідності самостійного встановлення, налаштування та обслуговування серверів, що суттєво знижує поріг входу навіть для невеликих команд. Автономна база даних в OCI автоматично керує оновленнями, масштабуванням, резервним копіюванням і забезпеченням безпеки даних.

Завдяки глибокій інтеграції між APEX і OCI вебдодатки можуть швидко розгортатися у хмарному середовищі, забезпечуючи високу доступність, відмовостійкість і можливість оперативного розширення ресурсів відповідно до потреб бізнесу. Таким чином, використання Oracle APEX у поєднанні з Oracle Cloud Infrastructure створює умови для створення сучасних масштабованих рішень без значних інвестицій у фізичну інфраструктуру.

Однією з головних переваг Oracle APEX є тісна інтеграція з базою даних Oracle. Усі операції — збереження, обробка, фільтрація даних — виконуються безпосередньо на рівні СУБД, що значно підвищує швидкість та надійність роботи системи. Такий підхід дозволяє уникнути проблем, пов'язаних із синхронізацією між фронтендом і бекендом, а також мінімізувати час відгуку додатку.

У контексті складських систем це особливо актуально, адже велика кількість щоденних операцій потребує оперативної обробки та точного збереження інформації. Наприклад, облік надходжень, оновлення залишків, ведення партій, інвентаризації — усі ці дії мають фіксуватись в базі в реальному часі, без затримок і дублювання. Використання Oracle APEX забезпечує

транзакційну цілісність даних, контроль унікальності записів, автоматичні перевірки та валідацію форм без потреби написання складного коду.

Ще однією важливою перевагою є гнучкий механізм побудови інтерфейсу. Завдяки вбудованим шаблонам, віджетам і конструкторам форм, навіть складні структури можна реалізувати швидко та інтуїтивно. Система підтримує створення багаторівневих панелей, динамічних звітів, діаграм, інтерактивних таблиць та фільтрів. Це дозволяє створити зручний для користувача інтерфейс, адаптований під специфіку складських задач.

Наприклад, оператору складу можна надати доступ до форми реєстрації надходжень і видач, з автоматичною перевіркою кількості товару та фіксацією дати. Менеджер побачить лише узагальнені звіти по групах товарів. Бухгалтер — історію операцій з можливістю вивантаження у формат Excel або PDF. Така розмежованість досягається завдяки гнучкій системі ролей і авторизації, яку Oracle APEX підтримує на рівні ядра.

Ще одним вагомим аргументом на користь використання Oracle APEX є зменшення часу та ресурсів, необхідних для навчання персоналу. Завдяки інтуїтивному інтерфейсу та гнучкому налаштуванню форм, система легко адаптується до звичних для працівників сценаріїв роботи. Багато функцій реалізовано у вигляді візуальних елементів — списків, перемикачів, вкладок — що значно знижує ймовірність помилок при введенні або редагуванні даних. Це особливо важливо для підприємств, де користувачі не мають спеціальної технічної підготовки.

Крім того, Oracle APEX підтримує повну локалізацію інтерфейсу, що дозволяє створювати системи з урахуванням мовних та культурних особливостей користувачів. У межах складського обліку це дозволяє уникнути непорозумінь, пов'язаних із термінологією, одиницями вимірювання, форматами дати та валюти, що є критичним для точного ведення документації.

Платформа також дозволяє впроваджувати внутрішні стандарти підприємства — як у вигляді формалізованих процесів, так і в структурі інтерфейсу. Це означає, що система може бути налаштована таким чином, щоб

відображати корпоративну політику обліку, контрольні етапи в роботі з товаром або автоматичні перевірки даних перед проведенням операцій. Наприклад, можна створити обов'язкову перевірку залишків перед відвантаженням або сповіщення про перевищення лімітів запасів.

На відміну від багатьох конкурентних рішень, Oracle APEX забезпечує довгострокову стабільність без необхідності частих змін архітектури або платформи. Це особливо вигідно в умовах обмеженого бюджету, коли важливо не лише швидко впровадити систему, а й забезпечити її надійне функціонування протягом тривалого часу без дорогих доопрацювань або повної заміни.

Крім того, платформа дозволяє розміщувати додатки як у хмарному середовищі (Oracle Cloud), так і локально, на сервері підприємства. Це забезпечує гнучкість у виборі моделі впровадження, особливо коли йдеться про конфіденційність або політику збереження даних. Багато організацій, зокрема державні або виробничі підприємства, не можуть використовувати хмарні сервіси через вимоги до безпеки — для них локальне розгортання APEX є ідеальним компромісом між контролем і зручністю.

Окремо варто згадати про підтримку мобільних пристроїв. Створені в APEX додатки адаптуються до екранів планшетів і смартфонів без потреби писати окремі версії. Це відкриває нові можливості для складської автоматизації — наприклад, працівник складу може одразу внести інформацію через мобільний пристрій, не повертаючись до робочого місця.

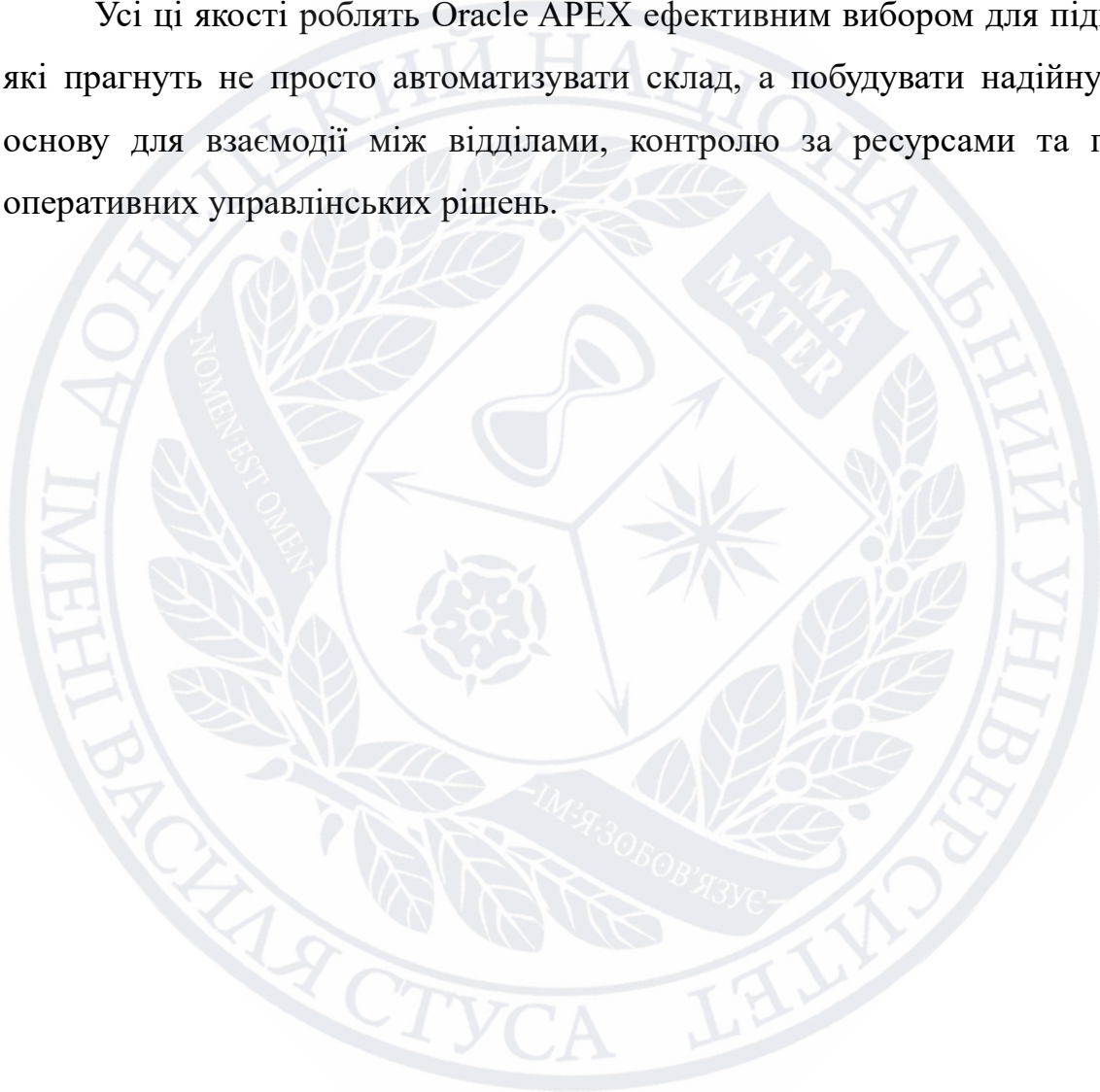
Щодо підтримки та розвитку — Oracle активно підтримує платформу APEX, регулярно випускаючи оновлення, які додають новий функціонал, виправляють вразливості та покращують продуктивність. Спільнота користувачів APEX також зростає, тож знайти приклади, навчальні матеріали або готові рішення не становить труднощів.

Таким чином, вибір Oracle APEX для розробки вебдодатку управління складом є технічно й організаційно обґрунтованим. Він забезпечує:

- швидкий старт і низький поріг входу завдяки low-code підходу;
- високу продуктивність і надійність завдяки роботі безпосередньо з БД;

- гнучкий інтерфейс для різних типів користувачів;
- централізоване управління доступом і безпекою;
- можливість локального або хмарного розгортання;
- підтримку мобільних пристроїв;
- легкість у масштабуванні, адаптації та підтримці без значних технічних знань.

Усі ці якості роблять Oracle APEX ефективним вибором для підприємств, які прагнуть не просто автоматизувати склад, а побудувати надійну цифрову основу для взаємодії між відділами, контролю за ресурсами та прийняття оперативних управлінських рішень.



РОЗДІЛ 3

РОЗРОБКА ВЕБДОДАТКУ ДЛЯ УПРАВЛІННЯ СКЛАДОМ

3.1. Проектування бази даних для складу одягу за допомогою Oracle Database

Для ефективного обліку надходження, зберігання, вибуття товарів, а також ведення взаєморозрахунків із контрагентами на складі одягу було спроектовано реляційну базу даних. Структура бази даних забезпечує централізоване зберігання всієї інформації про товари, операції, контрагентів, працівників та супровідні дані, необхідні для управління складськими процесами.

Основною метою проектування цієї бази даних було створення універсальної, масштабованої та нормалізованої структури, яка б мінімізувала дублювання даних, забезпечила цілісність інформації та дозволила оперативно проводити аналітику складських процесів. Особлива увага приділялася специфіці обліку товарів легкої промисловості (одяг та взуття), що передбачає детальну класифікацію за категоріями, статтю, сезоном та розмірами.

В основі структури лежать принципи класичної нормалізації баз даних (до третьої нормальної форми включно), що дозволяє забезпечити мінімізацію надлишковості даних, логічну цілісність і підвищити швидкість обробки запитів.

База даних має трирівневу логічну архітектуру:

- Рівень довідників: містить незмінні або рідко змінювані дані — категорії товару, категорії виробників, статі, сезони, розміри, контрагентів, працівників.
- Рівень документів та операцій: охоплює облік накладних на прихід та витрату товарів, а також детальний облік операцій над товарними позиціями.
- Рівень допоміжних даних: реалізує механізми для зберігання контактної інформації контрагентів і працівників, типізацію контактів та уточнення ролей співробітників.

Така побудова забезпечує масштабованість системи, дозволяючи легко розширювати базу при зростанні номенклатури товарів, кількості контрагентів чи обсягу операцій.

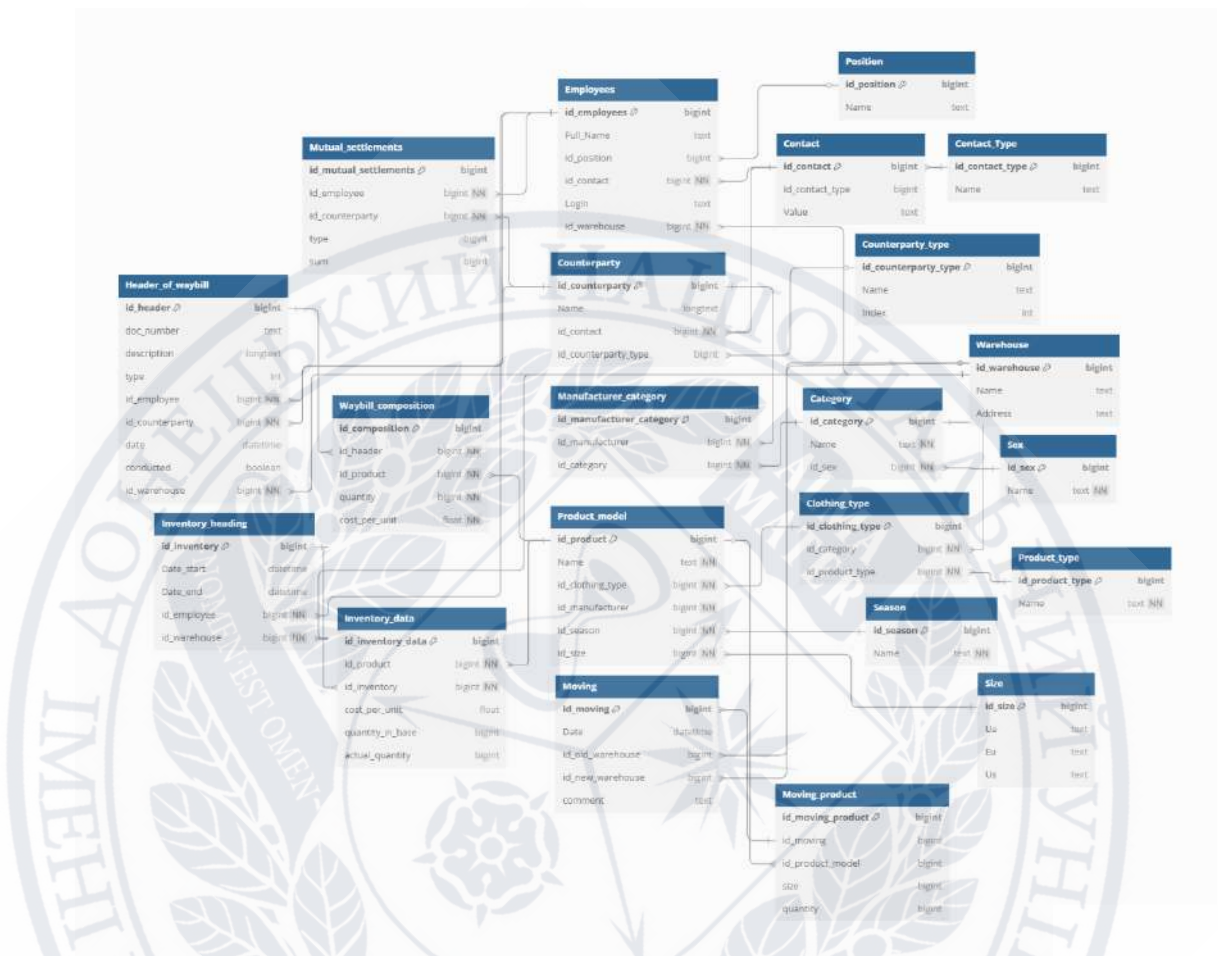


Рисунок 3.1 – Логічна структура розробленої бази даних

Далі детально розглянемо ключові сутності розробленої бази даних (рис. 3.1) для складу одягу.

Ключові сутності — товари, постачальники, накладні, операції — реалізовані у вигляді окремих таблиць із чіткими зв'язками через зовнішні ключі. Така структура дозволяє забезпечити гнучкість і масштабованість: зміну номенклатури продукції можна здійснювати без повної перебудови бази даних. У більшості випадків достатньо оновити або розширити вміст довідників (наприклад, категорії товарів, сезонність, розміри тощо), не змінюючи логіку зберігання операцій, накладних і взаєморозрахунків.

Разом із тим, варто зазначити, що у випадку повного переходу на новий тип продукції — наприклад, від одягу до побутової техніки чи будівельних матеріалів — може виникнути потреба у зміні структури самої таблиці товари, зокрема додаванні нових полів, специфічних для обраної категорії продукції. Проте навіть у цьому випадку архітектура системи залишається стабільною: усі інші таблиці, пов'язані з обліком, документообігом, фінансами та персоналом, залишаються незмінними, що значно спрощує адаптацію та супровід системи.

Таким чином, спроектована структура бази даних характеризується високим ступенем узагальненості, що забезпечує її придатність для обліку різних типів товарів у широкому спектрі комерційних та виробничих підприємств.

Окремої уваги заслуговує і той факт, що розділення даних за функціональними зонами — довідковою, транзакційною — не тільки покращує логіку зберігання інформації, а й позитивно впливає на продуктивність системи. Зокрема, операції додавання або зміни записів у таблицях накладних та складу не зачіпають структуру довідників, що дозволяє уникати блокувань і забезпечує стабільну роботу при одночасній взаємодії великої кількості користувачів. Такий підхід також спрощує виконання резервного копіювання, тестування та оновлення компонентів системи — наприклад, можна розширити логіку обробки розмірів або категорій товарів без ризику порушення функціональності інших модулів.

Розроблена логічна структура бази даних є цілісною, узгодженою та відповідає вимогам сучасного реляційного проектування. Усі сутності побудовані з дотриманням принципів нормалізації, зокрема третьої нормальної форми (3НФ) та форми Бойса-Кодда (BCNF), що дозволяє уникнути аномалій оновлення, вставки та видалення даних. Ключові таблиці поділено на транзакційні, довідникові та аналітичні, кожна з яких виконує чітко визначену функцію в системі.

3.2. Розробка вебдодатку за допомогою Oracle APEX

На основі спроектованої логічної структури бази даних було реалізовано вебдодаток для управління складом з використанням платформи Oracle APEX.



Рисунок 3.2 - Структурне представлення основних компонентів вебдодатку

Інтерфейс вебдодатку, структура якого представлено на рис. 3.2, охоплює всі основні процеси складського обліку: облік надходжень та вибуття товарів, перегляд стану складу, роботу з контрагентами, генерацію накладних та інвентаризацію. Також реалізовано механізми фільтрації, сортування, пошуку, а для кожної таблиці — окремий довідник і форма редагування.

Розглянемо основні складові розробленого вебдодатку:

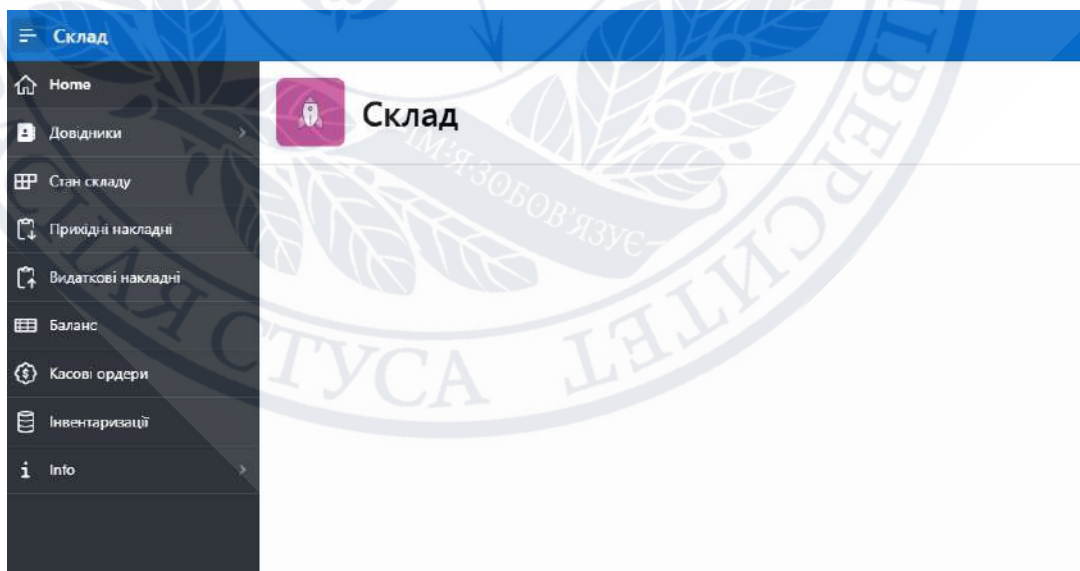


Рисунок 3.3 – Початкове вікно

Стартове вікно додатку, представлене на рис. 3.3, зустрічає користувача одразу при вході в систему. Це вікно в стандартних налаштуваннях містить в собі назву вебдодатку та панель всіх решти вікон – меню навігації. За бажанням воно

може бути змінене – наповнене брендовою символікою, інформацією про підприємство, тощо.

В основі кожного вікна, навіть початкового, лежить панель розробки (рис. 3.4), яка і визначає вигляд, функціональність, наповнення та й поведінку кожного вікна. Воно доступне лише в процесі розробки, і туди не можуть потрапити користувачі системи. Розглянемо детальніше елементи панелі.

Ліва панель є деревом сторінки. Ця структура складається з:

- Pre-Rendering / Post-Rendering — секції для коду, який виконується до або після побудови сторінки;
- Components — головних елементів сторінки.
- Breadcrumb Bar → Склад — елемент "хлібних крихт" (Breadcrumb), що показує, де користувач знаходиться в навігації по сторінках;
- Body — тіло, тобто основна частина сторінки, куди додаються регіони з вмістом.

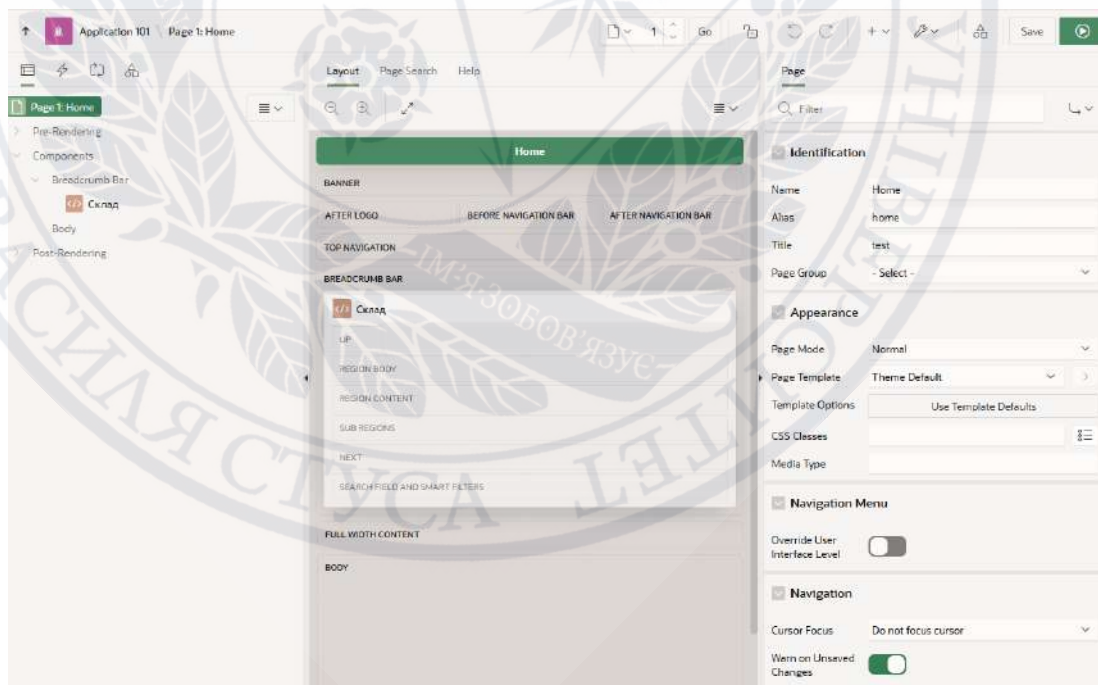


Рисунок 3.4 – Панель розробки вікна Oracle APEX

Центральна панель — макет сторінки (Layout), яка відображає візуальну структуру сторінки.

- Banner / Navigation Bar — зона верхньої панелі та банера;

- Breadcrumb Bar — зона для панелі навігації всього додатку. Його можна переглянути детально в панелі Shared Components;
- Region Body, Region Content — зарезервовані області для вмісту сторінки;
- Search Field and Smart Filters — компоненти, які можна додати для пошуку або фільтрування;
- Full Width Content / Body — область основного вмісту, яку можна наповнювати формами, звітами тощо.

Кожен елемент сторінки інтерактивний, його можна перетягувати в різні комірки, і отримувати зміни у вікні в додатку.

З правої сторони на рисунку – панель властивостей сторінки. Тут можна задавати правила її появи, її назву, можна вказувати додаткові CSS класи, медіа типи, параметри кешування, і ще безліч властивостей. Також, це поле існує і для кожного окремого елемента на сторінці. Це дає дуже великі можливості до розробки та в загальному можливостей вебдодатку.

При створенні кожного вікна є початкова панель налаштування, яка показана на рис. 3.5.

Рисунок 3.5 – Вікно налаштувань нового вікна

За допомогою цього кроку можна одразу визначити, яке саме вікно потрібне, як воно буде називатись, яка буде його позиція в панелі навігації.

В Oracle APEX також є можливість data binding – автоматичного зв'язування даних. За допомогою цього можна створювати форми, що

автоматично взаємодіють з таблицями бази даних, і система самостійно створить відповідні поля, відображення, кнопки збереження та валідацію, тобто перевірку.

Деякі елементи вікна, або ж і цілі вікна були зроблені за допомогою SQL – запитів, як, наприклад, зображений на рис. 3.6 елемент сторінки Стан складу, де відображається загальний звіт про наявність товарів на складі з розбивкою за розміром і виробником. Цей SQL-запит виконує агреговану вибірку з таблиці Operation, об'єднаної з таблицею Sizes, і повертає сумарну кількість товарів по кожній комбінації назви товару, виробника та розміру.

Ще одна важлива перевага – це SQL запити, які пишуться у панелі розробки. Вони автоматично синтаксично перевіряються, і система сама не дасть зберегти неправильний, а навіть підсвітить і вкаже, що саме неправильно. Це значно спрощує життя розробнику, який відразу зможе виправити запит на місці.

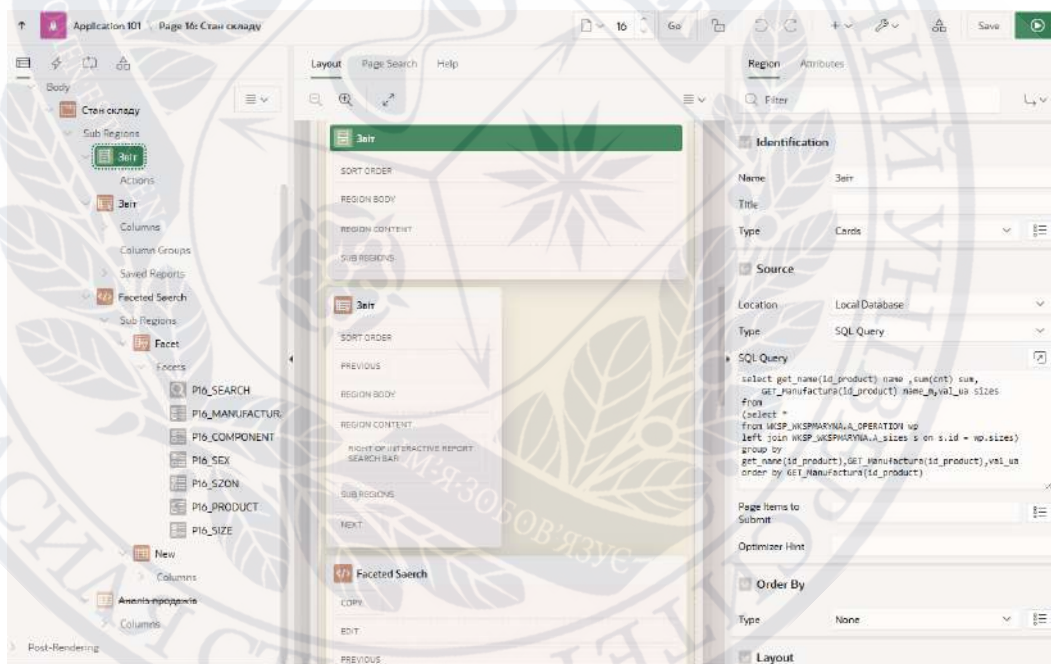


Рисунок 3.6 – Налаштування елемента Звіт вікна Стан складу

3.3. Функціональні можливості вебдодатку управління складом

На основі розробленої бази даних та інструментів Oracle APEX був розроблений вебдодаток, який охоплює ключові аспекти роботи складу.

Всі розроблені довідники, для зручності, в панелі навігації можна приховати. На рис. 3.7 можна побачити панель навігації саме по довідникам.

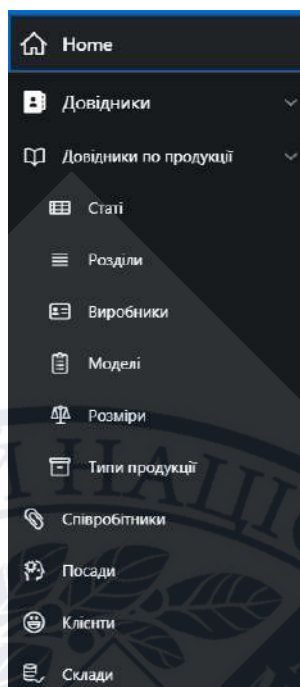


Рисунок 3.7 – Панель навігації простих довідників

Всі прості довідники мають схоже представлення в додатку – витяг з таблиці в базі даних. Так, наприклад, на рис. 3.8 можна побачити дані з двох таблиць – виробники та їх категорію, в якій вони постачають товар.

Виробники	
Назва	Категорія
Виробник1	Одяг Чоловіки
Виробник2	Одяг Жінки
Виробник3	Одяг Діти к.
Виробник4	Одяг Діти д.
Виробник5	Одяг Жінки
Виробник6	Одяг Діти д.
Виробник1	Одяг Чоловіки
Виробник2	Одяг Жінки
Виробник3	Одяг Діти к.
Виробник4	Взуття Чоловіки

Рисунок 3.8 – сторінка виробників

На цій сторінці можна додавати відомості як про виробника, так і про його категорію натиснувши відповідну кнопку Створити. Для зручності відображення, така інформація як контакти виробника приховані, проте її завжди можна відобразити, натиснувши Action. У випадяючому меню можна вибрати всю інформації, яку користувач хоче бачити.

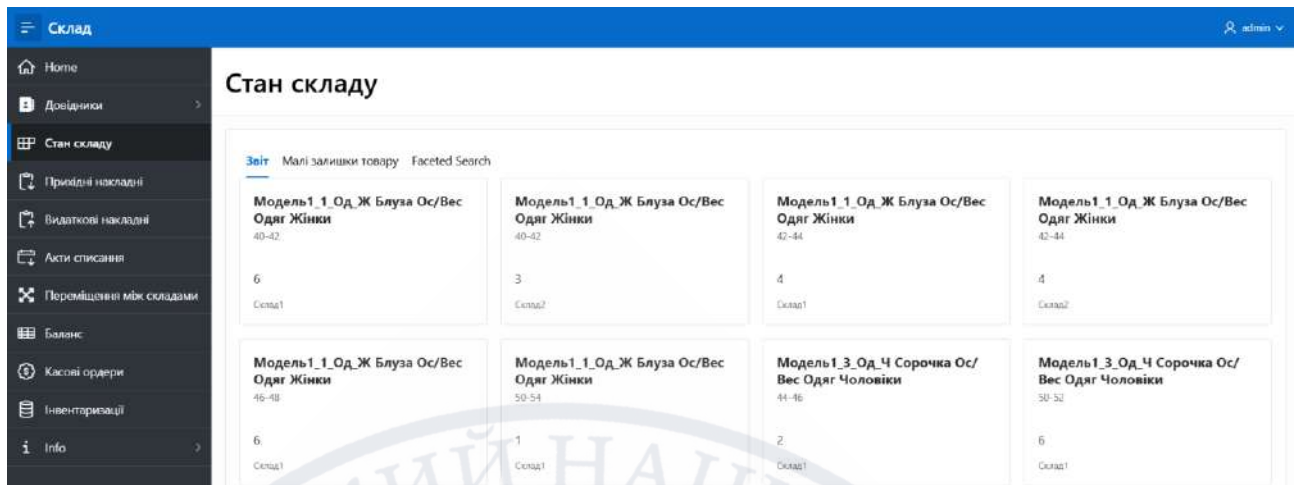


Рисунок 3.9 – Вікно стану складу

Щодо інших сторінок вебдодатку – їх структура відрізняється. На рис. 3.9 можна побачити стан складу. В цьому вікні не можна змінювати інформацію – це звіт який складається з карток моделей одягу. Можна переглянути всю ключову інформацію по складу – тобто скільки, якої моделі, якого розміру є на складі. В додаткових вкладках реалізований зручний пошук конкретних даних, з можливістю фільтрувати інформацію. Також в іншому вікні (рис. 3.10) реалізоване сповіщення про критичні пороги запасів одягу. Такі рядки автоматично підсвічуються рожевим кольором, що дозволяє оперативно реагувати на необхідність поповнення запасів. Порогове значення (у цьому випадку $\text{Sum} \leq 3$) можна змінювати відповідно до специфіки підприємства — воно задається у фільтрах таблиці та може бути налаштоване як статично, так і динамічно.

Стан складу

Звіт **Звіт** Faceted Search

Go Actions

Sum <= 3

Name	Sum	Name M	Sizes
Модель1_1_Од_Ж Блуза Ос/Вес Одяг Жінки	9	Виробник1	40-42
Модель1_1_Од_Ж Блуза Ос/Вес Одяг Жінки	8	Виробник1	42-44
Модель1_1_Од_Ж Блуза Ос/Вес Одяг Жінки	6	Виробник1	46-48
Модель1_1_Од_Ж Блуза Ос/Вес Одяг Жінки	1	Виробник1	50-54
Модель1_3_Од_Ч Сорочка Ос/Вес Одяг Чоловіки	2	Виробник1	44-46
Модель1_3_Од_Ч Сорочка Ос/Вес Одяг Чоловіки	6	Виробник1	50-52

Рисунок 3.10 – Порогове значення певних моделей

Цей функціонал значно спрощує моніторинг залишків, мінімізує ризик відсутності товару на складі й дозволяє приймати управлінські рішення своєчасно.

Реалізовано в вебдодатку можливість створювати накладні, та контролювати баланс взаєморозрахунків з контрагентами.

Id	Виробник	Номер документа	Відповідальний працівник	Коментар	Дата	Загальна сума	Чи проведена	Автор	Storage
381	Виробник1	ПН-002	Шеня Катерина		17.02.2025	12540	Проведено		Склад1
101	Виробник1	ПН-001	Шеня Катерина	Тестова накладна	19.02.2025	26400	Проведено		Склад1
243	Виробник4	ПН-0002	Шеня Катерина		28.02.2025	29800	Проведено	Адмін Адмін	Склад1
425	Виробник3	303	Козуба Віталій		06.03.2025	65900	Проведено	Кравченко Микола	Склад1
141	Виробник4	ПН-02	Ткаченко Ангеліна		15.02.2025	6250	Не проведено	Ткаченко Ангеліна	Склад1
241	Виробник3	ПН-3	Шеня Катерина		17.02.2025	14300	Проведено	Адмін Адмін	Склад1
421	Виробник1	303	Шеня Катерина		03.04.2025	50000	Проведено	Адмін Адмін	Склад1
423	Виробник1	402	Ткаченко Ангеліна		04.03.2025	123000	Проведено	Ткаченко Ангеліна	Склад1

Рисунок 3.11 – Вікно накладних

За допомогою вебдодатку можна створювати накладні – прибуткові та видаткові. На рис. 3.11 і показано всі створені прибуткові накладні, з візуальним виділенням – зелене для проведених та червоне – для тих, що чекають проведення. При натисненні на id відповідної накладної можна переглянути її деталі. Коли товари з накладної вже зараховані на склад – тобто вона проведена – зміни вносити не можна, інформація лише для перегляду.

Автор — це користувач, який створив запис у системі. Зазвичай це менеджер або оператор, який вводить інформацію про документ у базу даних. Автор фіксується автоматично на основі облікового запису, під яким виконано вхід у систему.

Відповідальний працівник — це особа, яка безпосередньо виконує або контролює складську операцію: приймає товар, перевіряє відповідність, підписує документ. Саме цей працівник є ключовою контактною особою у випадку подальших перевірок, уточнень або розслідувань щодо товару.

Важливо, щоб накладна не висіла довго в статусі “Не проведена”. Тому, якщо присутні такі випадки, система автоматично виводить повідомлення про те, що необхідно провести дії з такими накладними.

Баланс

Взаєморозрахунки

Пошук: Go [Додати касовий ордер](#)

Actions:

Тип	Виробник	Номер документа	Дата	Сума
Прихід товарів	Виробник1	421	03.03.2025	50000
Прихід товарів	Виробник1	401	05.04.2025	15170
Прихід товарів	Виробник1	181	17.02.2025	12640
Прихід товарів	Виробник1	101	13.02.2025	26490
Видатковий касовий ордер	Виробник1	107	16.02.2025	-2400
Видатковий касовий ордер	Виробник1	106	16.02.2025	-20000
Прихід товарів	Виробник2	426	06.03.2025	20340

Баланс

Пошук: Actions:

Виробник	Борг
Виробник5	82209
Виробник4	133000
Покупець3	-3360
Виробник1	81810
Покупець2	-78150
Виробник6	34300

Рисунок 3.12 – Сторінка взаєморозрахунків з контрагентами

Для забезпечення прозорого обліку взаєморозрахунків, були створені сторінки Баланс (рис. 3.12) та Касові ордери. У першому вікні можна переглядати всю історію взаєморозрахунків, оцінити всі грошові борги, а також фільтрувати по контрагентам. А у вікні Касові ордери є можливість подивитись детальнішу інформацію про кожну операцію – наприклад, хто є автором.

Наступний реалізований функціонал – інвентаризації. Рис. 3.13 демонструє вікно інвентаризацій, дозволяє переглянути інформацію про кожну проведену, та створити нову.

Інвентаризації

Пошук: Go Actions: [Create](#)

ID	Дата початку	Дата кінця
1	20.02.2025	21.02.2025
86	09.03.2025	
102	23.05.2025	
21	20.02.2025	
101	23.05.2025	

1 - 5

Рисунок 3.13 – Інвентаризації складу

При створенні нової інвентаризації система автоматично вивантажує поточний стан залишків зі складу — тобто ту кількість товарів, яка зафіксована в базі на момент початку інвентаризації. Відповідальний працівник безпосередньо на складі вносить фактичну кількість одиниць, наявних фізично. Після цього система порівнює обидва значення: якщо кількість збігається — рядок

підсвічується зеленим кольором; якщо товару менше, ніж за даними — червоним, що свідчить про недостачу; якщо більше — синім, що означає перевищення. Такий візуальний механізм дозволяє швидко виявити розбіжності між обліковими і фактичними даними та полегшує подальший аналіз.

Вікно Info містить в собі багато функціоналу. По – перше, це аналіз руху товару.

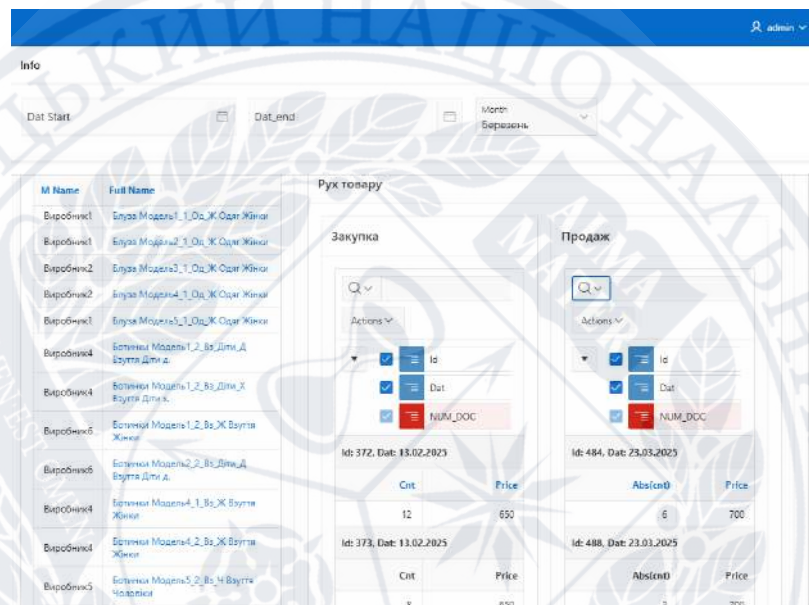


Рисунок 3.14 – Рух товару

Після вибору конкретного товару система виводить усі пов'язані з ним операції у двох окремих блоках: «Закупка» та «Продаж». У кожному блоці на рис. 3.14 можна побачити дату, номер документа, кількість товару та ціну. Таким чином, сторінка надає зручний і наочний спосіб відстеження руху товару: коли він надходив, за якою ціною, і коли був реалізований. Це суттєво полегшує аналіз попиту та контроль залишків, особливо у розрізі окремих товарів або періодів.

Наступна функціональна можливість вікна – це календар накладних, зображений на рис. 3.15.

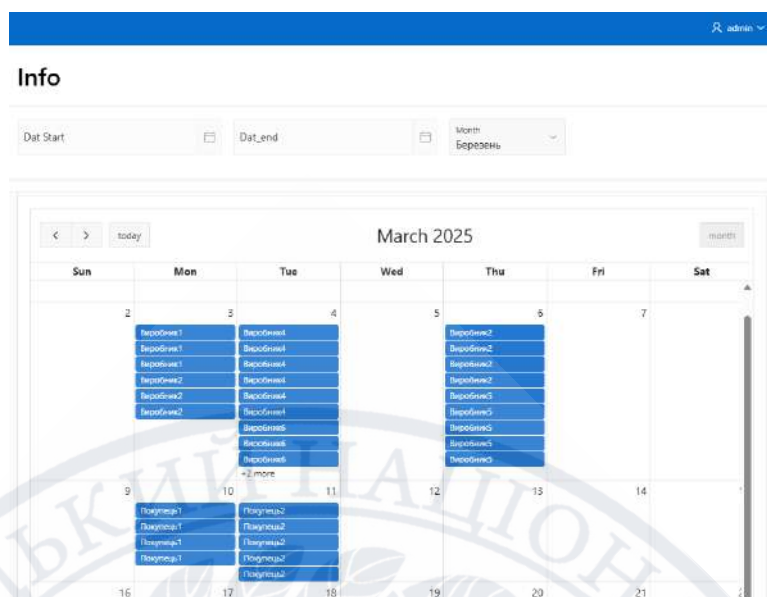


Рисунок 3.15 – Календар накладних за березень

Такий календар значно спрощує управлінський контроль та планування, адже дає змогу швидко оцінити активність поставань і продажів у кожен конкретний день.

Синім кольором відображаються записи по кожному з постачальників або покупців. Це дозволяє миттєво побачити, коли і з ким відбувалися операції. За допомогою додаткових фільтрів (дат початку, завершення, місяця) користувач може звужити вибірку до потрібного періоду або лише певного контрагента.

Функціонал цієї сторінки дозволяє:

- проводити візуальний аналіз завантаженості складу;
- виявляти дні з підвищеною або зниженою активністю;
- планувати майбутні необхідні надходження/відвантаження;
- виявляти затримки або нерівномірність у поставаннях;
- ефективніше координувати логістику та ресурси персоналу.

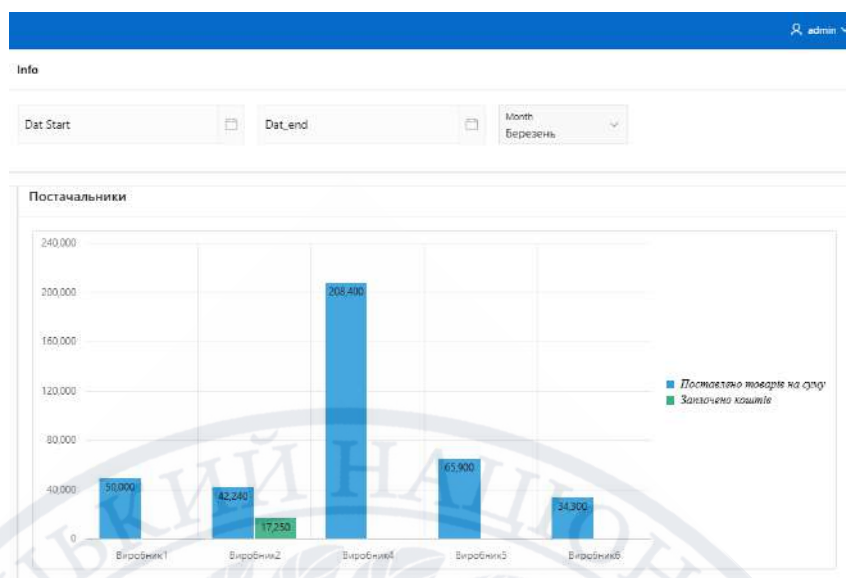


Рисунок 3.16 – Графік взаєморозрахунків з постачальниками

На рис. 3.16 представлено графік на сторінці додатку, що відображає фінансову активність постачальників у вигляді порівняння загальної вартості поставленого товару та фактично сплачених коштів. Цей інструмент дає змогу швидко виявити дисбаланс між постачанням і оплатою.

Гістограма показує дані по кожному виробнику: сині стовпчики — це загальна сума товарів, які були поставлені за обраний період, а зелені — скільки коштів уже було сплачено. Наприклад, для Виробника 2 видно, що частина суми ще не оплачена, а для решти потрібно віддати повну суму за всі накладні березня. Такі самі графіки реалізовано для покупців.

Також для співробітників виведена діаграма з кількістю прийнятих накладних, кількості позицій в них. Кількість оброблених накладних і позицій дозволяє визначити, наскільки рівномірно розподілене навантаження між співробітниками. Якщо один працівник систематично оформлює більшість документів, це може бути як показником ефективності, так і сигналом про ризики перевантаження.

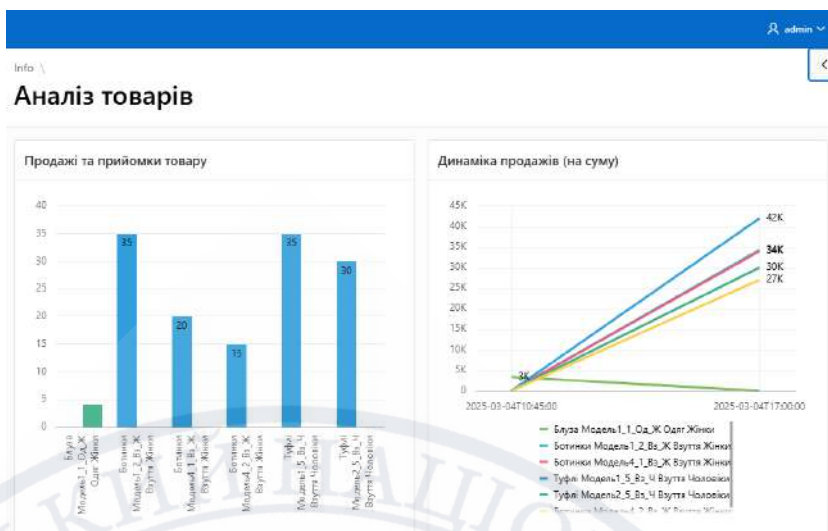


Рисунок 3.17 – Аналітична панель “Аналіз товарів”

На представлених графіках на рис. 3.16 реалізовано інструментарій для візуалізації даних, що підтримує прийняття управлінських рішень. У першому вікні користувач бачить два ключові графіки — стовпчикову діаграму з даними щодо продажу та приймання товарів у розрізі моделей, а також лінійну діаграму динаміки продажів за сумою у часовому вимірі. Це дозволяє оперативно визначити найактивніші позиції, проаналізувати стабільність продажів, виявити тренди та аномалії.

Додатково, система дозволяє реалізувати й інші види графіків. Наприклад, графік рентабельності окремих товарів наочно показує співвідношення закупівельної та продажної ціни, допомагаючи виявити найбільш прибуткові моделі. А кругова діаграма з категоріями товарів дає змогу швидко зорієнтуватися, яка саме категорія користується найбільшим попитом — це важливо при прийнятті стратегічних рішень щодо асортименту або планування закупівель.

Є можливість вибрати період, або окремий місяць, товар, і провести аналітику за ним.

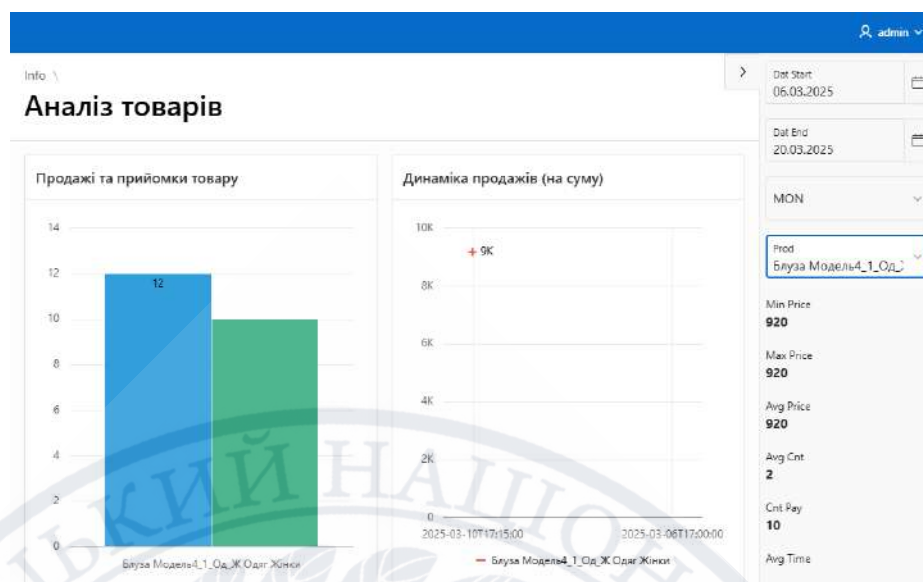


Рисунок 3.18 – Аналітика окремого товару

Якщо вибрати окремий товар, як показано на рис. 3.18 – про нього відображаються додаткові відомості, такі як ціна – максимальна, мінімальна, загальна кількість проданих і поставлених одиниць. Такий підхід дозволяє швидко оцінити прибутковість окремих позицій.

Також у додатку реалізовано різні рівні доступу, наприклад для керівника складу та адміністратора, від імені якого були продемонстровані можливості додатку, є доступ до всіх вкладок, тоді як у комерсанта – лише до необхідних йому накладних та інвентаризацій, виключаючи всі прості довідники та вкладку Аналіз товарів, які не мають відношення до його робочих обов'язків.

Реалізований вебдодаток може бути відкритий не лише з комп'ютера, а й з планшета чи телефону, що дозволяє працівникам не бути прив'язаними до певного місця, а працювати з системою з будь-якого місця.

ВИСНОВКИ

У межах бакалаврської роботи було розроблено та впроваджено вебдодаток для автоматизації обліку складських операцій на підприємстві, що спеціалізується на торгівлі одягом. У процесі виконання завдання було охоплено повний цикл створення прикладної інформаційної системи — від проектування логічної структури бази даних до реалізації інтерфейсів користувача, розгортання в середовищі Oracle APEX та наповнення системи інструментами аналітики й звітності.

На основі попереднього аналізу предметної області було спроектовано базу даних, яка відображає основні об'єкти та зв'язки складської діяльності: товари, категорії, постачальники, контрагенти, працівники, накладні, операції та інвентаризації. Особливістю побудови структури є її гнучкість — більшість змін у номенклатурі або бізнес-логіці можна реалізувати без змін до архітектури, просто оновлюючи довідники. Це забезпечує масштабованість і можливість подальшої адаптації до нових умов, наприклад, зміни типу товарів або розширення функціоналу.

Для реалізації інтерфейсної частини було обрано платформу Oracle APEX, яка дозволила значно скоротити час розробки за рахунок підтримки low-code підходу. Усі вікна додатку були створені з урахуванням принципів зручності та ефективності. Наприклад:

- розділи обліку надходжень і вибуття товарів дозволяють формувати накладні, призначати відповідальних працівників і контролювати статуси документів;
- вікно інвентаризацій реалізує фіксацію реального залишку на складі та порівняння його з базовими даними — система автоматично підсвічує розбіжності;
- блок взаєморозрахунків представлений двома логічно розділеними вікнами — "Баланс" для загального огляду ситуації по контрагентам та "Касові ордери" для детального перегляду окремих транзакцій, що дозволяє забезпечити прозорість фінансових взаємин;

- інформаційно-аналітичні розділи дають змогу візуалізувати динаміку продажів, структуру закупівель, активність співробітників, рентабельність товарів, а також переглядати календар подій по днях. Це дозволяє керівництву оперативно виявляти проблемні точки, приймати обґрунтовані рішення та ефективно планувати запаси.

Серед інших функціональних можливостей реалізовано:

- рольову модель доступу користувачів до функціоналу;
- налаштування параметрів сторінок через інтерфейс розробника;
- можливість редагування структури елементів без потреби перевиконання всієї програми;
- фільтри, сортування, пошук, вивантаження звітів у CSV, PDF;
- можливість адаптації інтерфейсу під бренд компанії (колірна схема, логотип, стартова сторінка тощо);
- можливість локального або хмарного розгортання рішення.

У підсумку, запропонований вебдодаток повністю відповідає сучасним вимогам до систем обліку на складах підприємства. Його гнучкість, функціональність та прозора логіка побудови роблять рішення на базі Oracle APЕХ потужною альтернативою як типових ERP-систем, так і складних розробок "з нуля".

СПИСОК ВИКОРИСТАНИХ ПОСИЛАНЬ

1. А. О С. Автоматизація складу - інвестиція у майбутнє підприємства. 2024. С. 195–196. URL: <https://confmanagement-proc.kpi.ua/article/view/303875>. (дата звернення: 04.04.2025).
2. Бондаренко Н. М., Климчук Л. Д. Захист інформації в базах даних. *Період трансформаційних процесів в світовій науці: задачі та виклики*. 2024. С. 322–324. URL: <https://sci.ldubgd.edu.ua/jspui/bitstream/123456789/13194/1/50-72-PB.pdf#page=322>. (дата звернення: 05.04.2025).
3. Відкритий код ERP та CRM. Odoo. Odoo. URL: https://www.odoo.com/uk_UA (дата звернення: 25.04.2025).
4. Гелешко І., Карабин О. Тенденції розвитку баз даних. *Інформаційна безпека та інформаційні технології* : Зб. тез доп. V Всеукр. науково-практ. конф. молодих уч., студентів і курсантів. 2021. С. 115–117. URL: [https://ldubgd.edu.ua/sites/default/files/3_nauka/konferenz/2021/Захист%20інформації%20\(збірник\)%20-%202021%202.0.pdf#page=115](https://ldubgd.edu.ua/sites/default/files/3_nauka/konferenz/2021/Захист%20інформації%20(збірник)%20-%202021%202.0.pdf#page=115). (дата звернення: 10.04.2025).
5. Геселева Н. В., Головачов І. А. Реляційні та нереляційні бази даних в бізнесі. 2018. С. 319–320. URL: <https://er.knutd.edu.ua/handle/123456789/11102>. (дата звернення: 10.04.2025).
6. Гніденко І., Воробйов І. Аналіз сучасних програмних продуктів для створення web-сайтів, їх переваги та недоліки. *Проблеми інформатизації та управління*. 2019. URL: <https://er.nau.edu.ua/items/de4fcc62-877e-4c8a-95b3-47ea551a0498>. (дата звернення: 13.04.2025).
7. Денисенко С., Кудрявцева К., Пруднікова К. Особливості застосування сучасних Low-Code/No-Code платформ для веб-розробок. 2025. С. 23–26. URL: <https://jrnl.nau.edu.ua/index.php/congress/article/view/19606> (дата звернення: 18.04.2025).

8. Доценко С. І. Організація та системи керування базами даних : навч. посіб. Харків, 2023. 117 с.
URL: <http://lib.kart.edu.ua/bitstream/123456789/13596/1/Навчальний%20посібник.pdf>. (дата звернення: 13.04.2025).
9. Єрукаєв А. В., Отрошко П. А., Касянчук А. В. Субд “oracle” та її роль в управлінні даними. *International scientific innovations in human life : Proceedings of XI International Scientific and Practical Conference*, м. Manchester. 2022. С. 213–216.
URL: <https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/41120/1/Манчестер.pdf#page=213>. (дата звернення: 15.04.2025).
10. Карпішен Б. С., Перепелиця А. С. Робота з реляційною моделлю даних. *Інформаційні технології у сучасному світі* : матеріали Міжнар. науково-практ. конф. здобувачів вищ. освіти і молодих вчен. Харків, 2024. С. 271–275. URL: <https://repo.btu.kharkov.ua/handle/123456789/53647>. (дата звернення: 15.04.2025).
11. Клімов О. Е., Дружиніна В. В. Іт-системи в логістиці – найважливіші технології. *XXXI Міжнародна науково-практична конференція студентів, аспірантів та молодих учених «Актуальні проблеми життєдіяльності суспільства»*. 2024. URL: <https://doi.org/10.32782/2222-5099.2024.7.4> (дата звернення: 05.04.2025).
12. Колосова К. К., Січко Т. В. Аналіз використання ERP систем. *Комп'ютерні технології обробки даних*. 2021.
URL: <https://jktod.donnu.edu.ua/article/view/11751>. (дата звернення: 05.04.2025).
13. Міщенко А. А. Інструменти розробника реляційної бази даних : Магістерська дисертація. Київ, 2024.
URL: <https://ela.kpi.ua/handle/123456789/71669>. (дата звернення: 15.04.2025).
14. Павловський В. І., Петрашенко А. В. Бази даних та засоби управління : підручник. Київ, 2024. 326 с.

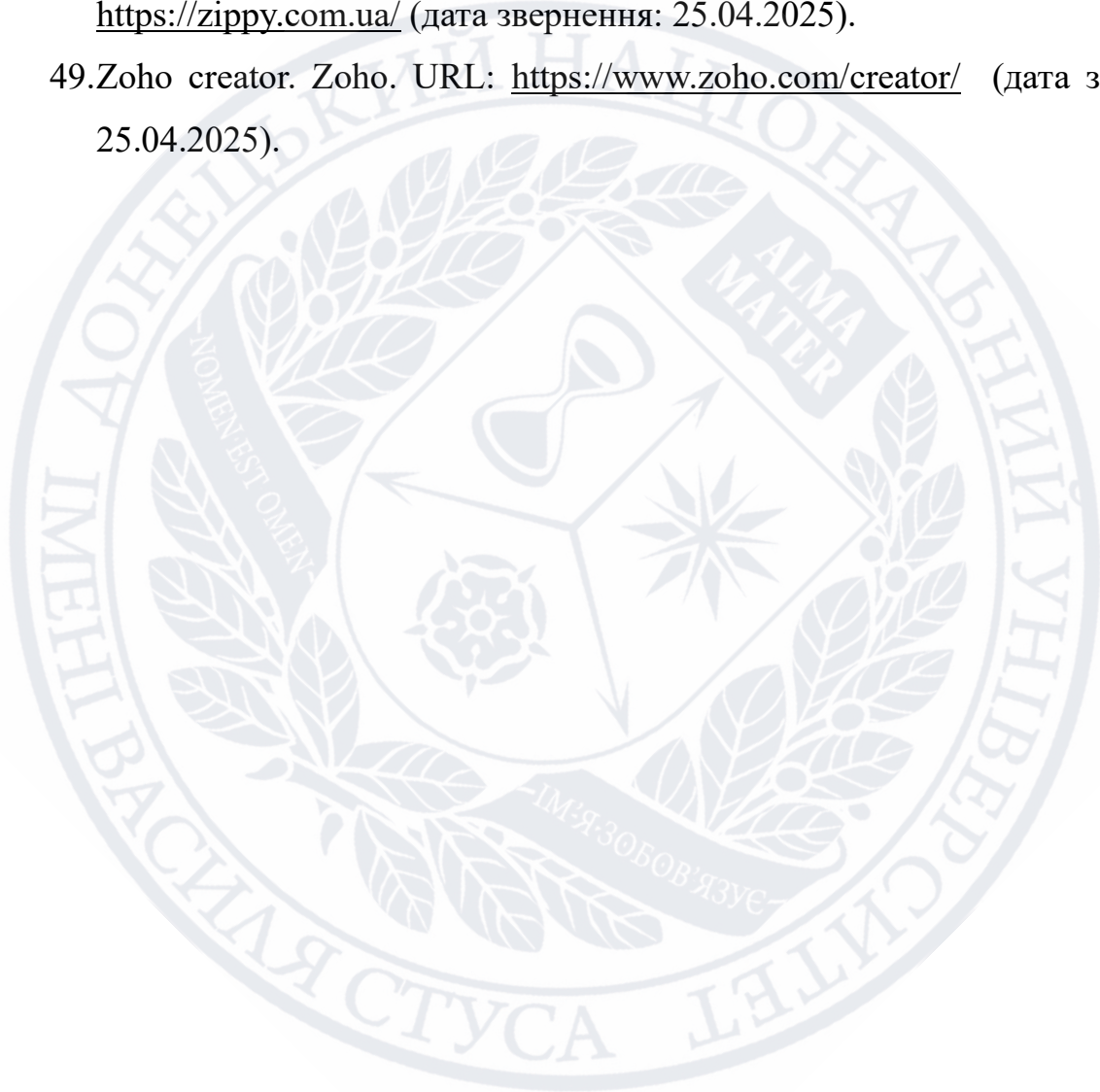
- URL: <https://ela.kpi.ua/handle/123456789/70913>. (дата звернення: 10.04.2025).
- 15.Рішення для бізнесу. Торгсофт. URL: <https://torgsoft.ua/soft/> (дата звернення: 25.04.2025).
- 16.УкрСклад. Програма Склад Україна - універсальний складський облік | Про програму. URL: <https://www.ukrsklad.com/> (дата звернення: 25.04.2025).
- 17.Чихіра І. В., Левицький В. О., Микитишин А. Етапи оптимізації баз даних. *Теоретичні та прикладні аспекти радіотехніки, приладобудування і комп'ютерних технологій* : Матеріали IV Міжнар. науково-техн. конф. „Теорет. та прикладні аспекти радіотехніки, приладобудування і комп'ютер. технологій“ присвяч. 80-ти річчю з дня народж. проф. Я.І. Проця, 2019. Тернопіль, 2019. С. 256–257. URL: <http://elartu.tntu.edu.ua/handle/lib/28848>. (дата звернення: 15.04.2025).
- 18.Чихіра І., Микитишин А., Стухляк П. Забезпечення логічної та фізичної цілісності баз даних. *Матеріали VII науково-технічної конференції „Інформаційні моделі, системи та технології“* : Матеріали VII науково-техн. конф. „Інформ. моделі, системи та технології“, 2019, м. Тернопіль. Тернопіль, 2019. С. 17. URL: <http://elartu.tntu.edu.ua/handle/lib/30376>. (дата звернення: 13.04.2025).
- 19.Штельмашук М. Цифровізація та автоматизація логістичних процесів: сучасний стан та перспективи. *Економіка та суспільство*. 2024. № 68. URL: <https://doi.org/10.32782/2524-0072/2024-68-193> (дата звернення: 05.04.2025).
- 20.A. S. Oracle APEX security: best practices for robust protection. *International journal on science and technology*. 2025. Vol. 16, no. 2. URL: <https://doi.org/10.71097/ijst.v16.i2.3721> (date of access: 19.04.2025)
- 21.Application of LCDP and technology in existing business systems in enterprise / L. Fu et al. 2024 9th international conference on communication, image and signal processing (CCISP), Gold Coast, Australia, 13–15 November 2024. 2024.

- P. 145–154. URL: <https://doi.org/10.1109/ccisp63826.2024.10765540> (date of access: 10.04.2025)
22. Architecture. Oracle APEX.
URL: <https://apex.oracle.com/en/platform/architecture/> (date of access: 19.04.2025).
23. B2C order fulfilment services. myWarehouse. URL: <https://www.orderfulfilment.co.uk/> (дата звернення: 25.04.2025).
24. Bala D. V. A pragmatic step to deploy low-code web apps on apex cloud services for emerging business assistance. *I-manager's journal on software engineering*. 2022. Vol. 16, no. 3. P. 1. URL: <https://doi.org/10.26634/jse.16.3.18697> (date of access: 20.04.2025)
25. Bock A. C., Frank U. Low-Code platform. *Business & information systems engineering*. 2021. Vol. 63, no. 6. P. 733–740. URL: <https://doi.org/10.1007/s12599-021-00726-8> (date of access: 22.04.2025)
26. Chandra D. B. Mastering PL/SQL through illustrations: from learning fundamentals to developing efficient PL/SQL blocks. BPB PUBLICATIONS, 2020. 288 p.
27. CRM apppointer. Apppointer. URL: <https://apppointer.ua/> (дата звернення: 25.04.2025).
28. Emerging trends in sales automation and software development for global enterprises / Y. Gadhiya et al. *International IT journal of research*. 2024. Vol. 2, no. 4. P. 200–208. URL: <https://itjournal.org/index.php/itjournal>. (date of access: 18.04.2025)
29. ERPNext documentation. EDocs. URL: <https://docs.erpnext.com/> (дата звернення: 25.04.2025).
30. Google AppSheet | Build apps with no code. AppSheet. URL: <https://about.appsheet.com/home/> (дата звернення: 25.04.2025).
31. InFlow inventory. inFlow. URL: <https://www.inflowinventory.com/> (дата звернення: 25.04.2025).

32. Inventory management - order fulfillment - manufacturing. Megaventory. URL: <https://www.megaventory.com/> (дата звернення: 25.04.2025).
33. Kantsedal N., Leha O., Morozov Y. Digitalization of logistics: new technologies to improve management and optimization. *Economic scope*. 2025. No. 199. P. 45–51. URL: <https://doi.org/10.30838/ep.199.45-51> (date of access: 18.04.2025)
34. Korada L. Low code/no code application development - opportunity and challenges for enterprises. *International journal on recent and innovation trends in computing and communication*. 2022. Vol. 10, no. 11. P. 209–218. URL: <https://doi.org/10.17762/ijritcc.v10i11.11038> (date of access: 22.04.2025)
35. Kuhn D., Kyte T. Expert oracle database architecture. Berkeley, CA : Apress, 2022. URL: <https://doi.org/10.1007/978-1-4842-7499-6> (date of access: 17.04.2025)
36. Kvet M. Rapid Application Development and data management using Oracle APEX and SQL. *2024 IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Stará Lesná, Slovakia, 25–27 January 2024. 2024. URL: <https://doi.org/10.1109/sami60510.2024.10432914> (date of access: 22.04.2025)
37. Low-code platforms and languages: the future of software development / G. Juhas et al. *2022 20th international conference on emerging elearning technologies and applications (ICETA)*, Stary Smokovec, Slovakia, 20–21 October 2022. 2022. URL: <https://doi.org/10.1109/iceta57911.2022.9974697> (date of access: 20.04.2025)
38. Oracle APEX as a tool for data analytics. *Good practices and new perspectives in information systems and technologies*. 2024. P. 203–214. URL: https://link.springer.com/chapter/10.1007/978-3-031-60328-0_20#citeas. (date of access: 22.04.2025)

- 39.Oracle database documentation - oracle database. *Oracle Help Center*.
URL: <https://docs.oracle.com/en/database/oracle/oracle-database/index.html>.
(date of access: 18.04.2025)
- 40.Perkumienė D., Puskunigytė K. Improvement of warehouse processes through digitalization solutions. *Humanities studies*. 2024. No. 21 (98). P. 213–219.
URL: <https://doi.org/10.32782/hst-2024-21-98-25> (date of access: 18.04.2025)
- 41.Pinho D., Aguiar A., Amaral V. What about the usability in low-code platforms? A systematic literature review. *Journal of computer languages*. 2022. P. 101185.
URL: <https://doi.org/10.1016/j.col.2022.101185> (date of access: 21.04.2025)
- 42.Podkopaiev D., Tsyfra T. Automation systems for managing relations with counterparties in construction. *Spatial development*. 2024. No. 9. P. 349–360.
URL: <https://doi.org/10.32347/2786-7269.2024.9.349-360> (date of access: 20.04.2025)
- 43.ReddyKavuluri H. V. Advanced Role-Based Access Control Mechanisms in Oracle Databases. *International Journal of AI, BigData, Computational and Management Studies*. 2024. Vol. 5, no. 3. P. 24–32.
URL: <https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I3P103>. (date of access: 28.04.2025)
- 44.Rikkula H. Leveraging oracle APEX for streamlined warehouse management in the automotive industry-in- time inventory, low-code development. *International journal of advanced research*. 2024. Vol. 1, no. 2.
URL: https://www.researchgate.net/publication/383213664_Leveraging_Oracle_APEX_for_Streamlined_Warehouse_Management_in_the_Automotive_Industry-In-_Time_Inventory_Low-Code_Development. (date of access: 22.04.2025)
- 45.Syed A. Oracle APEX development best practices: ensuring scalability and maintainability. *International journal of leading research publication*. 2024. Vol. 5, no. 4. P. 1–12. URL: <https://doi.org/10.5281/zenodo.15026525>. (date of access: 17.04.2025)

46. The next-generation app platform - Airtable. Airtable. URL: <https://www.airtable.com> (дата звернення: 25.04.2025).
47. Waszkowski R. Low-code platform for automating business processes in manufacturing. *IFAC-PapersOnLine*. 2019. Vol. 52, no. 10. P. 376–381. URL: <https://doi.org/10.1016/j.ifacol.2019.10.060> (date of access: 20.04.2025)
48. ZippyCRM. CRM система для складського обліку та торгівлі. URL: <https://zippy.com.ua/> (дата звернення: 25.04.2025).
49. Zoho creator. Zoho. URL: <https://www.zoho.com/creator/> (дата звернення: 25.04.2025).



Декларація щодо унікальності текстів роботи
та невикористання матеріалів інших авторів без посилань

ДЕКЛАРАЦІЯ

про дотримання академічної доброчесності

Я, _____

Повністю вказується ПІБ та статус (посада для працівників, освітня (освітньо-наукова) програма – для здобувачів вищої освіти)

що нижче підписалась/підписався, розуміючи та підтримуючи загальновизнані засади справедливості, доброчесності та законності,

ЗОБОВ'ЯЗУЮСЬ:

дотримуватися принципів та правил академічної доброчесності, що визначені законодавством України, локальними нормативними актами Донецького національного університету імені Василя Стуса, положеннями, правилами, умовами, визначеними іншими суб'єктами, та не допускати їх порушення.

ПІДТВЕРДЖУЮ:

що мені відомі положення статті 42 Закону України «Про освіту»;
що у даній роботі не представляла/представляв чийсь роботи повністю або частково як свої власні. Там, де я скористалася/скористався працею інших, я зробила/зробив відповідні посилання на джерела інформації;

що дана робота не передавалась іншим особам і подається вперше, не порушує авторських та суміжних прав закріплених статтями 21-25 Закону України «Про авторське право та суміжні права», а дані та інформація не отримувались в недозволений спосіб.

УСВІДОМЛЮЮ:

що ця робота може бути перевірена університетом на плагіат або інші порушення академічної доброчесності, в тому числі з використанням спеціалізованих сервісів;

що у разі порушення академічної доброчесності, до мене можуть бути застосовані процедури, передбачені законодавством України та Кодексом академічної доброчесності та корпоративної етики Донецького національного університету імені Василя Стуса, іншими локальними нормативними актами університету, та я можу бути притягнута/притягнутий до академічної відповідальності.

(дата)

(підпис)