

Міністерство освіти і науки України

Відокремлений структурний підрозділ «Тернопільський фаховий коледж
Тернопільського національного технічного університету імені Івана Пулюя»

(повне найменування вищого навчального закладу)

Відділення телекомунікаційних та електронних систем

(назва відділення)

Циклова комісія комп'ютерних наук

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи

фахового молодшого бакалавра

(освітньо-професійного ступеня)

на тему: “Розробка інформаційної системи складського обліку
будівельних матеріалів «BudSklad»”

Виконав: студентка IV курсу, групи КН-423

Спеціальності 122 Комп'ютерні науки
(шифр і назва спеціальності)

Максим КРУЦЬ

(ім'я та прізвище)

Керівник Володимир СЕРБІН
(ім'я та прізвище)

Рецензент _____
(ім'я та прізвище)

Тернопіль – 2026

ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ТЕРНОПІЛЬСЬКИЙ ФАХОВИЙ КОЛЕДЖ
ТЕРНОПІЛЬСЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
ІМЕНІ ІВАНА ПУЛЮЯ»

Відділення телекомунікацій та електронних систем
Циклова комісія комп'ютерних наук
Освітньо-професійний ступінь «фаховий молодший бакалавр»
Спеціальність 122 Комп'ютерні науки
Галузь знань 12 Інформаційні технології

ЗАТВЕРДЖУЮ

Голова циклової комісії
комп'ютерних наук

_____ Галина МАРЦІЯШ

« 02 » березня 2026 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ**

Круцю Максиму Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка інформаційної системи складського обліку будівельних матеріалів «BudSklad»

керівник роботи Сербін Володимир Сергійович,

(прізвище, ім'я, по батькові)

затверджені наказом вищого навчального закладу № 4/9-132 від 27.02.2026 р.

2. Строк подання студентом роботи: 19.06.2026 р.

3. Вихідні дані до роботи: технічне завдання на розробку програмного забезпечення, технічне завдання на розробку програмного забезпечення, мова програмування TypeScript, фреймворк Express.js, бібліотеки: React; СКБД PostgreSQL, стандарти IEEE 29148-2018, IEEE 29119, ДСТУ 8302:2015.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):

1 Загальний розділ

1.1 Аналітичний огляд існуючих рішень

1.2 Технічне завдання

1.2.1 Найменування та область застосування

1.2.2 Призначення розробки

1.2.3 Вимоги до функціоналу вебзастосунку

1.2.4 Вимоги до програмної документації

1.2.5 Техніко-економічні показники

1.2.6 Стадії та етапи розробки

1.2.7 Порядок тестування та прийому

2 Розробка технічного та робочого проекту

2.1 Розробка структури сторінок інформаційної системи

2.2 Створення та верстка сторінок інформаційної системи

2.3 Розробка структури бази даних інформаційної системи

2.4 Програмування інформаційної системи

2.4.1 Написання клієнтської частини

2.4.2 Написання серверної частини

2.5 Тестування інформаційної системи

3 Спеціальний розділ

3.1 Інструкція з розміщення інформаційної системи в Інтернеті

3.2 Інструкція з обслуговування та наповнення інформаційної системи

3.3 Інструкція з популяризації та підтримки інформаційної системи

4 Економічний розділ

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

4.3 Розрахунок витрат на електроенергію

4.4 Розрахунок суми амортизаційних відрахувань

4.5 Обчислення накладних витрат

4.6 Складання кошторису витрат та визначення собівартості системи

4.7 Розрахунок ціни інформаційної системи

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

5 Охорона праці, техніка безпеки та екологічні вимоги

5.1 Методи боротьби зі шкідливими речовинами на виробництві

5.2 Види виробничого освітлення

6 Висновки

Додаткові вказівки: виконання кваліфікаційної роботи із розробкою програмного продукту – інформаційної системи «BudSklad»..

5. Перелік графічного матеріалу:

1. Схема структурна клієнтської частини
2. UML- варіантів використання
3. ER-діаграма бази даних
4. Таблиця техніко-економічних показників

6. Консультанти розділів роботи

Розділ	Ім'я, прізвище та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний розділ	Любов КАЛУШКА		
Охорона праці, техніка безпеки та екологічні вимоги	Генадій ГОРЯЧЕК		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Отримання і аналіз технічного завдання	20.03.2026	
2	Збір і узагальнення інформації	01.05.2026	
3	Написання першого розділу	15.05.2026	
4	Розробка технічного та робочого проекту	29.05.2026	
5	Написання спеціального розділу	05.06.2026	
6	Розрахунок економічної частини	08.06.2026	
7	Написання розділу охорони праці	09.06.2026	
8	Виконання графічної частини	10.06.2026	
9	Оформлення пояснювальної записки	11.06.2026	
10	Погодження нормоконтролю	12.06.2026	
11	Попередній захист кваліфікаційної роботи	.06.2026	
12	Захист кваліфікаційної роботи	.06.2026	

7. Дата видачі завдання: 05 березня 2026 р.

Студент

_____ (підпис)

Максим КРУЦЬ

Керівник роботи

_____ (підпис)

Володимир СЕРБІН

ЗМІСТ

АНОТАЦІЯ.....	7
ABSTRACT.....	8
ВСТУП.....	9
1 ЗАГАЛЬНИЙ РОЗДІЛ.....	11
1.1 Аналітичний огляд існуючих рішень.....	11
1.2 Технічне завдання.....	15
1.2.1 Найменування та область застосування.....	15
1.2.2 Призначення розробки.....	16
1.2.3 Вимоги до функціоналу інформаційної системи.....	17
1.2.4 Вимоги до програмної документації.....	20
1.2.5 Техніко-економічні показники.....	21
1.2.6 Стадії та етапи розробки.....	22
1.2.7 Порядок контролю та прийому.....	24
2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ.....	27
2.1 Розробка структури інформаційної системи і web-сторінок.....	27
2.2 Створення та верстка сторінок інформаційної системи.....	30
2.3 Розробка структури бази даних інформаційної системи.....	44
2.4 Програмування інформаційної системи.....	48
2.4.1 Написання клієнтської частини.....	49
2.4.2 Написання серверної частини.....	52
2.5 Тестування інформаційної системи.....	53
3 СПЕЦІАЛЬНИЙ РОЗДІЛ.....	56
3.1 Інструкція з розгортання інформаційної системи.....	56
3.2 Інструкція з обслуговування та наповнення інформаційної системи.....	59
3.3 Інструкція з популяризації та підтримки інформаційної системи.....	63

					<i>2026.КВР.122.423.09.00.00 ПЗ</i>				
Зм.	Арк.	№ докум.	Підпис	Дата	<i>Розробка інформаційної системи складського обліку будівельних матеріалів «BudSklad» Пояснювальна записка</i>	Літ.	Арк.	Аркушів	
<i>Розроб.</i>		<i>Круць М.С.</i>						5	
<i>Перевір.</i>		<i>Сербін В.С.</i>							
<i>Реценз.</i>									
<i>Н. Контр.</i>		<i>Приймак В.А.</i>							
<i>Затверд.</i>								<i>ВСП ТФК ТНТУ КН-423 м. Тернопіль</i>	

4	ЕКОНОМІЧНИЙ РОЗДІЛ.....	66
4.1	Визначення стадій технологічного процесу та загальної тривалості проведення НДР	66
4.2	Визначення витрат на оплату праці та відрахувань на соціальні заходи	67
4.3	Розрахунок витрат на електроенергію	69
4.4	Розрахунок суми амортизаційних відрахувань інформаційної системи	70
4.5	Обчислення накладних витрат	71
4.6	Складання кошторису витрат та визначення собівартості інформаційної системи	71
4.7	Розрахунок ціни інформаційної системи	72
4.8	Визначення економічної ефективності і терміну окупності капітальних вкладень	72
5	ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ.....	74
5.1	Методи боротьби зі шкідливими речовинами на виробництві	74
5.2	Види виробничого освітлення	75
	ВИСНОВКИ.....	77
	ПЕРЕЛІК ПОСИЛАНЬ	78
	ДОДАТКИ.....	80
	Додаток А. Лістинг файлу «LoginPage.tsx».....	80
	Додаток Б. Лістинг файлу «AdminDashboard.tsx»	83
	Додаток В. Лістинг файлу «POSTerminal.tsx»	85
	Додаток Г. Лістинг файлу «useCartStore.ts»	94
	Додаток Д. Лістинг файлу «auth.middleware.ts»	96
	Додаток Е. Лістинг файлу «pos.controller.ts»	97

АНОТАЦІЯ

Тема виконаної кваліфікаційної роботи: Розробка інформаційної системи складського обліку будівельних матеріалів «BudSklad»

Метою цієї кваліфікаційної роботи є розробка клієнт-серверного вебзастосунку для автоматизації процесів складського обліку, обслуговування покупців на касі за допомогою сканування штрихкодів, дистанційного формування замовлень менеджерами та комплексного адміністрування товарного каталогу.

У загальному підсумку пояснювальна записка складається з 5 розділів. У загальній частині описуються аналітичний огляд існуючих рішень у сфері автоматизації складської діяльності та описано технічне завдання.

У другому розділі представлено процес створення програмного продукту, опис та обґрунтування вибору методів проектування інтерфейсу, структури бази даних, написання клієнтської та серверної частин системи, а також тестування та налагодження програмних модулів.

В спеціальній частині описані інструкції з розгортання системи, інструкція з користування системою для різних типів користувачів. Описано шляхи популяризації вебплатформи.

Для оцінки комерційної привабливості розрахунок вартості розробки та загальної економічної ефективності приведено в економічній частині пояснювальної записки. Питання охорони праці та техніки безпеки розглянуто в п'ятому розділі.

Загальний обсяг пояснювальної записки становить 98 сторінок.

До складу кваліфікаційної роботи входить графічна частина, яка складається з структурної схеми інформаційної системи, діаграми варіантів використання, діаграми бази даних, а також техніко-економічних показників, що виконані на окремих аркушах формату А1.

					2026.КВР.122.423.09.00.00 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

ABSTRACT

Title of the Qualification Project: Development of an Information System for Warehouse Inventory Management of Construction Materials “BudSklad”

The goal of this qualification project is to develop a client-server web application for automating warehouse inventory management processes, serving customers at the checkout via barcode scanning, enabling managers to create orders remotely, and providing comprehensive administration of the product catalog.

Overall, the explanatory note consists of 5 chapters. The general section provides an analytical review of existing solutions in the field of warehouse automation and describes the technical specifications.

The second section presents the process of creating the software product, including a description and justification of the chosen methods for interface design, database structure, development of the client- and server-side components of the system, as well as testing and debugging of the software modules.

The special section contains instructions for deploying the system and user guides for different types of users. It also describes ways to promote the web platform.

To assess commercial viability, calculations of development costs and overall economic efficiency are provided in the economic section of the explanatory note. Occupational health and safety issues are addressed in Chapter 5.

The explanatory note consists of a total of 98 pages.

The thesis includes a graphical section comprising a structural diagram of the information system, a use case diagram, a database diagram, and technical and economic indicators, all presented on separate A1-size sheets.

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	<i>Арк.</i>
						<i>8</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

ВСТУП

Сучасний ринок роздрібної торгівлі будівельними матеріалами характеризується високим рівнем конкуренції. Великі мережеві гравці активно впроваджують дорогі та комплексні системи автоматизації, що дозволяє їм оптимізувати логістику, контролювати залишки та підвищувати якість обслуговування. Водночас малий та середній бізнес часто залишається поза цим процесом. Більшість невеликих магазинів досі покладаються на застарілі, паперові методи ведення обліку або прості табличні редактори.

Такий підхід неминуче призводить до низки критичних проблем. Серед них варто виділити неточність даних про наявність товару, що унеможливорює ефективне планування закупівель. Також виникають значні фінансові втрати через неумисні помилки персоналу або умисні крадіжки, які важко відстежити за паперовими записами. Зрештою, це знижує швидкість та якість обслуговування покупців, що в умовах жорсткої конкуренції є неприпустимим. Необхідність вирішення цієї проблеми є очевидною, адже від цього залежить не лише прибутковість, але й саме виживання тисяч малих підприємств.

Актуальність даної роботи полягає у гострій потребі ринку в доступному, простому та спеціалізованому програмному забезпеченні, яке б закривало ключові потреби невеликих будівельних магазинів без необхідності величезних капіталовкладень та тривалого навчання персоналу.

Метою кваліфікаційної роботи є розробка комплексної хмарної платформи для автоматизації складського обліку та процесів продажу в невеликих магазинах будівельних матеріалів. Робота охоплює повний цикл створення продукту: від аналізу вимог та проектування архітектури до реалізації функціоналу та розробки бізнес-моделі його подальшого розповсюдження.

Для досягнення поставленої мети було визначено наступні ключові завдання:

– провести аналіз сучасного стану автоматизації в секторі малого бізнесу та визначити основні потреби цільової аудиторії;

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

- спроектувати гнучку та масштабовану архітектуру програмного продукту на основі сучасного технологічного стеку, зокрема платформи Node.js;
- розробити ключові програмні модулі, що реалізують базовий функціонал: ведення каталогу товарів, облік залишків, реєстрація продажів, сканування штрих-кодів та формування простої звітності;
- запропонувати та обґрунтувати економічно доцільну модель поширення готового програмного рішення за принципом «програмне забезпечення як послуга» (SaaS).

Об'єктом дослідження є процеси операційної діяльності в роздрібній торгівлі будівельними матеріалами.

Предметом дослідження є розробка та впровадження хмарної системи для автоматизації цих процесів.

Галузь можливого використання розробленого програмного продукту охоплює підприємства малого та середнього бізнесу, що спеціалізуються на продажу будівельних матеріалів, інструментів, сантехніки та супутніх товарів. Запропоновані програмні рішення та рекомендації можуть бути застосовані для створення аналогічних спеціалізованих систем обліку для інших ніш роздрібної торгівлі. Практична цінність роботи полягає у створенні готового до впровадження інструменту, який допомагає підвищити ефективність бізнесу та його конкурентоспроможність.

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	<i>Арк.</i>
						10
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

1 ЗАГАЛЬНИЙ РОЗДІЛ

1.1 Аналітичний огляд існуючих рішень

Комерційна діяльність великого будівельного магазину завжди супроводжується величезним потоком найрізноманітніших товарів від мішків цементу до дрібних металевих кріплень. Управління таким об'ємним товарним асортиментом вимагає надійного спеціалізованого програмного забезпечення для точного комп'ютерного фіксування кожної торгової операції [1]. Власники подібних будівельних підприємств зазвичай обирають один із двох доступних шляхів вирішення цієї проблеми під час початкового запуску бізнесу. Перший варіант завжди передбачає швидку купівлю або щомісячну оренду готової універсальної платформи управління торгівлею. Другий шлях безпосередньо полягає у замовленні дорогої розробки абсолютно нового власного програмного забезпечення виключно під конкретні щоденні запити своєї компанії.

Готові рішення часто виглядають надзвичайно привабливо завдяки можливості швидкого старту без жодної необхідності довго чекати написання коду програмістами. Розглянемо значно детальніше популярні універсальні системи управління підприємством на зразок відомої програми BAS або схожих громіздких настільних комп'ютерних систем. Вони завжди пропонують покупцям величезний набір готових функцій для бухгалтерського обліку, нарахування заробітної плати персоналу та автоматичного формування податкових звітів. Інтерфейс подібних старих програм зазвичай сильно перевантажений сотнями різних дрібних кнопок та незрозумілих електронних таблиць як зображено на рисунку 1.1.

Така надмірна функціональність часто стає недоліком для персоналу будівельного магазину, адже працівникам доводиться витратити забагато часу на базові операції, що сповільнює обслуговування клієнтів. Крім того, універсальні платформи вимагають тривалого навчання працівників, регулярного платного оновлення та часто містять купу зайвих модулів, за які підприємство змушене

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

переплачувати, навіть не використовуючи їх. Усе це створює додаткове фінансове та операційне навантаження на бізнес, змушуючи власників шукати більш гнучкі та інтуїтивно зрозумілі альтернативи.

Сформувати Показати настройки

Період: 01.01.2023 - 31.12.2023

ТОВ ПРОМІНЬ

Оборотно-сальдова відомість за 2023 р. ТОВ ПРОМІНЬ

Виведені дані: БО (дані бухгалтерського обліку)

Рахунок	Найменування						
1	Необоротні активи						
10	Основні засоби						
13	Знос (амортизація) необоротних активів						
2	Запаси						
23	Виробництво						
28	Товари	75 000,00					
3	Кошти, розрахунки та інші активи	57 600,00		13 600,00			71 200,00
36	Розрахунки з покупцями та замовниками	57 600,00				9 600,00	67 200,00
37	Розрахунки з іншими дебіторами			4 000,00			4 000,00
6	Поточні зобов'язання		1 272 100,00	15 395,80		89 495,80	1 346 200,00
63	Розрахунки з постачальниками та підрядниками			1 295 400,00		5 400,00	1 300 800,00
64	Розрахунки за податками й платежами	177 050,00			900,00	14 972,51	162 977,49
65	Розрахунки за страхуванням		33 000,00			11 000,00	44 000,00
66	Розрахунки з оплати праці		120 750,00	14 495,80		57 000,00	163 254,20
68	Розрахунки за іншими операціями					1 123,29	1 123,29
7	Доходи і результати діяльності		48 000,00	1 600,00		9 600,00	56 000,00
70	Доходи від реалізації		48 000,00	1 600,00		9 600,00	56 000,00
9	Витрати діяльності	252 833,33		73 000,00		4 500,00	321 333,33
90	Собівартість реалізації	4 500,00					9 000,00
91	Загально виробничі витрати			4 500,00		4 500,00	
92	Адміністративні витрати	248 333,33		61 000,00			309 333,33
94	Інші витрати операційної діяльності						3 000,00
Разом		1 320 100,00	1 320 100,00	112 594,80	112 594,80	1 402 200,00	1 402 200,00

Відповідальний: _____ (посада) _____ (підпис) _____ (розшифровка підпису)

Рисунок 1.1 - Робочий екран універсальної системи обліку товарів (BAS)

Сильна перевантаженість екрану зайвими даними сильно ускладнює процес швидкого навчання нових касирів, які приходять працювати у будівельний магазин. Працівник каси повинен максимально швидко сканувати штрихкоди важких мішків із сумішами та миттєво видавати паперовий чек наступному покупцю у довгій черзі. Натомість старі універсальні облікові системи завжди вимагають відкриття кількох різних вікон та повільного заповнення безлічі необов'язкових полів для проведення однієї швидкої транзакції. Таке незручне програмне середовище створює постійні штучні затримки на касі та закономірно викликає сильне роздратування втомлених покупців.

Іншим популярним різновидом існуючих готових рішень є легкі хмарні системи автоматизації роботи каси на зразок Poster або схожих вебзастосунків. Приклад такої хмарної програми зображено на рисунку 1.2. Ці сучасні програми

мають зручний та інтуїтивно зрозумілий сенсорний інтерфейс для швидкого пробиття товарів на мобільному планшеті чи комп'ютері.

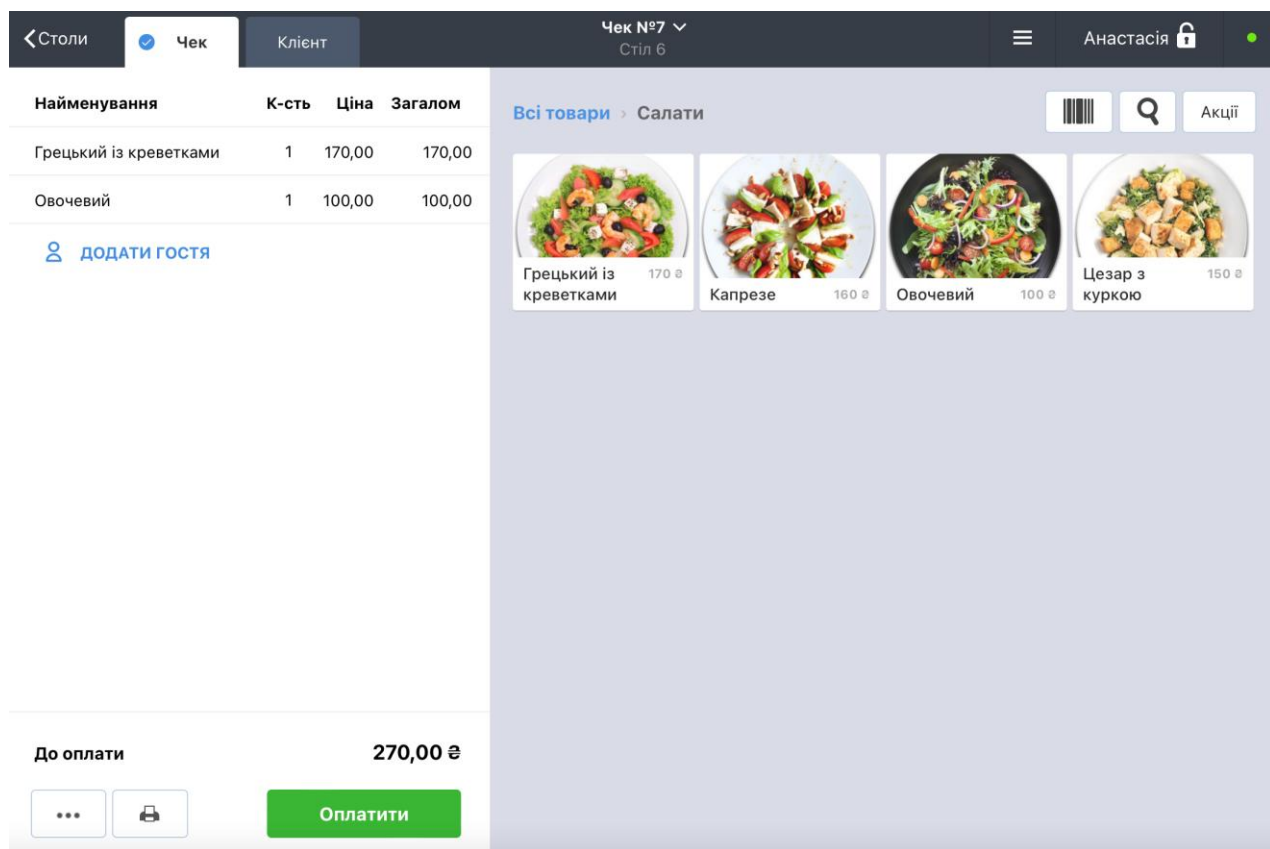


Рисунок 1.2 - Інтерфейс хмарної касової програми

Вони об'єктивно чудово підходять для невеликих кав'ярень або маленьких роздрібних кіосків із простим коротким переліком товарних позицій. Однак специфіка роботи великого будівельного магазину суворо вимагає не просто швидко продати один фізичний товар на касі. Велика крамниця зазвичай має окремого менеджера з продажу, який постійно приймає телефонні дзвінки від великих оптових замовників або бригадирів виконробів. Цей фахівець завжди формує об'ємні замовлення для подальшої доставки будівельних матеріалів вантажними автомобілями безпосередньо на будівельний майданчик клієнта.

Тут закономірно виникає серйозна технічна проблема конкурентного одночасного доступу до спільної інформаційної бази даних товарних залишків складу. Касир у торговому залі фізично бере останній мішок дорогого цементу

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

та починає швидко пробивати його через лазерний сканер на касі. Одночасно менеджер у своєму сусідньому кабінеті отримує телефонний дзвінок і швидко додає цей самий мішок цементу до великого віртуального замовлення на доставку. Більшість популярних легких хмарних касових програм абсолютно не вміють правильно обробляти такі одночасні паралельні мережеві запити від різних співробітників підприємства. Система може помилково продати один єдиний фізичний товар двом абсолютно різним клієнтам, що гарантовано призведе до конфлікту та фінансових втрат.

Готові системи також часто стягують постійну фіксовану щомісячну або щорічну плату за кожного нового підключеного працівника або робоче місце. Якщо будівельна компанія активно планує розширювати свій штат касирів та менеджерів, вартість володіння чужим програмним продуктом буде постійно швидко зростати. Власник бізнесу ніколи не отримує повного гарантованого контролю над своїми конфіденційними комерційними даними розташованими на чужих хмарних серверах.

Створення абсолютно власного спеціалізованого вебзастосунку Буд-склад дозволяє повністю назавжди уникнути всіх перелічених вище критичних недоліків готових чужих платформ. Цей правильний підхід гарантує розробку ідеально пристосованих графічних інтерфейсів для кожної окремої робочої посади у будівельному магазині. Касир отримає максимально простий темний екран із великими контрастними шрифтами спеціально розроблений для швидкої роботи зі сканером штрихкодів без використання мишки. Менеджер отримає зручну швидко багатовіконну систему для миттєвого пошуку товарів та формування красивих видаткових накладних для водіїв вантажівок. Головний адміністратор отримає надійно захищений доступ до детальної фінансової аналітики та зручних інструментів масового управління цінами на всі товари.

Власна продумана архітектура бази даних дозволить програмістам закласти надійний програмний механізм транзакцій для оставі вирішення проблеми паралельного продажу товарів. Програмний код буде миттєво

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		14

блокувати товарний залишок у базі саме в той момент, коли хтось із працівників додасть його до свого кошика. Таке надійне технічне рішення назавжди усуне неприємні ситуації подвійного помилкового продажу однакових фізичних позицій різним покупцям. Розроблена система належатиме виключно власнику магазину без жодних прихованих непередбачуваних щомісячних платежів за використання нових функцій або додавання нових працівників. Хоча процес тривалого проектування та написання коду вимагатиме певного часу, ці зусилля швидко окупляться завдяки високій фінансовій надійності готової програми.

Багато старих будівельних баз досі намагаються вести облік матеріалів за допомогою звичайних електронних таблиць Excel або навіть старих паперових журналів. Такий застарілий підхід абсолютно не витримує жодної раціональної критики при величезному щоденному потоці десятків покупців та сотень відвантажених позицій. Електронні таблиці зовсім не вміють автоматично друкувати фіскальні касові чеки або формувати зручні маршрутні листи для водіїв вантажівок. Працівники щодня витрачають багато цінного робочого часу на повільне ручне переписування комерційних даних з одного папірця на інший. Власник бізнесу взагалі не бачить реальної картини своїх щоденних доходів та товарних залишків до моменту проведення тривалої виснажливої ручної інвентаризації складу. Перехід від повільного паперового хаосу до швидкого надійного централізованого вебзастосунку є єдиним правильним шляхом для подальшого фінансового розвитку торгового підприємства.

1.2 Технічне завдання

1.2.1 Найменування та область застосування

Темою даного кваліфікаційного проекту визначено «Розробка інформаційної системи складського обліку будівельних матеріалів «BudSklad»». Створювана комп'ютерна програма формально відноситься до великої категорії систем планування ресурсів підприємства з потужним інтегрованим модулем

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

швидкого обслуговування локальних точок роздрібного продажу товарів. Головною визначеною областю практичного комерційного застосування нового програмного продукту є повна автоматизація абсолютно всіх внутрішніх щоденних бізнес-процесів компаній, які професійно торгують різноманітними будівельними матеріалами.

Система цілеспрямовано розрахована на постійне використання у специфічному робочому середовищі, де фізичний швидкий роздрібний продаж через звичайну касу постійно поєднується з дистанційним телефонним обслуговуванням великих оптових клієнтів. Робочими терміналами системи стануть звичайні персональні офісні комп'ютери, переносні мобільні планшети або компактні ноутбуки розташовані безпосередньо у торговому залі, офісі телефонних менеджерів та головному кабінеті керівника. Розроблений застосунок орієнтується на максимальне задоволення щоденних потреб трьох конкретних груп працівників будівельного магазину.

Першу групу складають рядові касири зали, чия головна задача полягає у максимально швидкому відпуску будівельних товарів покупцям із живої черги. Друга група включає швидких менеджерів з продажу, які приймають телефонні дзвінки, резервують важкі матеріали на складі та організують їх логістичну доставку. Третя найвища категорія користувачів складається з керуючих адміністраторів або власників бізнесу, які суворо контролюють фінансові потоки та керують великою базою товарів.

1.2.2 Призначення розробки

Головною метою створення комерційного вебзастосунку Буд-склад є повне об'єднання розрізнених процесів роздрібного та оптового продажу товарів в єдиному надійному цифровому середовищі. Розробник планує повністю назавжди усунути паперову бюрократичну тяганину та мінімізувати шкідливий вплив людського фактора на правильність списання товарних залишків зі складу. Написана програма повинна обов'язково стати центральним універсальним

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		16

інструментом для швидкого безперебійного обслуговування будь-якого клієнта магазину незалежно від способу його звернення до менеджерів.

Призначення розробки включає в себе надійну реалізацію кількох надзвичайно важливих практичних цілей для фінансового успіху бізнесу. Першою метою є значне прискорення щоденної роботи касира у торговому залі за рахунок створення зручного темного інтерфейсу спеціально адаптованого під лазерний сканер штрихкодів. Працівник каси повинен мати змогу швидко додавати товари у віртуальний кошик за лічені частки секунди без жодних зайвих кліків комп'ютерною мишкою. Другою вкрай важливою ціллю є швидке надання менеджерам потужного інструменту для формування складних замовлень безпосередньо під час телефонної розмови з оптовим клієнтом. Менеджер отримає чудову можливість миттєво бачити точну кількість наявного цементу або гіпсокартону на складі без найменшої необхідності йти туди фізично ногами.

Також електронної системи є суворий безперебійний фінансовий контроль усіх грошових операцій підприємства через спеціальний механізм касових робочих змін. Програма автоматично фіксуватиме точний час початку роботи кожного касира та початкову суму готівки у його металевій грошовій скриньці. Ввечері система повністю автоматично підрахує загальний денний виторг та гарантовано допоможе виявити будь-які неприємні фінансові нестачі під час закриття зміни. Четверта головна мета полягає у забезпеченні вищого керівництва зручними графічними звітами про щоденні продажі та постійний рух товарів для прийняття зважених управлінських рішень. Створений застосунок повинен надійно працювати без збоїв навіть за умови регулярного одночасного доступу багатьох працівників до однієї бази даних.

1.2.3 Вимоги до функціоналу інформаційної системи

Оскільки проєкт прямо передбачає створення складного інструментарію для різних посад, вимоги до програмного функціоналу чітко розділені на три

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	Арк.
						17
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

окремі логічні блоки. Кожен виділений блок відповідає за специфічне робоче комп'ютерне середовище конкретної групи користувачів вебзастосунку.

Перший блок докладно описує вимоги до робочого місця касира великої торгової зали. Графічний інтерфейс цієї частини програми має бути максимально простим, контрастним та зручним для довгої роботи при поганому освітленні магазину.

Система повинна надійно підтримувати миттєвий пошук потрібного товару у базі даних після швидкого зчитування його етикетки лазерним сканером штрихкодів. Касир обов'язково отримає можливість змінювати кількість однакових товарних позицій у відкритому чеку за допомогою швидких гарячих клавіш на клавіатурі. Програма має безперебійно підтримувати функцію відкладеного чека для тих ситуацій, коли покупець забув свій гаманець або пішов шукати додатковий товар у зал. Процес оплати повинен постійно пропонувати швидкий зручний вибір між внесенням паперової готівки або використанням банківського платіжного терміналу клієнтом. Після успішного комп'ютерного підтвердження транзакції система має повністю автоматично відправити команду на підключений принтер для фонового друку вузького паперового чека.

Другий великий блок стосується функціоналу для швидких менеджерів відділу продажу та логістичної доставки. Це комп'ютерне робоче місце повинно мати вигляд повноцінної складної системи управління взаємовідносинами з оптовими та роздрібними клієнтами бази. Менеджер обов'язково отримає зручне широке пошукове поле для швидкого знаходження специфічних будівельних матеріалів за назвою, заводським артикулом або категорією. Форма швидкого створення нового замовлення обов'язково міститиме текстові поля для введення імені замовника, його контактного мобільного телефону та точної адреси майбутньої доставки матеріалів. Програма повинна безпомилково підтримувати складний логічний життєвий цикл кожного замовлення з можливістю швидкої зміни його поточних статусів. Менеджер вільно зможе переводити створений

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
						18
Зм.	Арк.	№ докум.	Підпис	Дата		

документ зі стану абсолютно нового замовлення у стадію фізичного збирання на складі, а потім позначати його як успішно відправлений клієнту. Система обов'язково матиме зручну функцію швидкої генерації повноцінної видаткової накладної формату А4 у вигляді файлу PDF для зручного роздрукування на офісному принтері.

Третій важливий блок вимог суворо регламентує функціонал закритої панелі головного адміністратора магазину. Керівник отримає закритий доступ до захищеного розділу управління великою базою товарів для швидкого створення нових позицій або редагування старих цін постачальників. Розроблена система дозволить адміністратору генерувати абсолютно унікальні штрихкоди для нових товарів та роздруковувати паперові цінники для полиць торгового залу. Панель безпечного керування персоналом надасть зручну можливість створювати нові облікові записи для найнятих працівників та швидко скидати втрачені паролі доступу. Адміністратор матиме постійний цілодобовий доступ до перегляду історії всіх закритих касових змін для детального щоденного контролю за готівкою у касі магазину. Графічні дашборди повинні повністю автоматично малювати зрозумілі діаграми щоденного виторгу підприємства та показувати точну статистику найбільш популярних серед покупців будівельних матеріалів.

Діаграма варіантів використання інформаційної системи зображено на плакаті 2026.КВР.122.423.09.00.00 ДВ.

Окремою критично важливою технічною вимогою є надійна безперебійна програмна реалізація механізму транзакцій бази даних. Якщо касир та телефонний менеджер випадково одночасно спробують продати останній мішок шпаклівки, система повинна коректно обробити цей конфлікт. Серверна частина створеного застосунку миттєво віддасть залишок замовленого товару тому працівнику, який першим натисне кнопку проведення електронного документа. Інший працівник магазину одразу побачить на своєму робочому екрані текстове повідомлення про повну відсутність достатньої кількості потрібних матеріалів на складі підприємства.

					<i>2026.КВР.122.423.09.00.00 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>19</i>

1.2.4 Вимоги до програмної документації

Після повного успішного завершення тривалого процесу написання коду розробник беззаперечно зобов'язаний підготувати пакет супровідної технічної документації для майбутніх користувачів системи. Якісно і зрозуміло написані тексти інструкцій допоможуть керівництву магазину самостійно підтримувати працездатність застосунку без постійного звернення до програмістів по допомогу.

Першим важливим електронним документом стане детальна покрокова інструкція з розміщення готового вебзастосунку на локальному або орендованому сервері в глобальній мережі інтернет. Текст детально опише процес безпечного копіювання файлів написаної програми на хостинг та налаштування правильного з'єднання з базою даних паролями. Документ міститиме точний перелік необхідних консольних команд для терміналу, які запускають серверну частину програми у постійному фоновому робочому режимі. Розробник також доступно пояснить процедуру прив'язки красивого короткого доменного імені до числової IP адреси вибраного вебсервера для зручного доступу співробітників.

Другий великий документ стосуватиметься строгих правил регулярного технічного обслуговування системи та безпечного наповнення електронного каталогу товарів фотографіями. Текстова інструкція пояснить керівним адміністраторам порядок створення правильних регулярних резервних копій бази даних для надійного збереження комерційної інформації від випадкової втрати. Написаний текст наочно покаже механізм масового швидкого завантаження списків нових товарів із файлів електронних таблиць до бази даних вебзастосунку. Створена документація також опише правильну процедуру безпечного оновлення програмного коду сайту у випадку майбутнього випуску нових покращених версій системи розробником.

Також обов'язковим електронним документом є просте та максимально зрозуміле коротке керівництво користувача для касирів та менеджерів з

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

продажу. Цей текст свідомо писатиметься максимально простою звичайною мовою без найменшого використання складної незрозумілої технічної термінології програмістів. Написане керівництво міститиме багато реальних знімків екрану з детальним поясненням прямого призначення кожної робочої кнопки у комп'ютерній програмі. Створений електронний документ дозволить власнику бізнесу швидко навчати нових найнятих працівників без необхідності витратити свій власний дорогоцінний час на пояснення азів.

1.2.5 Техніко-економічні показники

Реальний економічний успіх будь-якого комерційного програмного студентського проєкту безпосередньо залежить від жорсткого регулярного контролю за витраченим часом та грошовими ресурсами. Усі проведені фінансові розрахунки для швидкого створення вебзастосунку Буд-склад цілком базуються на офіційних методичних вказівках до виконання економічного розділу дипломної роботи.

Планується повністю завершити активну фазу написання програмного коду та тестування системи за визначений період, який зовсім не перевищує 180 людино годин. Цей виділений час повністю включає в себе:

- Планування та моделювання – 50 годин;
- детальну розробку серверної частини API – 50 годин;
- верстку всіх графічних інтерфейсів програми – 50 годин;
- Тестування – 20 годин;
- Підготовку документації - 5 годин;
- Впровадження - 5 годин;

Кінцева сумарна загальна вартість розробки та успішного впровадження нового вебзастосунку на діючому підприємстві жорстко обмежується сумою у 85703 гривень. Вказана сума повністю покриває віртуальну оплату праці розробника, річну оренду надійного хостингу та швидку купівлю зручного

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

короткого доменного імені для додатку. Детальні розрахунки представлені в четвертому розділі записки (Економічна частина). А техніко-економічні показники подано на плакаті 2026.КВР.122.423.09.00.00 ТБ.

Розрахунковий планований короткий термін повної економічної окупності впровадженої комп'ютерної системи управління торгівлею має неодмінно скласти значно менше одного календарного року. Швидке гарантоване повернення вкладених інвестицій стане повністю можливим завдяки повному усуненню типових помилок персоналу під час відвантаження товарів та припиненню крадіжок готівки з відкритої каси. Впровадження точного комп'ютерного обліку залишків дозволить вищому керівництву назавжди припинити закупівлю абсолютно зайвих будівельних матеріалів та швидко звільнити заморожені оборотні кошти підприємства.

1.2.6 Стадії та етапи розробки

Створення подібного складного комп'ютерного інструменту вимагає суворого постійного дотримання правильної затвердженої послідовності етапів написання програмування. Розробка нового вебзастосунку Буд-склад послідовно проходить через кілька визначених стадій від початкового теоретичного планування до фізичного запуску коду на віддалених серверах.

Першою довгою стадією є глибокий детальний аналіз предметної області та детальне технічне планування майбутньої програмної архітектури. Розробник ретельно збирає всі технічні вимоги від гіпотетичних замовників та складає вичерпний повний перелік необхідних корисних функцій для касирів та адміністраторів. На цьому початковому етапі також відбувається остаточний вибір технологічного стеку, де перевага свідомо надається середовищу Node.js для сервера та бібліотеці React для клієнта. Програміст швидко складає детальну графічну схему всіх таблиць бази даних та визначає правильні логічні зв'язки між товарами, чеками та замовленнями клієнтів.

Друга важлива стадія повністю присвячена візуальному графічному

					<i>2026.КВР.122.423.09.00.00 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		22

прототипуванню та дизайну інтерфейсів робочих сторінок. Автор самостійно малює приблизні макети майбутніх робочих екранів програми на папері або у спеціальних потужних графічних векторних редакторах. Головна пильна увага приділяється зручності розташування елементів управління та розміру шрифтів для комфортної довгої роботи людей з поганим зором. Дизайнер ретельно обирає приємну темну колірну палітру, яка зовсім не викликати швидкої втоми очей касира під час довгої десятигодинної робочої зміни.

Третя складна стадія безпосередньо включає саму розробку серверної частини системи управління товарами. Програміст швидко налаштовує середовище розробки та створює моделі даних бази за допомогою зручного сучасного інструменту Prisma ORM. На цьому довгому кроці пишеться логіка безпечної перевірки авторизації всіх користувачів за допомогою зашифрованих електронних токенів доступу. Розробник паралельно створює захищені цифрові маршрути API для швидкої обробки всіх мережевих запитів від клієнтських пристроїв до спільної віддаленої бази даних. Найважливішим технічним завданням є написання складного механізму транзакцій для безпечного списування товарних залишків під час масового продажу матеріалів вдень.

Четверта велика стадія сильно фокусується на швидкому написанні клієнтської частини програми за допомогою сучасної бібліотеки React. Програміст поступово перетворює намальовані графічні макети на живі красиві інтерактивні вебсторінки з клікабельними кнопками та зручними формами введення даних. Студент успішно реалізує складну візуальну логіку віртуального кошика покупця та вікна формування великого оптового замовлення менеджера. Готовий створений інтерфейс повністю підключається до серверного API для отримання реальних списків товарів та відправлення готових фінансових чеків на постійне збереження у базу.

П'ята стадія цілком присвячена ретельному функціональному тестуванню та покращенню швидкості роботи вже написаного програмного коду. Уважно перевіряється правильність обчислення математичних сум у чеках та коректність

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

роботи механізму паралельного доступу до бази даних товарів. Виявляються та оперативно швидко виправляються всі знайдені дрібні логічні помилки або збої у відображенні кнопок на екранах різних розмірів. Також ретельно проводиться робота зі зменшення об'єму файлів програми для їх швидкого завантаження на старих повільних офісних комп'ютерах.

Шоста остання стадія є фінальним довгоочікуваним етапом релізу та фізичного розгортання всього створеного проєкту в мережі. Готова перевірена програма повністю переноситься з локального комп'ютера розробника на надійний орендований сервер у глобальній мережі інтернет. Проводиться фінальну швидку перевірку працездатності сайту в реальних робочих умовах та створює перші паролі для облікових записів адміністраторів магазину. Ця остання стадія також передбачає детальне планування подальшої підтримки застосунку та швидкого виправлення будь-яких прихованих помилок, які можуть виникнути у перші тижні реальної експлуатації.

1.2.7 Порядок контролю та прийому

Процедура ретельного тестування та формального прийому готової роботи є надзвичайно важливим вирішальним етапом життєвого циклу розробки будь-якого програмного забезпечення. сувора перевірка всіх компонентів коду повністю гарантує відсутність критичних збоїв у системі під час її постійного використання в реальних комерційних умовах магазину. Порядок проведення тестування великого інтернет-проєкту Буд-склад складається з кількох чітко визначених послідовних кроків.

Першим початковим кроком є глибоке та ретельне функціональне тестування абсолютно всіх запланованих робочих ролей майбутніх користувачів системи. Імітуюся щоденні звичні дії рядового касира, постійно скануючи штрихкоди товарів, тимчасово відкладаючи незавершені чеки та успішно проводячи оплату готівкою або віртуальними безготівковими грошима. Після цього обов'язково перевіряється щоденна робота менеджера шляхом постійного

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	Арк.
						24
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

створення нових складних замовлень, швидкої зміни їхніх логічних статусів та автоматичної генерації супровідних документів у зручному форматі PDF.

Особлива пильна увага приділяється повноцінному тестуванню закритого модуля адміністратора, включаючи складний процес створення абсолютно нових товарів та правильного графічного формування фінансових звітних графіків. Найретельніше перевіряється поведінка спільної реляційної бази даних при навмисній спробі одночасного паралельного продажу останнього мішка будівельної сухої суміші абсолютно різними користувачами з різних комп'ютерів.

Другий довгий крок стосується візуальної перевірки сумісності всіх готових інтерфейсів на різноманітних мобільних та стаціонарних пристроях. Автор постійно відкриває робочі сторінки програми у популярних веб-браузерах на зразок Chrome або Firefox для пошуку можливих неприємних розбіжностей у відображенні. Створений інтерфейс ретельно тестується на великих офісних моніторах, екранах старих ноутбуків та навіть на дисплеях маленьких мобільних телефонів. Програміст уважно перевіряє, щоб жодні життєво важливі робочі кнопки не виходили за межі екрану і завжди залишалися повністю доступними для натискання.

Третій важливий крок присвячений детальному вимірюванню загальної продуктивності та швидкодії роботи великого вебзастосунку при навантаженнях. Розробник уважно аналізує загальний час повного завантаження робочого місця касира та швидкість пошуку товару у великій базі даних, яка містить тисячі різних позицій. Усі сторінки повинні реагувати на звичні дії користувача практично миттєво, щоб зовсім не створювати довгих черг роздратованих покупців біля працюючої каси. Для успішного досягнення цієї мети автор сильно мінімізує розмір переданих через мережу картинок та максимально пришвидшує складні запити до сервера.

Четвертим етапом є базова надійна перевірка стійкості написаного застосунку до типових загроз безпеці конфіденційної комерційної інформації.

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		25

Розроблена система авторизації суворо тестується на здатність швидко блокувати будь-які спроби входу з неправильними паролями співробітників. Серверна частина бази перевіряється на відсутність відомих вразливостей, які могли б дозволити стороннім зловмисникам стягнути базу товарів або повністю видалити історію продажів компанії.

Останній вирішальний крок передбачає формальну процедуру офіційної здачі виконаної роботи представникам замовника або приймальній комісії навчального закладу. Розробник дипломного проєкту повинен представити повністю робочий перевірений вебзастосунок, який беззаперечно відповідає абсолютно всім пунктам узгодженого технічного завдання. також необхідно передати повний архів із вихідним кодом створеної програми на звичайному оптичному або флеш-носії інформації членам комісії.

Сам офіційний прийом завершеної роботи відбувається під час спеціального засідання комісії у строго визначеній послідовності дій студента. Спочатку відбувається виступ з короткою доповіддю про успішно виконану роботу та детально пояснює обрані архітектурні програмні рішення. Після цього обов'язково відбувається наочна публічна демонстрація роботи програми на великому екрані з поясненням функцій касира та менеджера присутнім слухачам. Члени комісії мають повне право вимагати негайного проведення додаткових контрольних випробувань прямо під час процесу захисту. Наприклад, вони можуть несподівано попросити спробувати продати товар, якого вже фізично зовсім немає на залишках віртуального складу магазину. Процедура закономірно успішно завершується довгою сесією запитань від викладачів та детальними відповідями автора на абсолютно всі технічні зауваження щодо роботи системи.

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		26

2 РОЗРОБКА ТЕХНІЧНОГО ТА РОБОЧОГО ПРОЄКТУ

2.1 Розробка структури інформаційної системи і web-сторінок

Процес створення будь-якого сучасного програмного продукту для сфери торгівлі та логістики починається з ретельного планування його архітектури, структури екранів та логіки взаємодії користувача з інтерфейсом. Програма під назвою Буд-склад виконує дві функції одночасно, вона допомагає керувати залишками матеріалів на складі та оперативно обслуговувати покупців на точці роздрібного продажу. Головна мета проектування інтерфейсу полягає у створенні простого інтуїтивного робочого простору для працівників різної кваліфікації [2]. Увесь інтерфейс було розділено на три окремі частини, кожна з яких орієнтується на конкретну посаду співробітника компанії. Програма надає індивідуальні набори інструментів для адміністраторів, менеджерів торгового залу та касирів.

Проектування інтерфейсу розпочалося з аналізу вимог та створення детальної карти переходів між майбутніми екранами. Мапа сайту допомагає зрозуміти шлях користувача під час виконання типових операцій. Робочий процес над дизайном було розділено на кілька логічних етапів. Спочатку було розроблено концептуальну схему розташування елементів на головних сторінках. Далі було створено чорнові графічні ескізи для внутрішніх вікон та форм вводу. Після обговорення прототипів з працівниками торгової мережі було внесено корективи у розміри шрифтів та кнопок. На фінальному етапі було затверджено візуальні макети всіх сторінок системи. Кольорова гама інтерфейсу використовує м'які пастельні відтінки, щоб очі персоналу не втомлювалися протягом зміни тривалістю 8-12 годин. Яскраві кольори виділяють виключно кнопки швидкої дії.

Загальне візуальне розташування елементів та ієрархію екранів для кожної ролі показує структурна схема інформаційної системи на плакаті 2026.КВР.122.423.09.00.00 СС.

					2026.КВР.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		27

Кожен користувач починає взаємодію з програмою з екрану авторизації. Ця сторінка містить поля для електронної пошти та пароля. Після перевірки даних на сервері програма визначає роль працівника та автоматично спрямовує його у потрібну робочу зону.

Сторінка авторизації має простий лаконічний вигляд. Вона містить логотип системи, два поля вводу з підказками та кнопку входу. У випадку помилки під час введення пароля програма підсвічує відповідне поле червоним кольором та виводить коротке текстове повідомлення. Це допомагає швидко зрозуміти причину проблеми без додаткових дій.

Касир працює з інтерфейсом, який було оптимізовано під швидке сканування штрихкодів та миттєвий продаж. Його робоче місце містить наступні екрани:

- екран відкриття та закриття робочої зміни касира;
- головний екран продажу товарів та наповнення кошика;
- вікно вибору способу оплати та друку фіскального чека.

На екрані відкриття зміни касир бачить власне ім'я, дату та поле для введення суми початкового залишку готівки в касі. Після натискання кнопки відкриття програма активує основний робочий простір продажу. Цей простір було розділено на три функціональні зони. Зліва знаходиться велике поле для автоматичного зчитування штрихкодів. Також тут було розміщено рядок текстового пошуку для ручного вибору товарів за назвою чи артикулом. У центрі екрану відображається таблиця поточного чека з назвами товарів, цінами, кількістю та загальною сумою. Справа знаходиться панель швидких дій, яка дозволяє застосувати знижку або очистити кошик. Окреме вікно розрахунку містить великі кнопки вибору оплати готівкою чи карткою, поле для введення отриманої від клієнта суми та автоматичний розрахунок решти.

Менеджер торгового залу використовує інтерфейс для роботи з клієнтами та оптовими замовленнями. Для цієї посади було розроблено такі сторінки:

- сторінка переліку активних та архівних замовлень покупців;

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		28

- форма оформлення нового замовлення на доставку;
- картка детальної інформації про статус виконання замовлення;
- довідник постійних клієнтів з історією їхніх покупок.

Сторінка переліку замовлень містить фільтри за датою, статусом та менеджером. Менеджер може швидко знайти потрібний документ за номером або телефоном клієнта. Форма створення нового замовлення містить блок введення контактних даних покупця, адресу доставки та вибір способу отримання товарів. Нижче розташована область додавання будматеріалів, яка підключена до загального каталогу складу. Програма автоматично перевіряє залишки продукту під час спроби додати його у замовлення. Якщо матеріалу недостатньо, програма блокує кнопку додавання та показує доступну на складі кількість. Картка статусу дозволяє менеджеру відстежувати рух замовлення від моменту збирання на складі до доставки покупцеві.

Адміністратор отримує повний контроль над усіма процесами компанії та налаштуваннями системи. До його інтерфейсу входять такі сторінки:

- інформаційна панель з аналітикою та графіками продажів;
- сторінка управління каталогом будівельних матеріалів;
- форма додавання нових позицій та редагування цін;
- список облікових записів працівників з налаштуванням прав.

Інформаційна панель адміністратора містить графічні діаграми, які відображають дохід за вибраний місяць, топ популярних товарів та продуктивність окремих касирів. Сторінка каталогу матеріалів виконана у вигляді таблиці з розширеними можливостями фільтрації та пошуку. Адміністратор може відсортувати товари за категоріями, брендами або кількістю на складі. Картка створення товару містить поля для назви, опису, ціни, артикулу та штрихкоду. Також вона має кнопку завантаження зображення через підключений хмарний сервіс. Сторінка управління користувачами дозволяє адміністратору створювати профілі для нових менеджерів чи касирів, змінювати їхні паролі або блокувати доступ у разі звільнення працівника.

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		29

2.2 Створення та верстка сторінок інформаційної системи

Для розробки користувацького інтерфейсу програми Буд-склад було використано компонентний підхід на базі бібліотеки React [3] та середовища Vite. Візуальний стиль елементів було реалізовано за допомогою Tailwind CSS [4], що спрощує верстку завдяки використанню готових класів безпосередньо в коді компонентів. Навігацію між екранами було налаштовано через пакет react-router-dom, а глобальний стан авторизації та кошика було збережено у сховищі Zustand. Всі іконки інтерфейсу було імпортовано з бібліотеки Lucide React.

Процес безпосереднього написання коду та створення файлів web-сторінок було реалізовано у текстовому редакторі Visual Studio Code, який обрано через наявність вбудованого терміналу та підтримку автоматичного доповнення коду для мови TypeScript і класів Tailwind CSS. Для стилізації сторінок було використано прийоми адаптивної верстки на основі сіток CSS Grid та гнучких блоків Flexbox. Це дозволило правильно розташувати всі елементи інтерфейсу та налаштувати їхню поведінку при зміні розмірів вікна браузера. Усі кольорові токени, розміри шрифтів та відступи налаштовано через конфігураційний файл tailwind.config.js, що спрощує зміну дизайну та дотримання єдиного візуального стилю по всьому проекту.

Сторінка входу виконує авторизацію працівників. Для реалізації форми було використано хуки useState, які фіксують введені email та пароль. При натисканні кнопки входу програма відправляє POST запит на адресу auth/login. Кнопка входу містить іконку LogIn та анімований індикатор завантаження у вигляді спінера, який з'являється під час очікування відповіді від сервера. Для візуального оформлення сторінки було використано логотип з іконкою Warehouse, а також закруглені кути для полів вводу та кнопки.

Для сторінки входу було розроблено спеціальне оформлення з вертикальним вирівнюванням елементів по центру за допомогою класу space-y-8. Брендований блок з логотипом виділено фірмовим індиго кольором bg-indigo-600 з додатковим тіншовим ефектом shadow-indigo-200. Поля вводу отримали

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

світло-сірий фон `bg-slate-50` та тонкі рамки `border-slate-200`, які динамічно змінюють колір рамки на синій при отриманні фокусу. Для кнопки входу було налаштовано інтерактивний ефект зменшення при натисканні `active:scale-[0.98]`, що створює тактильний відгук інтерфейсу. Зовнішній вигляд форми авторизації показано на рисунку 2.1.

Вітаємо в BudSkлад
СИСТЕМА УПРАВЛІННЯ ТОРГІВЛЕЮ

EMAIL АДРЕСА
admin@budsklad.com

ПАРОЛЬ
.....

→] Увійти в кабінет

Якщо ви забули пароль, зверніться до адміністратора

Рисунок 2.1 - Сторінка входу в систему

Структуру верстки форми авторизації користувача подано на лістингу 2.1.

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

```

<form onSubmit={handleSubmit} className="space-y-6">
  <div>
    <label className="block text-sm font-semibold text-slate-700">Електронна пошта</label>
    <input type="email" required className="mt-1 block w-full px-3 py-2 bg-slate-50 border border-slate-200 rounded-md focus:ring-2 focus:ring-indigo-500" />
  </div>
  <div>
    <label className="block text-sm font-semibold text-slate-700">Пароль</label>
    <input type="password" required className="mt-1 block w-full px-3 py-2 bg-slate-50 border border-slate-200 rounded-md focus:ring-2 focus:ring-indigo-500" />
  </div>
  <button type="submit" className="w-full flex justify-center py-2 px-4 rounded-md text-sm font-medium text-white bg-indigo-600 active:scale-[0.98] transition-transform">
    Увійти
  </button>
</form>

```

Лістинг 2.1 - Верстка форми авторизації у LoginPage.tsx

У цьому лістингу продемонстровано структуру розмітки форми авторизації користувача. Опис елементів інтерфейсу виконано з використанням стандартних тегів форми, текстових полів введення та кнопки надсилання даних. За допомогою утиліт стилізації визначено параметри відступів, кольорової схеми елементів, заокруглення кутів, а також налаштовано візуальні ефекти при фокусуванні на полях введення та натисканні кнопки. Повний код компонента LoginPage.tsx подано в додатку А пояснювальної записки.

Адмін панель було розділено на кілька автономних частин для полегшення адміністрування. Перший екран має назву Статистика системи. Він

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		32

відображає основні показники роботи будівельного магазину. Тут було розміщено чотири картки з кольоровими іконками, які показують загальну кількість товарів у базі, активних користувачів, обсяг продажів за день та замовлення в роботі. Під картками знаходиться графічна область для виведення динаміки продажів у реальному часі. Зовнішній вигляд панелі статистики зображено на рисунку 2.2.

Особливістю стилізації статистика системи є використання адаптивної сітки grid-cols-1 з автоматичним розширенням до чотирьох колонок на великих екранах. Картки показників отримали білий фон bg-white та легку рамку border-slate-100 для відділення від сірого фону сторінки. Кожна іконка розміщена всередині окремого квадратного блоку із заокругленими кутами rounded-xl, який має власний насичений колір для швидкої візуальної ідентифікації показника: синій bg-blue-500 для товарів, фіолетовий bg-indigo-500 для персоналу, зелений bg-emerald-500 для грошових продажів та помаранчевий bg-amber-500 для замовлень.

Область графіка оформлена у вигляді рамки з пунктирним стилем border-dashed border-slate-200.

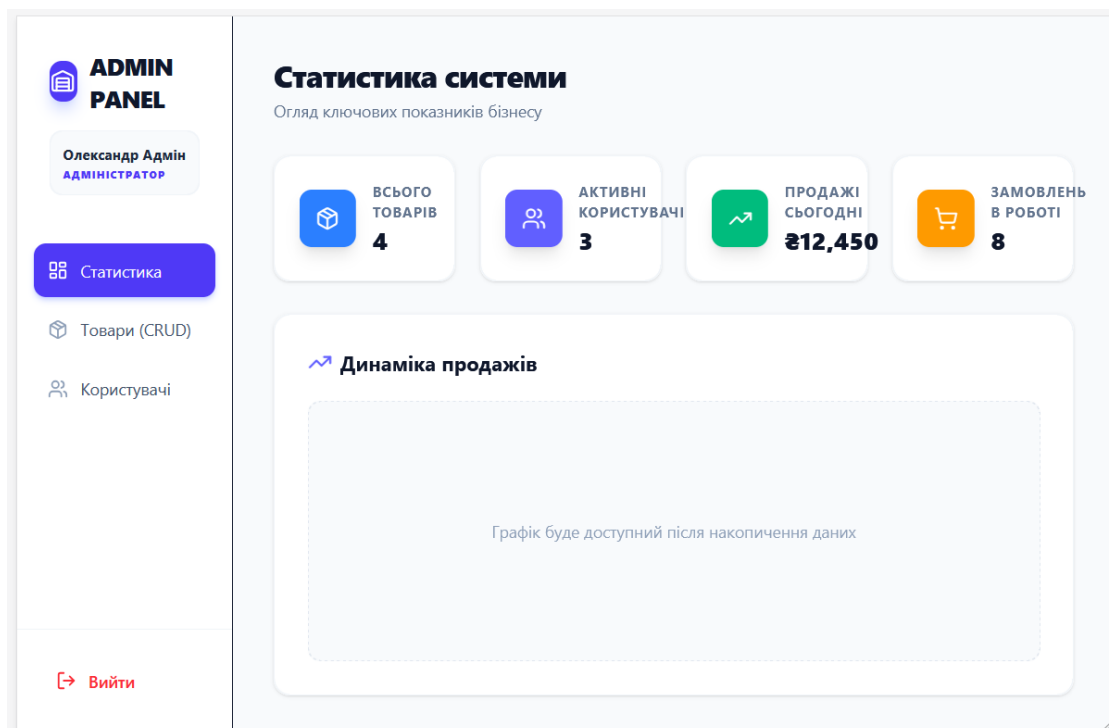


Рисунок 2.2 - Розділ Статистика системи в адмін панелі

Структуру верстки блоку статистичних показників в адмін-панелі подано на лістингу 2.2.

```
<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-6">
  {stats.map((stat, i) => (
    <div key={i} className="bg-white p-6 rounded-2xl border border-slate-100 shadow-sm flex items-center space-x-4">
      <div className={` ${stat.color} p-4 rounded-xl text-white shadow-lg`} >
        <stat.icon size={24} />
      </div>
      <div>
        <p className="text-sm font-bold text-slate-500 uppercase tracking-wider">{stat.label}</p>
        <p className="text-2xl font-black text-slate-900 mt-0.5">{stat.value}</p>
      </div>
    </div>
  ))}
</div>
```

Лістинг 2.2 - Розмітка карток статистики у AdminDashboard.tsx

У цьому лістингу продемонстровано структуру розмітки блоку карток для відображення статистичних показників. Шаблон побудовано з використанням адаптивної сітки, яка автоматично змінює кількість колонок залежно від розміру екрана. Кожна окрема картка містить внутрішній контейнер для іконки з відповідним кольоровим фоном та текстовий блок для виведення опису і значення показника з налаштованими параметрами шрифтів. Повний код компонента AdminDashboard.tsx подано в додатку Б пояснювальної записки.

Наступний розділ має назву Управління товарами. Він містить таблицю товарів з фотографіями, артикулами, назвами, цінами та залишками на складі. У

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

верхній частині екрану знаходиться поле пошуку з іконкою Search та кнопка Додати товар з іконкою Plus. Кожен рядок таблиці містить кнопки дій для редагування та видалення. Список товарів відображається у вигляді таблиці, яку представлено на рисунку 2.3.

У розділі управління товарами стилізація таблиці базується на використанні горизонтальної прокрутки overflow-x-auto для мобільних пристроїв. Для рядків таблиці налаштували плавний перехід фоновому кольору при наведенні курсору hover:bg-slate-50/80 transition-colors. Текстові артикули товарів отримали моноширинний шрифт font-mono та оформлені у вигляді сірих плашок px-2 py-1 bg-slate-100 rounded-md. Кнопки швидкого редагування та видалення товару у крайньому правому стовпчику за замовчуванням приховані за допомогою класу opacity-0 та стають видимими лише при наведенні курсору миші на рядок таблиці.

ФОТО	АРТИКУЛ / ВС	НАЙМЕНУВАННЯ	ЦІНА (€)	ЗАЛИШОК	ДІЇ
	GB-600-300-200 4820007654338	Газоблок Аерос 600x300x200	3200	27	
	GL-CER-ST17 4820001234574	Клей Ceresit CM 11 (25 кг)	210	500	
	BR-RED-M100 4820007654321	Цегла рядова М-100	7.2	5000	
	CM-M500-25 4820001234567	Цемент ПЦ-500 (25 кг)	185.5	1000	

Рисунок 2.3 - Список товарів у розділі управління товарами

Для додавання та редагування товарів було розроблено спеціальне вікно.

Форма у лівій частині вікна містить поля для артикулу, штрихкоду, назви, ціни та кількості. Справа знаходиться область для керування зображеннями, де адміністратор може завантажувати файли на server через поле типу file. Вікно також підключає інтегрований сканер штрихкодів, який використовує веб-камеру для зчитування кодів з етикеток. Форму додавання та редагування товарів зображено на рисунку. 2.4.

Візуальний стиль вікна редагування товару використовує ефект затемнення заднього плану за допомогою класу `bg-black/50` та розмиття фону `backdrop-blur-sm`. Контейнер форми отримав фіксовану висоту `max-h-[90vh]` для запобігання виходу за межі екрану та внутрішню вертикальну прокрутку. Поля форми розділено вертикальною лінією `divide-x`. Область завантаження фотографій отримала рамку з пунктирним стилем `border-2 border-dashed border-slate-200`, яка підсвічується синім кольором при наведенні файлу.

The screenshot shows a modal window titled "Редагувати товар" (Edit Item). It is divided into two main sections. The left section contains form fields for: "АРТИКУЛ (SKU)" (GB-600-300-200), "ШТРИХ-КОД" (48200076543), "НАЗВА ТОВАРУ" (Газоблок Аерос 600x300x200), "ЦІНА (€)" (3200), and "КІЛЬКІСТЬ НА СКЛАДІ" (27). The right section, titled "ЗОБРАЖЕННЯ ТОВАРУ" (Item Image), shows a placeholder image of a brick and a "ДОДАТИ ФОТО" (Add Photo) button. At the bottom of the modal, there are two buttons: "Скасувати" (Cancel) and "Зберегти зміни" (Save Changes).

Рисунок 2.4 - Вікно для додавання та редагування товару

Розділ Управління користувачами відображає картки всіх зареєстрованих працівників торгової мережі. Кожна картка містить ім'я, роль у системі,

електронну адресу та поточний статус доступу. Для додавання працівника відкривається форма, де вводять ім'я, пошту, початковий пароль та обирають роль із випадаючого списку. Форма редагування дозволяє змінити статус доступу або оновити пароль працівника. Список облікових записів працівників подано на рисунку 2.5.

Для розділу управління користувачами застосували сітку карток grid-cols-1 з розширенням на великих екранах. Картки персоналу мають білий фон bg-white та окремий блок заголовка з іконкою користувача у круглому сірому колі. Кожна роль працівника отримала індивідуальну кольорову плашку для швидкого розрізнення: наприклад, роль касира або менеджера виділено фірмовим синім текстом на світлому фоні bg-indigo-50. Кнопка редагування отримала сірий нейтральний стиль bg-slate-50, а кнопка деактивації оформлена у світло-червоних тонах bg-rose-50 та text-rose-600.

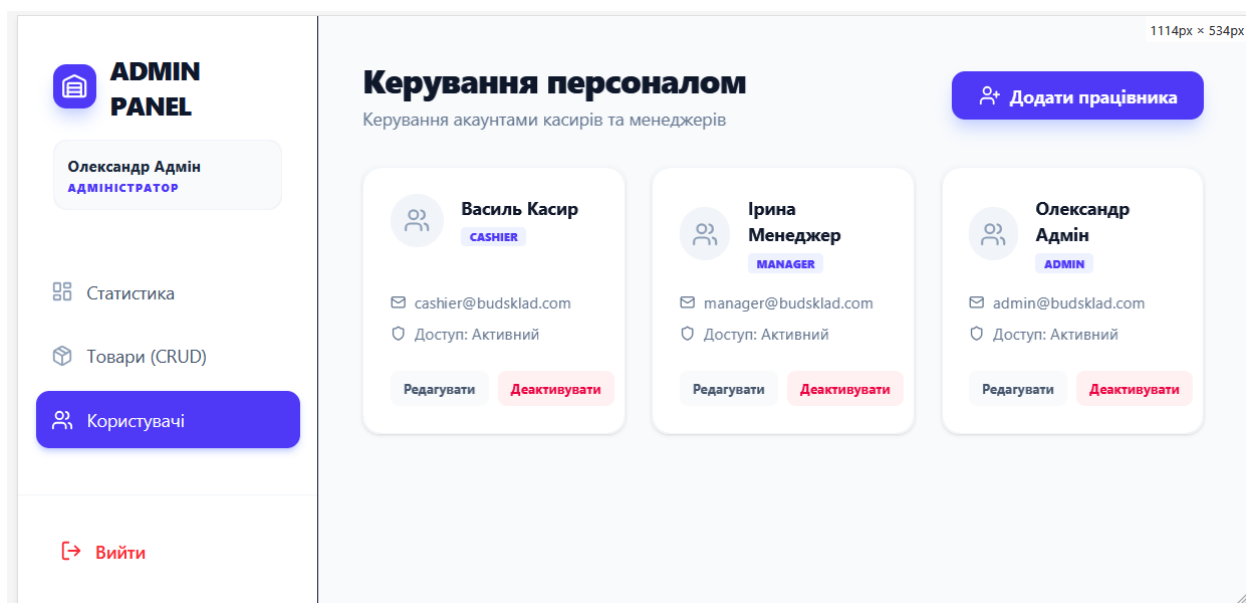


Рисунок 2.5 - Розділ управління користувачами

Для внесення нових облікових записів до бази даних було розроблено окреме вікно з формою введення персональних даних та вибору повноважень працівника. Структуру та поля форми додавання нового користувача наведено на рисунку 2.6.

					2026.КВР.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

Рисунок 2.6 - Форма додавання нового користувача

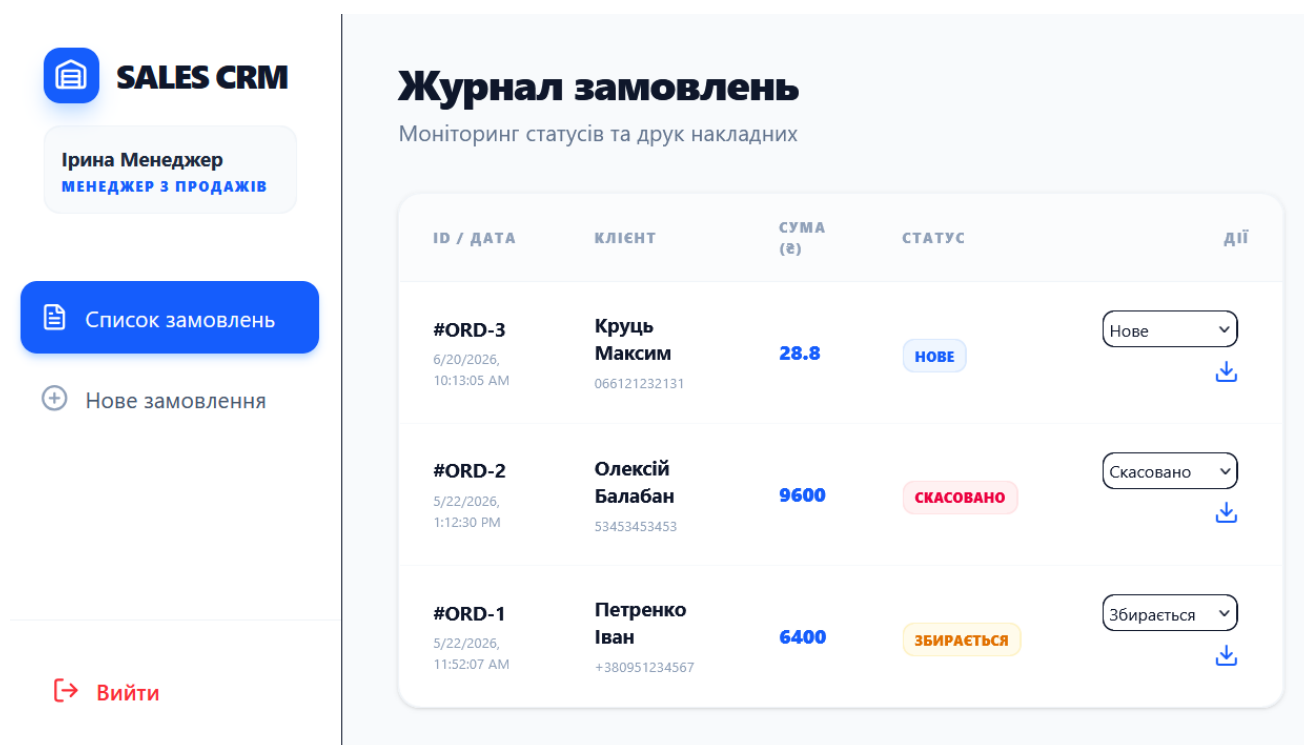
Для модифікації наявних облікових записів, зміни паролів або блокування доступу було реалізовано модальне вікно редагування профілю. Зовнішній вигляд форми зміни параметрів користувача представлено на рисунку 2.7.

Рисунок 2.7 - Форма редагування користувача

Усі елементи інтерфейсу в межах даного розділу було спроектовано з дотриманням принципу одноманітності візуальних форм, що спрощує взаємодію адміністратора з обліковими записами персоналу.

Сторінка менеджера орієнтована на обробку клієнтських замовлень на доставку будматеріалів. Перший розділ містить список замовлень, де у табличному вигляді відображаються номери замовлень, дати створення, імена клієнтів, суми до оплати та статус виконання. Менеджер може змінити статус за допомогою випадаючого списку або завантажити накладну у форматі PDF для друку. Журнал замовлень наведено на рисунку 2.8.

Стилістичною особливістю журналу замовлень є використання кольорових статусних бейджів із заокругленими кутами `rounded-lg` та дрібним жирним текстом `text-[10px] font-black`. Кожен статус замовлення отримав індивідуальне колірне кодування: нове замовлення підсвічується синім кольором `bg-blue-50` та `border-blue-100`, етап збирання товарів помаранчевим `bg-amber-50`, статус відправлено фіолетовим `bg-indigo-50`, успішно доставлено зеленим `bg-emerald-50`, а скасовані замовлення червоним `bg-rose-50`.



ID / ДАТА	КЛІЄНТ	СУМА (€)	СТАТУС	Дії
#ORD-3 6/20/2026, 10:13:05 AM	Круць Максим 066121232131	28.8	НОВЕ	Нове ↓
#ORD-2 5/22/2026, 1:12:30 PM	Олексій Балабан 53453453453	9600	СКАСОВАНО	Скасовано ↓
#ORD-1 5/22/2026, 11:52:07 AM	Петренко Іван +380951234567	6400	ЗБИРАЄТЬСЯ	Збирається ↓

Рисунок 2.8 - Список замовлень на сторінці менеджера

Розділ Нове замовлення складається з форми підбору товарів та форми введення даних клієнта. Зліва менеджер шукає матеріали в каталозі та додає їх у кошик замовлення, регулюючи кількість кнопками плюс та мінус. Справа розміщено блок для введення імені, телефону та адреси доставки. Внизу відображається загальна сума замовлення та велика кнопка для його збереження на сервері. Візуальний інтерфейс формування замовлення зображено на рисунку 2.9.

Для інтерфейсу нового замовлення використали двоколонковий макет на базі CSS Grid з пропорцією колонок 2 до 1. Спливаюче вікно пошуку товарів отримало абсолютне позиціонування absolute, тіньовий ефект shadow-xl та обмеження видимості overflow-hidden, щоб результати пошуку не зміщували інші елементи інтерфейсу. Блок регулювання кількості товарів має вигляд єдиного сірого контейнера bg-slate-50, розділеного тонкими лініями, з кнопками зміни значень та жирним шрифтом для числа.

Рисунок 2.9 - Створення нового замовлення на сторінці менеджера

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		40

Сторінка касира розроблена спеціально для обслуговування покупців. Вона розділена на два основні екрани. Перший екран виконує функції керування зміною. Тут касир відкриває зміну перед початком робочого дня, вказуючи суму грошей для розміну. В кінці дня касир бачить загальну суму продажів та закриває зміну, після чого програма блокує термінал та формує Z-звіт. Екран для управління касовою зміною касира показано на рисунку 2.10.

Стилістичне оформлення розділу керування зміною виконане повністю у темних тонах на базі глибокого темно-синього фону `bg-slate-900` для зменшення навантаження на очі при роботі за монітором. Картки дій отримали контрастні рамки `border-2`, які змінюють колір залежно від стану зміни: зелений колір `border-emerald-500` для активного стану відкриття зміни та червоний `border-rose-500` для картки закриття. Неактивні дії автоматично затемнюються за допомогою класу `opacity-50 grayscale` та стають недоступними для натискання.

Рисунок 2.10 - Розділ керування зміною касира

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		41

Другий екран касира це робочий термінал для сканування та продажу товарів. Зліва знаходиться велике поле пошуку з підтримкою гарячих клавіш та таблиця поточного чека. Справа було розміщено панель вибору способу оплати готівкою або карткою та вікно розрахунку. Програма автоматично розраховує суму решти при введенні готівки від покупця. Після успішної оплати з'являється вікно завершення транзакції. Візуальний вигляд робочого терміналу касира зображено на рисунку 2.11.

Робочий термінал касира використовує інтерфейс із темною кольоровою гамою та великими шрифтами для кращої видимості. Поле пошуку отримало збільшений розмір тексту text-xl та рамку, яка підсвічується зеленим кольором focus:ring-emerald-500/50. Панель вибору способу оплати містить дві великі кнопки-вкладки з товстою рамкою border-2, які стають яскраво-зеленими при виборі та зафарбовуються напівпрозорим кольором bg-emerald-500/10. Модальне вікно успішного чека отримало плавний ефект збільшення animate-in zoom-in-95.

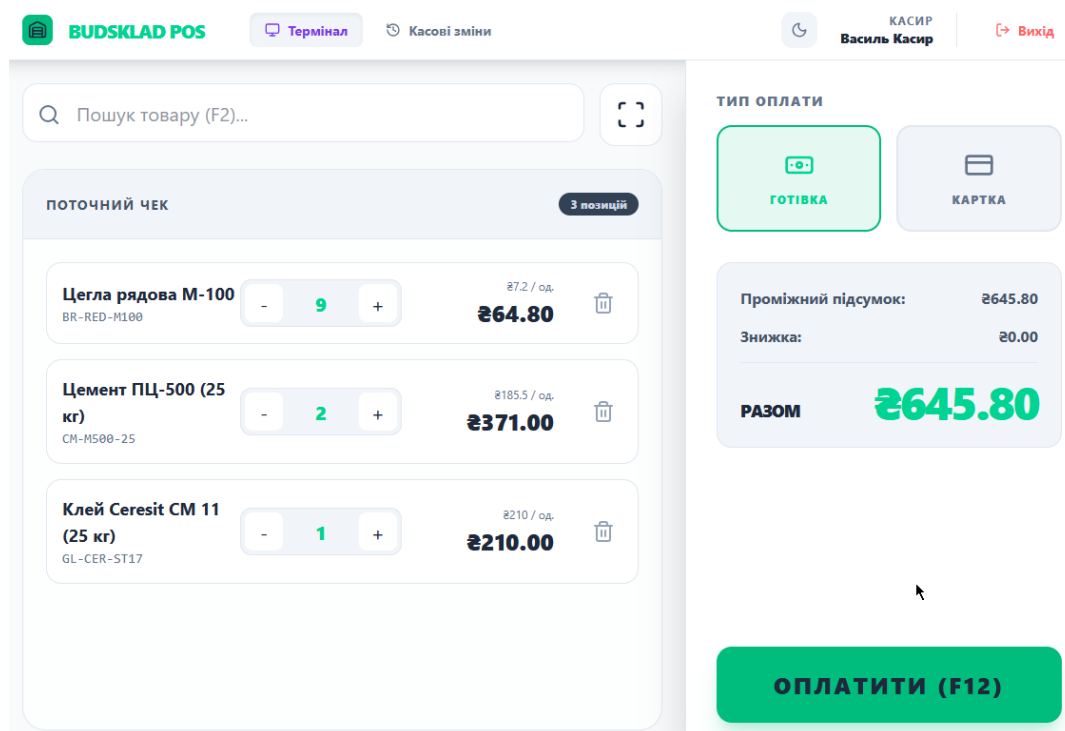


Рисунок 2.11 - Термінал касира

Структуру верстки платіжного блоку робочого місця касира подано на

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

лістингу 2.3.

```
<div className="bg-slate-900 border-1 border-slate-800 p-6 flex flex-col justify-between h-full">
  <div className="space-y-6">
    <h2 className="text-xl font-bold text-white">Оплата замовлення</h2>

    <div className="grid grid-cols-2 gap-4">
      <button className="flex flex-col items-center justify-center p-4 border-2 border-emerald-500 bg-emerald-500/10 rounded-xl text-white">
        <span className="text-sm font-semibold">Готівка</span>
      </button>

      <button className="flex flex-col items-center justify-center p-4 border-2 border-slate-800 rounded-xl text-slate-400">
        <span className="text-sm font-semibold">Картка</span>
      </button>
    </div>
  </div>

  <div className="border-t border-slate-800 pt-6">
    <div className="flex justify-between text-lg text-slate-400">
      <span>До сплати:</span>
      <span className="font-bold text-white text-2xl">0.00 €</span>
    </div>
  </div>
</div>
```

Лістинг 2.3 - Розмітка платіжного блоку у POSTerminal.tsx

У цьому лістингу представлено структуру верстки платіжного блоку робочого місця касира. Розмітка містить панель вибору способу оплати у вигляді кнопок-вкладок та секцію підсумкової вартості чека. Усі елементи оформлено у темній кольоровій гамі за допомогою відповідних класів відступів, меж та

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

кольорів тексту, що забезпечує чітке візуальне розділення інтерфейсних зон. Повний код компонента POSTerminal.tsx подано в додатку В пояснювальній записки.

У результаті виконаної роботи було повністю зверстано всі інтерфейсні шаблони сторінок інформаційної системи. Кожен розроблений компонент відповідає вимогам адаптивності та коректно відображається на різних роздільних здатностях екранів. Використаний підхід дозволив отримати високу швидкість завантаження елементів та спростити щоденну роботу персоналу будівельного підприємства.

2.3 Розробка структури бази даних інформаційної системи

Для забезпечення надійного збереження даних, високої швидкості обробки запитів та підтримання цілісності інформації було спроектовано структуру реляційної бази даних. Простування бази даних будівельного складу було реалізовано на основі концептуальної моделі предметної області, що відображає основні бізнес-процеси підприємства: облік товарних запасів, авторизацію користувачів із розподілом ролей, реєстрацію касових змін, формування чеків та обробку замовлень на доставку. Базу даних було розгорнуто під керуванням СКБД PostgreSQL [5], яка забезпечує підтримку транзакцій ACID та високу продуктивність при одночасній роботі багатьох користувачів. Зв'язки між таблицями та типи даних було описано за допомогою схеми Prisma ORM [6], що дозволило автоматизувати процес генерації міграцій та гарантувати відповідність структури бази даних коду програми.

Логічну структуру та зв'язки між сутностями системи представлено на плакаті 2026.КВР.122.423.09.00.00 БД.

База даних складається з восьми основних таблиць, кожна з яких виконує певну роль у збереженні бізнес-інформації та підтриманні зв'язків:

– Таблиця users призначена для збереження облікових записів працівників торгової мережі та містить такі поля:

					2026.КВР.122.423.09.00.00 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

- id (тип Int, первинний ключ) - унікальний ідентифікатор запису з автоматичним збільшенням;
- email (тип String, унікальний індекс) - адреса електронної пошти користувача для авторизації;
- password_hash (тип String) - хеш пароля для безпечного збереження облікових даних;
- name (тип String) - ім'я та прізвище працівника для виведення в інтерфейсі;
- role (тип Role, перелік) - роль працівника в системі (ADMIN, MANAGER або CASHIER);
- created_at (тип DateTime) - час створення облікового запису;
- updated_at (тип DateTime) - час останнього оновлення профілю.
- Таблиця products містить номенклатуру будівельних матеріалів складу та містить такі поля:
 - id (тип Int, первинний ключ) - унікальний ідентифікатор товару;
 - sku (тип String, унікальний індекс) - текстовий артикул товару для внутрішнього обліку;
 - barcode (тип String, унікальний індекс) - штрихкод товару для швидкого сканування;
 - name (тип String) - найменування будівельного матеріалу;
 - price (тип Decimal) - ціна товару за одиницю з точністю до двох знаків після коми;
 - stock_quantity (тип Decimal) - кількість на складі з точністю до трьох знаків після коми;
 - created_at (тип DateTime) - час додавання позиції до каталогу;
 - updated_at (тип DateTime) - час останньої зміни інформації про товар.
- Таблиця product_images призначена для збереження посилань на фотографії товарів та містить такі поля:

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

- id (тип Int, первинний ключ) - унікальний ідентифікатор запису;
 - product_id (тип Int, зовнішній ключ) - зв'язок з таблицею товарів з каскадним видаленням;
 - url (тип String) - адреса файлу зображення у хмарному сховищі;
 - public_id (тип String) - ідентифікатор медіафайлу для адміністрування сховища;
 - created_at (тип DateTime) - час завантаження фотографії.
- Таблиця cash_shifts фіксує відкриття та закриття касових змін касирами та містить такі поля:
- id (тип Int, первинний ключ) - унікальний ідентифікатор касової зміни;
 - cashier_id (тип Int, зовнішній ключ) - зв'язок з таблицею користувачів;
 - opened_at (тип DateTime) - час початку касової зміни;
 - closed_at (тип DateTime, може бути порожнім) - час завершення касової зміни;
 - start_cash (тип Decimal) - сума готівки в касі на момент початку зміни;
 - end_cash (тип Decimal, може бути порожнім) - сума готівки на момент закриття зміни;
 - total_sales (тип Decimal) - загальний обсяг продажів за поточну зміну.
- Таблиця receipts використовується для реєстрації чеків продажів та містить такі поля:
- id (тип Int, первинний ключ) - унікальний номер касового чека;
 - shift_id (тип Int, зовнішній ключ) - зв'язок з відповідною касовою зміною;
 - total_amount (тип Decimal) - загальна сума продажу за чеком;
 - payment_type (тип PaymentType, перелік) - спосіб оплати (CASH,

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

CARD або POSTPAID);

- createdAt (тип DateTime) - час проведення транзакції.
- Таблиця receipt_items деталізує вміст чеків продажу та містить такі поля:

id (тип Int, первинний ключ) - унікальний ідентифікатор рядка чека;

receipt_id (тип Int, зовнішній ключ) - зв'язок з головним чеком з каскадним видаленням;

product_id (тип Int, зовнішній ключ) - зв'язок з проданим товаром;

quantity (тип Decimal) - кількість реалізованого товару;

price_at_sale (тип Decimal) - ціна реалізації товару на момент формування чека.

- Таблиця orders містить інформацію про замовлення на доставку та містить такі поля:

- id (тип Int, первинний ключ) - унікальний ідентифікатор замовлення;

- manager_id (тип Int, зовнішній ключ) - зв'язок з менеджером, який створив замовлення;

- client_name (тип String) - ім'я та прізвище клієнта;

- client_phone (тип String, може бути порожнім) - контактний номер телефону замовника;

- delivery_address (тип String, може бути порожнім) - адреса доставки товарів;

- status (тип OrderStatus, перелік) - поточний стан замовлення;

- total_amount (тип Decimal) - загальна вартість замовлення;

- payment_status (тип PaymentStatus, перелік) - статус оплати (UNPAID або PAID);

- created_at (тип DateTime) - час створення замовлення;

- updated_at (тип DateTime) - час останньої модифікації замовлення.

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>47</i>

– - Таблиця `order_items` містить специфікацію товарів для доставки та містить такі поля:

- `id` (тип `Int`, первинний ключ) - унікальний ідентифікатор рядка замовлення;
- `order_id` (тип `Int`, зовнішній ключ) - зв'язок з головним замовленням з каскадним видаленням;
- `product_id` (тип `Int`, зовнішній ключ) - зв'язок з товаром;
- `quantity` (тип `Decimal`) - кількість замовлених одиниць будівельного матеріалу;
- `price_at_sale` (тип `Decimal`) - ціна продажу товару на момент оформлення замовлення.

Спроектвана схема бази даних повністю забезпечує зберігання всієї необхідної інформації про роботу будівельного складу. Завдяки налаштуванню зовнішніх ключів та каскадних правил видалення було досягнуто високого рівня цілісності даних. Використання індексів для унікальних полів дозволило значно прискорити пошук товарів за штрихкодами та авторизацію користувачів за адресою електронної пошти.

2.4 Програмування інформаційної системи

Програмування функціоналу інформаційної системи будівельного складу полягало в інтеграції клієнтської та серверної частин у єдиний програмний комплекс, що забезпечує автоматизацію обліку товарів та обслуговування покупців. Написання програмного коду скриптів було виконано в інтегрованому середовищі розробки Visual Studio Code. Цей текстовий редактор було обрано через високу швидкість роботи, гнучкість налаштувань, вбудовану підтримку системи контролю версій Git та велику кількість розширень для автодоповнення коду TypeScript і стилів Tailwind CSS. Проєкт побудовано за клієнт-серверною

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

архітектурою, де інтерфейс користувача взаємодіє з сервером бази даних за допомогою прикладного програмного інтерфейсу на основі протоколу HTTP та архітектурного стилю REST.

2.4.1 Написання клієнтської частини

Розробку клієнтської частини системи було реалізовано на базі сучасного фреймворку React [7] із використанням інструменту збірки Vite та мови програмування TypeScript. Для створення початкової структури проєкту було виконано команду ініціалізації `npm create vite` з вибором шаблону React-TypeScript. Після цього було встановлено основні залежності програми: бібліотеку `react-router-dom` для організації маршрутизації, менеджер стану `zustand` для управління глобальними даними, бібліотеку `react-query` для кешування запитів до сервера та `lucide-react` для виведення векторних іконок. В результаті було сформовано модульну структуру каталогів клієнтської частини додатку, яку представлено на рис. 2.12.

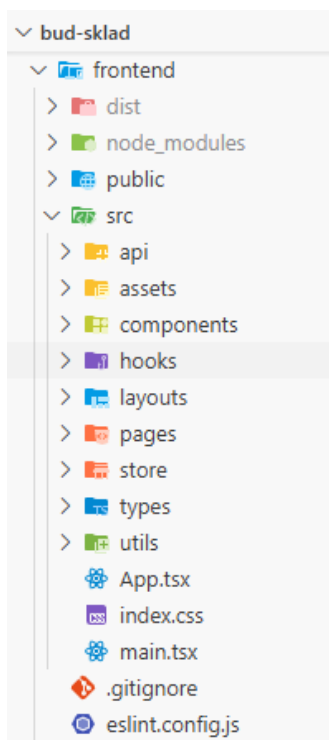


Рисунок 2.12 - Структура каталогів клієнтської частини інформаційної системи

Для полегшення навігації та підтримки чистоти коду всі файли структуровано за відповідними каталогами:

- `api` - містить налаштування клієнта для виконання запитів до сервера;
- `components` - містить загальні компоненти інтерфейсу користувача;
- `hooks` - містить користувацькі React-хуки для винесення повторюваної логіки;
- `layouts` - містить шаблони розмітки сторінок для різних ролей працівників;
- `pages` - містить компоненти окремих екранів та сторінок додатку;
- `store` - містить сховища стану Zustand для глобального управління даними;
- `types` - містить описи інтерфейсів та типів TypeScript;
- `utils` - містить допоміжні функції спільного використання.

Принцип функціонування клієнта базується на компонентному підході, де кожна сторінка складається з окремих незалежних модулів, що повторно використовуються. Головним вхідним файлом програми є `App.tsx`, у якому налаштовано провайдери контексту та дерево маршрутів.

Доступ до сторінок обмежено за допомогою спеціального компонента `RoleGuard`, який перевіряє роль авторизованого користувача перед відображенням контенту. Якщо користувач не має відповідних прав або не пройшов аутентифікацію, система автоматично перенаправляє його на головну сторінку або екран входу.

Управління станом авторизації працівників побудовано на основі сховища Zustand. Скрипт ініціалізації сховища та методи керування даними сесії користувача представлено на лістингу 2.4.

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

localStorage та оновлює стан програми для негайного оновлення інтерфейсу. Метод logout видаляє дані з пам'яті та сховища при виході користувача із системи.

Для обробки замовлень покупців та формування чеків у робочому місці менеджера та касира було розроблено сховище кошика товарів.

У цьому лістингу показано структуру сховища кошика товарів, яке використовує посередник persist для автоматичної синхронізації вмісту з локальним сховищем. Метод addItem виконує перевірку наявності товару у списку: якщо товар уже додано, його кількість збільшується, інакше створюється новий елемент. Методи removeItem та updateQuantity дозволяють відповідно видаляти позиції та коригувати кількість будівельних матеріалів. Повний код файлу useCartStore.ts подано в додатку Г пояснювальної записки.

2.4.2 Написання серверної частини

Для розробки серверної частини інформаційної системи було обрано програмну платформу Node.js [8] та фреймворк Express. Серверна частина виконує обробку бізнес-логіки системи, взаємодіє з базою даних через Prisma [6] ORM та надає захищений інтерфейс прикладного програмування для клієнтської частини [9]. Для захисту даних та контролю доступу було реалізовано авторизацію на основі токенів JSON Web Token (JWT). Перевірка вхідних даних здійснюється за допомогою бібліотеки Zod, що гарантує відповідність типу даних перед збереженням у базі даних.

Для підтримання чистоти та простоти архітектури вихідний код сервера було розділено за функціональними каталогами:

- controllers - містить класи контролерів, які обробляють запити від клієнта;
- middlewares - містить проміжні обробники для перевірки авторизації та завантаження файлів;
- routes - визначає кінцеві точки доступу та пов'язує їх з обробниками;

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

- schemas - описує схеми валідації вхідних даних;
- services - реалізує бізнес-логіку та безпосередні запити до бази даних;
- utils - містить інструменти для структурування стандартних відповідей сервера.

Безпека системи забезпечується проміжними обробниками, які перевіряють наявність дійсного токена авторизації у заголовках запитів клієнта. Повний код проміжного програмного забезпечення для автентифікації та авторизації користувачів `auth.middleware.ts` подано в додатку Д пояснювальної записки.

У цьому файлі реалізовано логіку перевірки токена доступу у проміжному обробнику `authenticate`. Метод вилучає токен із заголовка `Authorization`, перевіряє його дійсність за допомогою секретного ключа та записує розшифровані дані про ідентифікатор і роль користувача в об'єкт запити `req.user`. Обробник `authorize` перевіряє відповідність ролі користувача списку дозволених ролей для поточної кінцевої точки.

Обробка запитів від касового терміналу виконується в класі контролера. Повний код контролера обробки касових операцій `pos.controller.ts` подано в додатку Е пояснювальної записки.

У цьому файлі описано структуру методів контролера `POSController`. Кожен метод виконує валідацію вхідного тіла запити за допомогою відповідної схеми `Zod`, після чого передає очищені дані до методів класу сервісу `POSService` для безпосередньої роботи з базою даних. У разі виникнення помилок вони передаються до наступного обробника через функцію `next` для централізованої обробки.

2.5 Тестування інформаційної системи

Для перевірки працездатності, стабільності та відповідності розробленого програмного забезпечення технічним вимогам було виконано комплексне тестування інформаційної системи будівельного складу. В процесі перевірки

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

було застосовано різні методи та види тестування, які дозволили всебічно оцінити якість роботи клієнтської та серверної частин додатку, а також надійність збереження даних. Детальний план тестування (Test Plan) розробленої системи представлено в додатку 3 пояснювальної записки.

При виконанні перевірки було використано такі основні види тестування:

– Функціональне тестування полягало у верифікації відповідності реалізованих функцій системи вимогам технічного завдання. Перевірку було спрямовано на бізнес-логіку додатку: процес авторизації працівників із різними рівнями доступу, операції додавання, редагування та видалення будівельних матеріалів у каталозі, створення нових замовлень менеджерами, а також проведення торгових транзакцій через касовий термінал. Кожну функцію перевірено шляхом моделювання реальних сценаріїв використання системи персоналом складу.

– Тестування інтерфейсу користувача було орієнтовано на перевірку коректності візуального відображення елементів на екрані. Було виконано тестування адаптивності верстки при зміні роздільної здатності моніторів та мобільних пристроїв. Особливу увагу приділено поведінці інтерактивних компонентів: анімації завантаження (спінерів) при очікуванні відповіді від сервера, підсвічуванню активних полів введення, доступності кнопок та коректності відображення кольорових статусних плашок замовлень і ролей.

– Інтеграційне тестування проводилося з метою перевірки взаємодії між окремими програмними модулями клієнтської та серверної частин. Було протестовано правильність обробки HTTP-запитів клієнта, передачі параметрів у форматі JSON, стабільність підключення до сервера бази даних через Prisma ORM та коректність виконання транзакцій при списанні залишків товарів зі складу після проведення чека.

– Тестування безпеки було спрямовано на перевірку механізмів захисту інформації. Було проведено тестування обмеження доступу до сторінок за допомогою компонента RoleGuard. Перевірено неможливість переходу до адмін-

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		54

панелі користувачами з роллю касира чи менеджера. Також було проконтрольовано блокування запитів до серверної частини при відсутності або некоректності токена JWT у заголовках запитів із поверненням статусів помилок HTTP 401 та 403.

Для систематизації процесу перевірки було сформовано контрольний список (checklist) протестованих компонентів системи:

- авторизація користувачів (валідація полів введення пошти та пароля, перенаправлення на сторінки відповідно до ролі працівника);
- статистика системи (коректність зчитування та відображення загальних показників продажів, кількості товарів та активних користувачів);
- управління товарами (створення нових карт товарів, пошук за артикулом чи штрихкодом, редагування параметрів, завантаження зображень);
- управління користувачами (реєстрація нових працівників, зміна паролів та блокування доступу до системи);
- список замовлень менеджера (відображення поточної інформації про клієнтів, зміна статусів доставки та генерація друкованих форм);
- касовий термінал (відкриття та закриття робочої зміни, сканування штрихкодів товарів, розрахунок решти покупця та вибір способу оплати);
- безпека даних (шифрування паролів у базі даних, перевірка терміну дії сесії авторизації користувача).

В результаті проведеного тестування було підтверджено повну працездатність усіх модулів системи. Виявлені в процесі перевірки незначні помилки інтерфейсу було оперативно виправлено, що дозволило забезпечити високу стабільність функціонування інформаційної системи будівельного складу.

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		55

3 СПЕЦІАЛЬНИЙ РОЗДІЛ

Спеціальний розділ розкриває питання практичного розгортання готової системи на хмарних серверах в мережі інтернет. Текст містить детальний опис кроків публікації програмного продукту та налаштування бази даних. Наступна частина пояснює інструкції для персоналу магазину щодо наповнення каталогу будівельних товарів та щоденного обслуговування. Останній підрозділ пропонує бізнес-модель просування розробленого продукту на ринку спеціалізованих програмних рішень.

3.1 Інструкція з розгортання інформаційної системи

Для розміщення та забезпечення стабільного функціонування інформаційної системи будівельного складу в мережі інтернет було розроблено покрокову процедуру розгортання її компонентів. Хостинг системи базується на використанні хмарних платформ (PaaS), що дозволяє автоматизувати обслуговування інфраструктури та спростити процес публікації додатку.

Процедура розгортання хмарної бази даних під керуванням СКБД PostgreSQL на платформі Supabase складається з таких кроків [10]:

- Крок 1. Зареєструйтеся або авторизуйтеся на веб-порталі хмарної платформи Supabase.
- Крок 2. Створіть новий проєкт (New Project) у робочому просторі організації, вкажіть назву бази даних, задайте пароль для адміністратора бази та оберіть найближчий географічний регіон сервера.
- Крок 3. Після створення бази даних перейдіть до розділу Database Settings та скопіюйте рядок підключення Connection String у форматі URI для зв'язку з сервером. Конфігурацію параметрів підключення до бази даних зображено на рисунку 3.1.

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		56

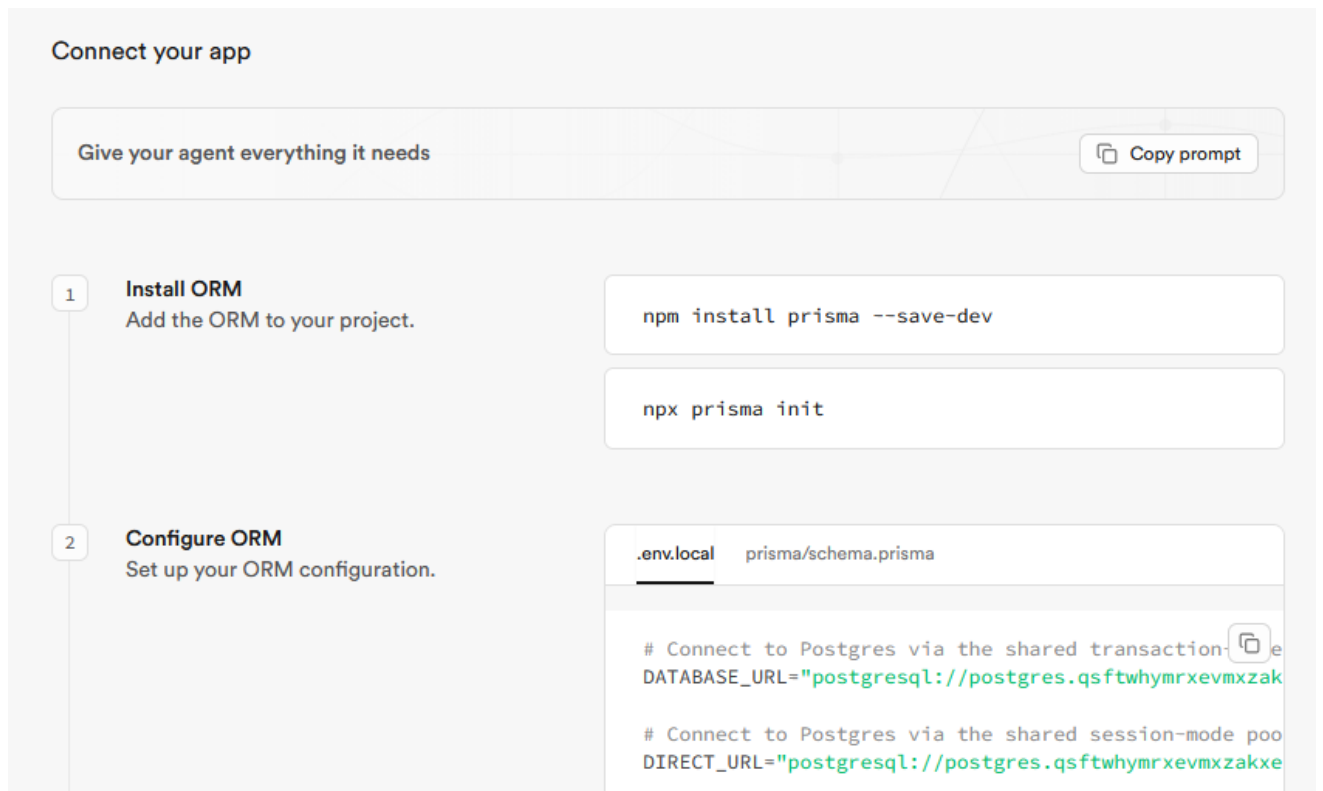


Рисунок 3.1 - Налаштування параметрів бази даних на платформі Supabase

Процедура розгортання серверної частини додатку на платформі Render складається з таких кроків [11]:

- Крок 4. Авторизуйтеся на хмарному хостингу Render та підключіть власний обліковий запис GitHub.
- Крок 5. Створіть нову веб-службу Web Service та оберіть репозиторій з кодом серверної частини додатку.
- Крок 6. У налаштуваннях веб-служби вкажіть команду збірки `npm install && npm run build`, а командою для запуску сервера призначте `node dist/index.js`.
- Крок 7. У розділі Environment Variables додайте змінні оточення `DATABASE_URL` (із підставленим рядком підключення URI від Supabase) та `JWT_SECRET` (секретний ключ для авторизації). Параметри веб-служби бекенду наведено на рисунку 3.2.

Branch

The Git branch to build and deploy.

main

Edit

Root Directory Optional

If set, Render runs commands from this directory instead of the repository root. Additionally, code changes outside of this directory do not trigger an auto-deploy. Most commonly used with a [monorepo](#).

backend

Edit

Build Command

Render runs this command to build your app before each deploy.

backend/ \$ npm install && npm run build

Edit

Рисунок 3.2 - Параметри розгортання веб-служби бекенду на платформі Render

– Крок 8. Для застосування структури таблиць до бази даних з локального комп'ютера розробника виконайте команду Prisma міграції `prisma db push`, після чого на Render запустить автоматичний деплой.

Процедура розгортання клієнтського інтерфейсу на хостингу Vercel складається з таких кроків [12]:

– Крок 9. Перейдіть до панелі керування Vercel та імпортуйте репозиторій з кодом клієнтської частини додатку.

– Крок 10. У конфігурації проєкту додайте змінну оточення `VITE_API_URL`, значенням якої вкажіть URL-адреса запущеної на Render веб-служби бекенду.

– Крок 11. Натисніть кнопку Deploy для запуску автоматичної збірки статичних файлів React додатку за допомогою Vite. Результат успішного розгортання клієнтської частини в мережі інтернет відображено на рисунку 3.3.

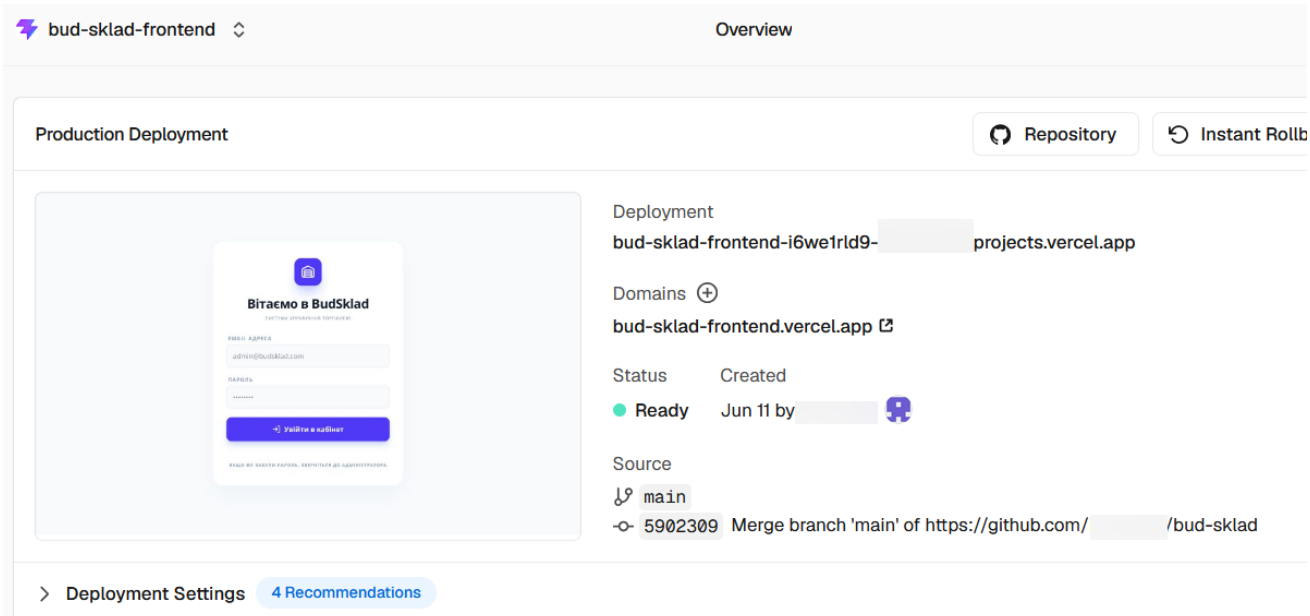


Рисунок 3.3 - Статус успішного розгортання інтерфейсу у панелі Vercel

У результаті виконання зазначених кроків було повністю завершено публікацію всіх складових частин інформаційної системи будівельного складу в мережі інтернет. Завдяки використанню сучасних хмарних PaaS-платформ було отримано стійке до навантажень робоче середовище додатку, що дозволяє користувачам взаємодіяти із базою даних у реальному часі.

3.2 Інструкція з обслуговування та наповнення інформаційної системи

Для підтримання стабільного функціонування розробленої системи обліку будівельних матеріалів та забезпечення актуальності каталогу продукції встановлено чіткі правила обслуговування. Процес технічного супроводу спрямовано на запобігання втраті даних, підтримання працездатності серверного обладнання та забезпечення безперебійної роботи касових вузлів.

Перелік регулярних дій для технічного обслуговування системи складається з таких операцій:

- щоденна перевірка доступності веб-ресурсів на хостингах Render та Vercel,

- щотижневе створення резервних копій бази даних через інтерфейс Supabase для збереження інформації про продажі,
- моніторинг обсягу вільного місця на хмарному сховищі зображень товарів,
- щомісячний аналіз швидкодії виконання запитів до бази даних PostgreSQL,
- контроль збігу фізичних залишків товарів на складі відомостям у віртуальному каталозі,
- перевірка журналів авторизації користувачів для виявлення підозрілих спроб входу в систему,
- оновлення версій системних бібліотек та конфігураційних файлів для усунення вразливостей.

Адміністративна частина веб-сайту надає повний контроль над товарними запасами та користувацькими профілями. Для взаємодії з інтерфейсом керування адміністратор використовує персональний комп'ютер з підключеним доступом до мережі інтернет та будь-яким сучасним веб-браузером.

Процедура додавання нового будівельного матеріалу до каталогу складається з таких кроків:

- Крок 1. Перейдіть до головного меню адміністративної панелі та відкрийте розділ управління товарами.
- Крок 2. Знайдіть у правому верхньому куті кнопку додавання нового товару та натисніть на неї.
- Крок 3. Введіть назву будівельного матеріалу у першому полі форми, уникайте занадто довгих описів.
- Крок 4. Заповніть поле артикулу або скористайтеся сканером для автоматичного зчитування фабричного штрихкоду з упаковки товару.
- Крок 5. Вкажіть вартість однієї одиниці продукції у полі ціни та поточну кількість матеріалу на складі.
- Крок 6. Виберіть потрібну категорію товару із випадючого списку

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

для правильної класифікації у каталозі.

– Крок 7. Завантажте фотографію виробу, перетягнувши файл у спеціальну область форми або обравши його на диску комп'ютера.

– Крок 8. Натисніть кнопку збереження для надсилання інформації на сервер та запису даних у базу. Зовнішній вигляд сторінки після внесення інформації наведено на рисунку 3.4.

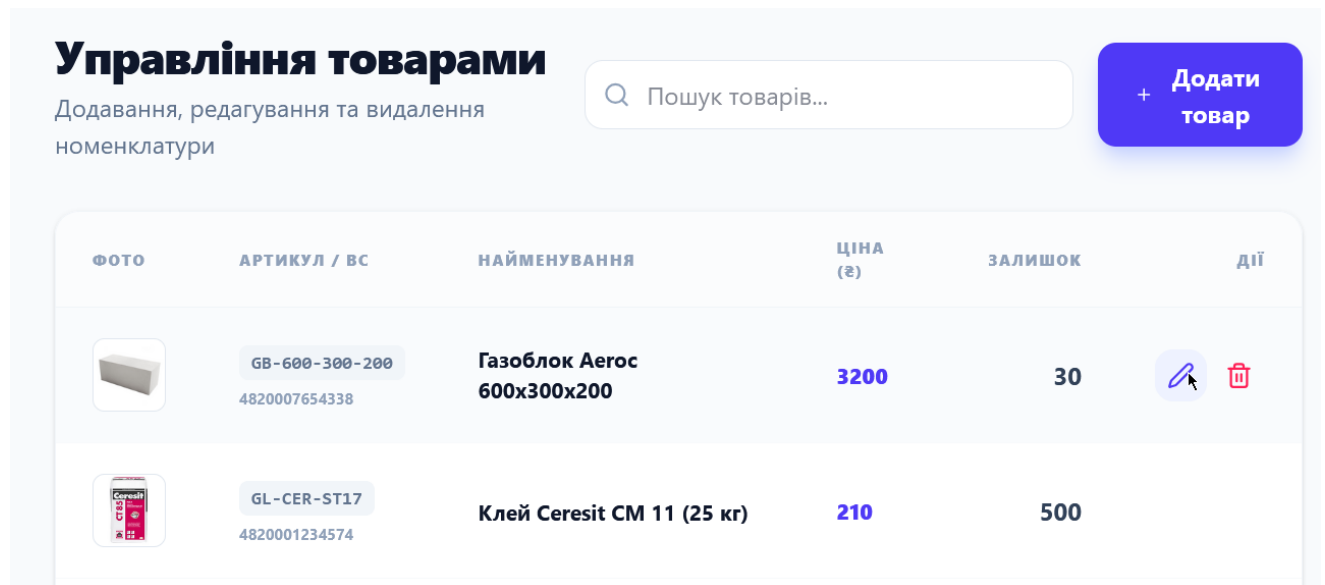


Рисунок 3.4 - Інтерфейс каталогу товарів

Процедура редагування параметрів наявного будівельного матеріалу складається з таких кроків:

- Крок 9. Знайдіть необхідну позицію у таблиці товарів за назвою або артикулом за допомогою рядка швидкого пошуку.

- Крок 10. Натисніть кнопку редагування з іконкою олівця, яка розташована у правому стовпчику таблиці навпроти назви товару.

- Крок 11. Змініть застарілу інформацію у формі, наприклад, оновіть роздрібну ціну чи відкоригуйте опис. Процес заповнення форми наведено на рисунку 3.5.

Додати новий товар
×

АРТИКУЛ (SKU) <input type="text" value="23423423"/>	ШТРИХ-КОД <input type="text" value="234234234"/>	ЗОБРАЖЕННЯ ТОВАРУ <div style="border: 1px dashed gray; height: 100px; display: flex; align-items: center; justify-content: center;"> </div> <p style="text-align: center; font-size: small;">ЗОБРАЖЕННЯ МОЖНА ДОДАТИ ПІСЛЯ СТВОРЕННЯ ТОВАРУ</p>
НАЗВА ТОВАРУ <input type="text" value="Монтажна піна 1л"/>		
ЦІНА (€) <input type="text" value="500"/>	КІЛЬКІСТЬ НА СКЛАДІ <input type="text" value="30"/>	
<div style="display: flex; justify-content: space-around;"> Скасувати Зберегти зміни </div>		

Рисунок 3.5 - Процес заповнення інформації про новий матеріал

– Крок 12. Натисніть кнопку збереження змін для оновлення інформації в базі даних та закриття форми редагування.

Процедура видалення застарілих або помилково внесених товарів складається з таких кроків:

- Крок 13. Знайдіть потрібний будівельний матеріал у таблиці каталогу.
- Крок 14. Натисніть кнопку видалення з іконкою червоного сміттевого кошика в кінці цього рядка.
- Крок 15. Підтвердьте видалення у діалоговому вікні для уникнення випадкової втрати даних із бази.

Для стабільного хостингу серверної частини системи визначено мінімальні апаратні вимоги до віртуального виділеного сервера. Обчислювальна потужність процесора має становити не менше одного віртуального ядра з тактовою частотою від двох гігагерц. Мінімальний обсяг оперативної пам'яті сервера має становити два гігабайти, що дозволяє паралельно виконувати запити до бази даних та обробляти сесії користувачів. Для збереження операційної системи та логів роботи сервера необхідно виділити щонайменше двадцять гігабайт на швидкому твердотільному накопичувачі стандарту SSD. Головний

каталог фотографій зберігається у зовнішньому хмаровому сховищі, тому локальний диск сервера використовується ощадливо.

3.3 Інструкція з популяризації та підтримки інформаційної системи

Створений програмний продукт має великий комерційний потенціал на вітчизняному ринку. Багатий функціонал системи повністю вирішує щоденні проблеми багатьох магазинів будівельних матеріалів. Проект використовує продаж платних ліцензій іншим малим підприємствам як головну бізнес-модель. Цей популярний формат поширення спеціалізованого програмного продукту працює за відомими принципами B2B.

Продаж комерційних ліцензій передбачає надання постійного доступу до хмарної системи за фіксовану щомісячну плату. Кожен новий клієнт платформи отримує повністю ізольований простір всередині загальної бази даних. Конфіденційна інформація різних будівельних магазинів абсолютно не перетинається між собою.

Просування такого складного спеціалізованого продукту кардинально відрізняється від масової реклами звичайних споживчих товарів. Успішний продаж дорогої ліцензії вимагає прямого особистого контакту з власниками будівельного бізнесу. Маркетологи завжди починають роботу зі складання детальної бази потенційних клієнтів компанії. Менеджери активно збирають прямі контакти локальних будівельних магазинів та великих гуртових баз.

Спеціаліст з продажу телефонує керівникам підприємств та пропонує коротку онлайн-презентацію нової системи. Розмова будується виключно навколо швидкого розв'язання гострих проблем власника конкретного магазину. Менеджер яскраво розповідає про надійний контроль залишків та радикальне зменшення крадіжок недобросовісним персоналом. Фахівець наочно демонструє зручність проведення інвентаризації та високу швидкість обслуговування клієнтів на касі.

Дієвим методом залучення уваги підприємців виступає створення

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

повноцінного демонстраційного сайту. Технічний адміністратор розгортає повну копію програми з тестовими записами на окремому рекламному домені. Зацікавлений власник бізнесу може цілком самостійно перевірити роботу всіх модулів системи. Керівник створює кілька тестових роздрібних продажів та бачить блискавичну реакцію програми в реальному часі. Надання безкоштовного тестового періоду на чотирнадцять днів сильно підвищує фінальну вірогідність успішної покупки повноцінної ліцензії.

Маркетологи створюють спеціальну цільову сторінку в інтернеті для активного залучення гарячих клієнтів. На цьому рекламному сайті спеціалісти детально розписують всі об'єктивні технічні переваги системи складського обліку. Сторінка містить реальні позитивні відгуки від перших магазинів. Додані зображення гарно показують зручний інтерфейс програми та описують корисну функцію автоматичного сканування штрихкодів.

Рекламний відділ компанії направляє якісний платний трафік на цю створену посадкову сторінку. Спеціаліст грамотно налаштовує контекстну рекламу в найпопулярніших пошукових системах мережі. Розумна система показує рекламні оголошення лише людям, які активно шукають тематичні запити на кшталт "програма для будівельного магазину" або "онлайн облік складу будматеріалів". Цей вузькоспеціалізований трафік коштує дорого, але завжди приносить конверсію високого рівня. Зацікавлені відвідувачі сайту самостійно залишають заявку на розширену технічну консультацію через форму зворотного зв'язку.

Маркетинговий план також обов'язково містить просування у професійних соціальних мережах та на ділових бізнес-форумах. Досвідчені копірайтери пишуть глибокі експертні статті про поліпшення складських процесів малого підприємства. Автори текстів акуратно та ненав'язливо згадують функціонал розробленого застосунку всередині абзаців. Маркетологи регулярно поширюють ці корисні публікації у профільних групах підприємців. Така публічна активність швидко створює високий рівень довіри до бренду

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	<i>Арк.</i>
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		64

компанії-творця.

Надзвичайно важливою частиною загальної стратегії просування виступає постійна технічна підтримка існуючих активних клієнтів. Робота великого роздрібного складу не може зупинятися через несподівані технічні помилки хмарної програми. Компанія організовує зручний закритий чат для швидких щоденних консультацій користувачів. Клієнти постійно задають питання щодо правильної роботи незрозумілих функцій програми та миттєво отримують кваліфіковані відповіді. Регулярний вихід оновлень з новими корисними модулями надійно утримує клієнтів від болісного переходу до програмних продуктів прямих конкурентів. Власники магазинів чітко бачать постійний розвиток продукту і продовжують справправно оплачувати щомісячну підписку на використання хмарної системи.

					2026.КВР.122.423.09.00.00 ПЗ	Арк.
						65
Зм.	Арк.	№ докум.	Підпис	Дата		

4 ЕКОНОМІЧНИЙ РОЗДІЛ

Метою економічної частини даного дипломного проєкту є проведення економічних розрахунків, спрямованих на визначення економічної ефективності розробки інформаційної системи складського обліку будівельних матеріалів «BudSklad» прийняття рішення про подальший розвиток і впровадження або ж недоцільність проведення відповідної розробки.

Об'єктом розробки є інформаційна система «BudSklad».

Розрахунок вартості розробки виконується в декілька етапів:

- описати технологічний процес розробки із зазначенням трудомісткості кожної операції;
- визначити суму витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
- обчислити витрати на електроенергію;
- нарахувати суму амортизаційних відрахувань;
- визначити суму накладних витрат;
- скласти кошторис та визначити собівартість робіт;
- розрахувати ціну робіт;
- визначити економічну ефективність та термін окупності.

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення НДР

В цьому підрозділі розглянемо основні етапи технологічного процесу для розробки «BudSklad» Для визначення загальної тривалості проведення робіт доцільно дані витрат часу по окремих операціях технологічного процесу звести у таблицю 4.1.

					2026.КВР.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

Таблиця 4.1 - Середній час виконання робіт по обслуговуванню та стадії (операції) технологічного процесу

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1	Планування та аналіз	Розробник Бекенд	10
		Дизайнер	5
2	Розробка технічного завдання	Розробник Бекенд	10
3	Дизайн інтерфейсу та структури БД	Дизайнер	25
4	Розробка функціоналу	Розробник Фронтенд	50
		Розробник Бекенд	50
5	Тестування та відладка	Розробник Фронтенд	10
		Розробник Бекенд	10
6	Документування	Розробник Бекенд	5
7	Розгортання та підтримка	Розробник Фронтенд	5
Разом			180

Сумарний час виконання операцій технологічного процесу становить 180 години.

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

У даному підрозділі проводиться аналіз і розрахунок витрат, пов'язаних з оплатою праці та відрахуваннями на соціальні заходи, що необхідні для розробки «BudSkлад»

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та діяльності підприємства.

Основна заробітна плата розраховується за формулою:

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

$$Z_{\text{осн.}} = T_c * K_r \quad (4.1)$$

де: T_c – тарифна ставка, грн. (приймаємо для дизайнера– 200 грн./год, розробника Фронтенд– 250 грн./год., розробника Бекенд – 280 грн./год); K_r – кількість відпрацьованих годин.

Отже, основна заробітна плата для:

Дизайнер	$Z_{\text{осн.}} = 30 * 200 = 6\,000$ грн.
Розробник Фронтенд	$Z_{\text{осн.}} = 250 * 65 = 16\,250$ грн..
Розробник Бекенд	$Z_{\text{осн.}} = 280 * 85 = 23\,800$ грн

Сумарна основна заробітна плата становить

$$Z_{\text{осн.}} = 6\,000 + 16\,250 + 23\,800 = 46\,050 \text{ грн}$$

Додаткова заробітна плата становить 10 – 15 % від суми основної заробітної плати.

$$Z_{\text{дод.}} = Z_{\text{осн.}} * K_{\text{допл.}} \quad (4.2)$$

де: $K_{\text{допл.}}$ – коефіцієнт додаткових виплат працівникам.

Отже додаткова заробітна плата по категоріях працівників становить:

Дизайнер	$Z_{\text{дод.}} = 6\,000 * 0,1 = 600$ грн.
Розробник Фронтенд	$Z_{\text{дод.}} = 16\,250 * 0,1 = 1625$ грн.
Розробник Бекенд	$Z_{\text{дод.}} = 23\,800 * 0,1 = 2380$ грн.

Загальна додаткова заробітна плата становить:

$$Z_{\text{дод.}} = 600 + 1625 + 2380 = 4\,605 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($V_{\text{о.п.}}$) визначаються за формулою:

$$V_{\text{о.п.}} = Z_{\text{осн.}} + Z_{\text{дод.}} \quad (4.3)$$

$$V_{\text{о.п.}} = 46\,050 + 4\,605 = 50\,655 \text{ грн.}$$

Єдиний соціальний внесок (ЄСВ – 22%) визначається за формулою:

$$V_{\text{ЄСВ}} = V_{\text{о.п.}} * 0,22 \quad (4.4)$$

$$V_{\text{ЄСВ}} = 50\,655 * 0,22 = 11\,144 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці наведено у таблиці 4.2.

					2026.КВР.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

Таблиця 4.2 – Зведені розрахунки витрат на оплату праці

№ п / п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	ЄСВ, грн.	Всього витрати на оплату праці, грн. 6 = 3+4+5
		Тарифна ставка, грн.	К- сть годи н	Фактично нарах. зарплати, грн.			
		1	2	3	4	5	6
1	Дизайнер	200	30	6 000	600	1 452	8 052
2	Розробник Фронтенд	250	65	16 250	1 625	3 933	21 808
3	Розробник Бекенд	280	85	23 800	2 380	5 759	31 939
Разом				46 050	4 605	11 144	61 799

Отже, загальні витрати на оплату праці становлять 61 799 грн.

4.3 Розрахунок витрат на електроенергію

Розрахуємо вартість електроенергії. Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_{\text{в}} = W * T * S \quad (4.5)$$

де: W – необхідна потужність, кВт; T – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії (приймаємо 15,94 грн).

В нашій системі є 1 ПК. Витрати на електроенергію для цього комп'ютера обчислимо окремо, взявши за основу, що час роботи обладнання обчислюється в залежності від виконуваних робіт (згідно табл. 4.1) і споживані

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

потужності наступні: комп'ютер – 0,6 кВт/год.

$$Зек = 0,6 * 180 * 15,94 = 1\,722 \text{ грн.}$$

Витрати на електроенергію становлять 1 722 грн.

4.4 Розрахунок суми амортизаційних відрахувань інформаційної системи

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення

Комп'ютери та оргтехніка належать до четвертої групи основних фондів.

Амортизація на них нараховується лише в випадку, якщо мінімально допустимі строки їх корисного використання 2 роки. Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{Бв * Н_A}{100\%} * T, \quad (4.6)$$

де: А – амортизаційні відрахування за звітний період, грн.; Бв – балансова вартість групи основних фондів на початок звітного періоду, грн.; Н_А – норма амортизації, 0,04 %.

Оскільки для написання програми та її тестування використовується один ПК, вартістю 50000,00 грн., то сума амортизаційних відрахувань становитиме:

$$A = \frac{40\,000,00 * 0,04}{150} * 180 = 1\,920 \text{ грн.}$$

Сума амортизаційних відрахувань становитиме 1 920 грн.

					<i>2026.КВР.122.423.09.00.00 ПЗ</i>	Арк.
						70
Зм.	Арк.	№ докум.	Підпис	Дата		

4.5 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці. В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20–60 % від суми основної та додаткової заробітної плати працівників.

$$H_B = V_{o.p.} * 0,2..0,6 \quad (4.7)$$

де: H_B – накладні витрати.

$$H_B = 50\,655 * 0,4 = 20\,262 \text{ грн.}$$

4.6 Складання кошторису витрат та визначення собівартості інформаційної системи

Для складання кошторису витрат та визначення собівартості, результати проведених вище розрахунків зведемо у таблиці 4.4.

Таблиця 4.4 - Кошторис витрат «BudSklad»

№	Зміст витрат	Сума, грн.	В % до загальної суми
1.	Витрати на оплату праці	61 799	72
2.	Витрати на електроенергію	1 722	2
3.	Амортизаційні відрахування	1 920	2
4.	Накладні витрати	20 262	24
5.	Собівартість	85 703	100

Собівартість (C_B) НДР розрахуємо за формулою:

$$C_B = V_{o.p.} + V_{c.z} + 3e + A + H_B \quad (4.8)$$

Отже, собівартість дорівнює $C_B = 85\,703$ грн.

4.7 Розрахунок ціни інформаційної системи

Розрахунок ціни науково-дослідної роботи включає в себе урахування різноманітних факторів, таких як рівень рентабельності, собівартість та податкова ставка.

Ціну робіт можна визначити за формулою:

$$Ц = C_{в} * (1 + P_{рен}) * (1 + ПДВ), \quad (4.9)$$

де: $C_{в}$ – собівартість; $P_{рен}$ – рівень рентабельності; ПДВ – ставка податку на додану вартість.

$$Ц = 85\,703 * (1 + 0,3) * (1 + 0,2) = 133\,697 \text{ грн.}$$

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Для визначення ефективності продукту розраховують чисту теперішню вартість (ЧТВ) і термін окупності (Ток).

$$ЧТВ = -C_{в} + \sum_{i=1}^t \frac{\Gamma_{п}}{(1+i)^t}, \quad (4.11)$$

де: $C_{в}$ – собівартість розробки; $\Gamma_{п}$ – грошовий потік за t – ий рік; t – відповідний рік проекту; i – величина дисконтної ставки (10...15%).

$$ЧТВ = -85\,703 + \frac{47994}{(1+0,1)^1} + \frac{47994}{(1+0,1)^2} + \frac{47994}{(1+0,1)^3} = 33\,676 \text{ грн}$$

Якщо $ЧТВ \geq 0$, то проект може бути рекомендований до впровадження.

Термін окупності визначається за формулою:

$$T_{ок} = T_{пв} + \frac{H_{в}}{\Gamma_{пр}} \quad (4.12)$$

де: $T_{пв}$ – період до повного відшкодування витрат, років; $H_{в}$ – невідшкодовані витрати на початок року, грн.; $\Gamma_{пр}$ – грошовий потік на початок

					2026.КВР.122.423.09.00.00 ПЗ	Арк.
						72
Зм.	Арк.	№ докум.	Підпис	Дата		

року, грн.

$$T_{\text{ок}} = 2 + \frac{2\,409}{47\,994} = 2,05 \text{ р.}$$

Всі дані внесемо в зведену таблицю 4.5.

Таблиця 4.5 – Техніко-економічні показники «BudSklad»

№ п/п	Показник	Значення
1.	Собівартість, грн.	85 703
2.	Плановий прибуток або грошовий потік, грн.	47 994
3.	Ціна, грн.	133 697
4.	Чиста теперішня вартість, грн.	33 676
5.	Термін окупності, рік	2,05

Прибутковість проекту та термін окупності свідчать про його фінансову ефективність та здатність повернути капітальні вкладення протягом ...року. Отже, на основі отриманих показників можна зробити висновок, що розробка «BudSklad» є доцільною з економічної точки зору.

5 ОХОРОНА ПРАЦІ, ТЕХНІКА БЕЗПЕКИ ТА ЕКОЛОГІЧНІ ВИМОГИ

5.1 Методи боротьби зі шкідливими речовинами на виробництві

Сучасні промислові підприємства часто використовують у технологічних процесах хімічні сполуки, які за певних умов можуть чинити негативний вплив на здоров'я працюючих. Шкідливі речовини здатні проникати в організм людини через органи дихання, шлунково-кишковий тракт або шкірні покриви, викликаючи гострі чи хронічні професійні отруєння та захворювання [13].

Для забезпечення безпечних умов життєдіяльності працівників на виробництві впроваджують комплекс інженерно-технічних, санітарно-гігієнічних та організаційних заходів. Головною метою цих заходів є повне усунення шкідливих факторів або зниження їх концентрації в повітрі робочої зони до рівнів, які не перевищують встановлені нормативи [14].

Основними методами зниження концентрації шкідливих речовин та захисту персоналу є такі заходи:

- Технологічні заходи передбачають повну або часткову заміну токсичних вихідних матеріалів на безпечні чи менш шкідливі аналоги;
- Герметизація обладнання та повна автоматизація процесів мінімізують безпосередній контакт працюючих із небезпечними речовинами;
- Місцева та загальнообмінна вентиляція забезпечує своєчасне видалення забрудненого повітря з робочих приміщень;
- Засоби індивідуального захисту застосовуються у випадках, коли технічні методи не дозволяють повністю очистити повітряне середовище;

Контроль за вмістом небезпечних домішок у повітрі здійснюється шляхом порівняння фактичних показників із гранично допустимими концентраціями. Усі хімічні речовини розподіляються за ступенем небезпеки на відповідні категорії.

Основні класи небезпеки шкідливих речовин та приклади їх представників наведені у таблиці 5.1.

					2026.КВР.122.423.09.00.00 ПЗ	Арк.
						74
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 5.1 - Класи небезпеки шкідливих речовин

Клас небезпеки	Ступінь небезпеки речовини	Приклади хімічних сполук
1	Надзвичайно небезпечні	Ртуть, свинець, озон
2	Високонебезпечні	Кислоти, луги, хлор
3	Помірно небезпечні	Спирти, ацетон, фенол
4	Малонебезпечні	Бензин, аміак, гас

Правильний вибір технологічних та санітарно-гігієнічних методів дозволяє суттєво зменшити ризик виникнення хронічних патологій у працівників підприємства. Оцінка ефективності впроваджених захисних систем має проводитися регулярно за допомогою лабораторно-інструментальних вимірювань хімічних факторів виробничого середовища .

5.2 Види виробничого освітлення

Раціонально організоване освітлення робочих місць є одним із ключових елементів охорони праці на будь-кому підприємстві. Оптимальні світлові умови забезпечують зниження зорової та загальної втоми, підвищують продуктивність праці та суттєво зменшують ризик виникнення травматизму .

Залежно від джерела світлової енергії, що використовується у приміщеннях, виробниче освітлення поділяють на три основні види [15].

Класифікація освітлення за типом джерела світла включає такі варіанти:

- Природне освітлення забезпечується безпосереднім сонячним світлом або дифузним світлом від небосхилу через віконні прорізи;
- Штучне освітлення створюється за допомогою електричних ламп у приміщеннях, де недостатньо природного світла або під час роботи вночі. Штучне освітлення є найбільш керованим та функціональним, тому його додатково поділяють на кілька видів залежно від призначення .

Основними видами штучного освітлення за функціональним

призначенням є такі системи:

- Робоче освітлення призначене для забезпечення нормального виконання виробничого процесу, проходу людей та руху транспорту;
- Аварійне освітлення використовується для продовження роботи в разі раптового відключення основного джерела живлення;
- Евакуаційне освітлення забезпечує безпечний вихід працівників із приміщень під час виникнення надзвичайних ситуацій;
- Охоронне освітлення встановлюється вздовж меж території підприємства для візуального контролю в нічний час;
- Сигнальне освітлення застосовується для фіксації небезпечних зон або вказує безпечні напрямки руху евакуації.

Кількісні та якісні показники штучного освітлення нормуються відповідно до розміру об'єктів розрізнення та характеру зорової роботи.

Параметри мінімальної освітленості для різних категорій точності робіт наведені у таблиці 5.2.

Таблиця 5.2 - Норми штучного освітлення для різних умов праці

Характеристика зорової роботи	Розмір об'єкта розрізнення, мм	Мінімальна освітленість, лк
Найвищої точності	Менше 0.15	1250
Дуже високої точності	Від 0.15 до 0.30	500
Високої точності	Від 0.30 до 0.50	300
Середньої точності	Від 0.50 до 1.00	200

Дотримання наведених нормативних показників штучної освітленості на робочих місцях є обов'язковим для попередження розвитку професійної міопії та інших порушень зорового апарату. Проектування освітлювальних установок повинно здійснюватися з урахуванням типу ламп, характеристик світильників та особливостей виконання конкретних технологічних операцій.

ВИСНОВКИ

Під час виконання випускної кваліфікаційної роботи було успішно проведено комплексний цикл проектування та технічної розробки системи для автоматизації поточної діяльності будівельного складу. Через виражену специфіку обраної галузі вдалося врахувати критичні потреби роздрібної торгівлі та складського обліку. В основу технічної архітектури розробленого рішення покладено сучасний стек інструментів на базі середовища Node.js, фреймворку React та мови програмування TypeScript. З допомогою обраних засобів вдалося значно підвищити стабільність коду та мінімізувати кількість логічних помилок.

Завдяки впровадженню рольової моделі доступу було чітко розмежовано права касирів, менеджерів та адміністраторів для гарантування високого рівня внутрішньої безпеки всієї корпоративної інформації. Для надійного захисту бази даних під час масових запитів від користувачів впроваджено механізм розподілених транзакцій безпосередньо на рівні серверної частини коду. У такий спосіб було виключено ризик подвійного списання залишків товарів при одночасній роботі кількох касових вузлів.

У процесі верстки користувацького інтерфейсу значну увагу було сконцентровано на зручності модуля касира та швидкості навігації. Шляхом прямої інтеграції зі сканерами штрих-кодів забезпечено високу пропускну здатність торгових точок у години пікового навантаження. З метою суттєвого покращення внутрішньої логістики було створено окрему систему миттєвих сповіщень про зміну поточних статусів оформлених замовлень на продаж товарів.

У межах адміністративної панелі реалізовано окремі блоки для глибокого моніторингу запасів та формування фінансової звітності. На фінальному етапі роботи здійснено повний перенос коду на віддалений сервер. Отримані результати комплексного тестування довели повну готовність продукту до експлуатації в реальних умовах будівельного бізнесу.

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
						77
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ПОСИЛАНЬ

- 1) Richards G. Warehouse Management: A Complete Guide to Improving Efficiency and Minimizing Costs in the Modern Warehouse. London : Kogan Page, 2021. 504 p.
- 2) Ткачук М.В. Теорія та практика проектування складних програмних систем : навч. посіб. Харків : НТУ «ХПІ», 2021. 256 с.
- 3) Schwarzmüller M. React Key Concepts: Consolidate your knowledge of React's core features. Birmingham : Packt Publishing, 2022. 402 p.
- 4) Tailwind CSS Documentation : вебсайт. URL: <https://tailwindcss.com/docs> (дата звернення: 20.06.2026).
- 5) Riggs S., Ciolli G. PostgreSQL 16 Administration Cookbook. Birmingham : Packt Publishing, 2024. 490 p.
- 6) Prisma Documentation : вебсайт. URL: <https://www.prisma.io/docs> (дата звернення: 20.06.2026).
- 7) React Documentation : вебсайт. URL: <https://react.dev> (дата звернення: 20.06.2026).
- 8) Node.js v20.x Documentation : вебсайт. URL: <https://nodejs.org/docs/latest-v20.x/api/> (дата звернення: 20.06.2026).
- 9) Herron D. Node.js Web Development: Client-side development with React, Node.js, Express, and MongoDB. Birmingham : Packt Publishing, 2023. 712 p.
- 10) Supabase Documentation : вебсайт. URL: <https://supabase.com/docs> (дата звернення: 20.06.2026).
- 11) Render Documentation : вебсайт. URL: <https://render.com/docs> (дата звернення: 20.06.2026).
- 12) Vercel Documentation : вебсайт. URL: <https://vercel.com/docs> (дата звернення: 20.06.2026).
- 13) Жидецький В.Ц. Основи охорони праці. Львів : Афіша, 2005. 349 с.
- 14) Гандзюк М.П., Желібо Є.П., Халімовський М.О. Охорона праці. К. : Каравела, 2004. 408 с.

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		78

15) Державна служба України з питань праці : вебсайт. URL:
<https://dsp.gov.ua> (дата звернення: 13.06.2026).

					2026.КВР.122.423.09.00.00 ПЗ	Арк.
						79
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

Додаток А. Лістинг файлу «LoginPage.tsx»

```
import { useState } from 'react';
import { useNavigate } from 'react-router-dom';
import api from '../api/axios';
import { Warehouse, LogIn } from 'lucide-react';
import { useAuthStore } from '../store/useAuthStore';
export default function LoginPage() {
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [error, setError] = useState('');
  const [loading, setLoading] = useState(false);
  const navigate = useNavigate();
  const setAuth = useAuthStore((state) => state.setAuth);
  const handleSubmit = async (e: React.FormEvent) => {
    e.preventDefault();
    setLoading(true);
    setError('');
    try {
      const response = await api.post('/auth/login', { email, password });
      setAuth(response.data.data.user, response.data.data.token);
      navigate('/');
    } catch (err: any) {
      setError(err.response?.data?.error || 'Помилка при вході. Перевірте дані.');
```

```

    <div className="inline-flex items-center justify-center w-16 h-16
rounded-2xl bg-indigo-600 text-white shadow-xl shadow-indigo-200 mb-6">
      <Warehouse size={32} />
    </div>
    <h2 className="text-3xl font-black text-slate-900 tracking-
tight">Вітаємо в BudSklad</h2>
    <p className="text-slate-500 mt-2 font-medium uppercase tracking-
widest text-[10px]">Система управління торгівлею</p>
  </div>
  {error && (
    <div className="bg-rose-50 border border-rose-100 text-rose-600 p-
4 rounded-xl text-sm font-bold animate-fade-in flex items-center space-x-
2">
      <span className="w-1.5 h-1.5 rounded-full bg-rose-600 mr-
1"></span>
      {error}
    </div>
  )}
  <form onSubmit={handleSubmit} className="space-y-5">
    <div>
      <label className="block text-xs font-black text-slate-400
uppercase tracking-widest mb-1.5 ml-1">Email адреса</label>
      <input
        type="email"
        required
        placeholder="admin@budsklad.com"
        className="w-full bg-slate-50 border border-slate-200 rounded-
xl p-3.5 focus:ring-2 focus:ring-indigo-500 focus:border-transparent
outline-none transition-all font-medium"
        value={email}
        onChange={(e) => setEmail(e.target.value)}
      />
    </div>
    <div>

```

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		81

```

    <label className="block text-xs font-black text-slate-400
uppercase tracking-widest mb-1.5 ml-1">Пароль</label>
    <input
      type="password"
      required
      placeholder="••••••••"
      className="w-full bg-slate-50 border border-slate-200 rounded-
xl p-3.5 focus:ring-2 focus:ring-indigo-500 focus:border-transparent
outline-none transition-all font-medium"
      value={password}
      onChange={(e) => setPassword(e.target.value)}
    />
  </div>
  <button
    type="submit"
    disabled={loading}
    className="w-full flex justify-center items-center space-x-2 py-
4 px-4 bg-indigo-600 hover:bg-indigo-700 text-white rounded-xl font-bold
shadow-lg shadow-indigo-200 transition-all active:scale-[0.98]
disabled:opacity-70 disabled:cursor-not-allowed"
    > {loading ? (
      <div className="w-5 h-5 border-2 border-white/30 border-t-
white rounded-full animate-spin"></div>
    ) : (
      <>
        <LogIn size={20} />
        <span>Увійти в кабінет</span>
      </> )}
  </button> </form>
  <div className="pt-4 text-center">
    <p className="text-[10px] font-bold text-slate-400 uppercase
tracking-widest">
      Якщо ви забули пароль, зверніться до адміністратора
    </p> </div> </div>);}

```

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		82


```

slate-100 shadow-sm flex items-center space-x-4">
    <div className={` ${stat.color} p-4 rounded-xl text-white
shadow-lg`} >
        <stat.icon size={24} />
    </div>
    <div>
        <p className="text-sm font-bold text-slate-500 uppercase
tracking-wider">{stat.label}</p>
        <p className="text-2xl font-black text-slate-900 mt-
0.5">{stat.value}</p>
    </div>
</div>
    )})
</div>

    <div className="bg-white rounded-2xl border border-slate-100 shadow-
sm p-8">
        <h2 className="text-xl font-bold text-slate-900 mb-6 flex items-
center">
            <TrendingUp className="mr-2 text-indigo-500" /> Динаміка
продажів
        </h2>
        <div className="h-64 flex items-center justify-center bg-slate-50
rounded-xl border border-dashed border-slate-200">
            <p className="text-slate-400 font-medium">Графік буде доступний
після накопичення даних</p>
        </div>
    </div>
</div>
);
}

```

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		84


```

    setIsSuccessModalOpen(true);
    queryClient.invalidateQueries({ queryKey: ['activeShift'] });
  },
  onError: (err: any) => {
    alert(err.response?.data?.error || 'Помилка при проведенні чека');
  }
});

const addToCart = (product: Product) => {
  const existing = cart.find(i => i.product.id === product.id);
  if (existing) {
    setCart(cart.map(i => i.product.id === product.id ? { ...i,
quantity: i.quantity + 1 } : i));
  } else {
    setCart([...cart, { product, quantity: 1 }]);
  }
  setSearchQuery('');
};

const updateQuantity = (productId: number, quantity: number) => {
  if (quantity <= 0) {
    setCart(cart.filter(i => i.product.id !== productId));
  } else {
    setCart(cart.map(i => i.product.id === productId ? { ...i, quantity
} : i));
  }
};

const total = cart.reduce((sum, i) => sum + i.product.price *
i.quantity, 0);

const handleCheckout = () => {
  if (!activeShift) return;
  if (cart.length === 0) return;
  receiptMutation.mutate({
    shiftId: activeShift.id,
    paymentType,
    items: cart.map(i => ({ productId: i.product.id, quantity:

```

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		86

```

i.quantity })),
  });
};
if (isShiftLoading) return <div className="p-20 text-center text-slate-400">Завантаження...</div>;
if (!activeShift) {
  return (
    <div className="flex-grow flex items-center justify-center bg-slate-900">
      <div className="bg-slate-800 p-10 rounded-3xl border border-slate-700 shadow-2xl text-center max-w-md w-full">
        <div className="w-20 h-20 bg-emerald-500/10 text-emerald-500 rounded-full flex items-center justify-center mx-auto mb-6">
          <Monitor size={40} />
        </div>
        <h2 className="text-2xl font-black text-white mb-2 uppercase tracking-tight">Зміна закрита</h2>
        <p className="text-slate-400 mb-8">Для початку роботи необхідно відкрити касову зміну та вказати початкову суму в касі.</p>
        <button
          onClick={() => window.location.href = '/pos/shift'}
          className="w-full py-4 bg-emerald-500 text-white rounded-2xl font-black uppercase tracking-widest hover:bg-emerald-600 transition-all shadow-lg shadow-emerald-500/20"
        >
          Перейти до управління зміною
        </button>
      </div>
    </div>
  );
}
return (
  <div className="flex-grow flex h-full overflow-hidden">
    { /* Left side: Search and Products */ }

```

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		87

```

<div className="flex-grow p-6 flex flex-col space-y-6 overflow-
hidden">
  <div className="flex space-x-4">
    <div className="relative flex-grow">
      <Search className="absolute left-4 top-1/2 -translate-y-1/2
text-slate-500" size={24} />
      <input
        type="text"
        placeholder="Пошук товару (F2)..."
        className="w-full pl-14 pr-4 py-4 bg-slate-800 border
border-slate-700 rounded-2xl text-xl text-white outline-none focus:ring-2
focus:ring-emerald-500/50 transition-all"
        value={searchQuery}
        onChange={(e) => setSearchQuery(e.target.value)}
        autoFocus
      />
      {searchResults && searchQuery.length > 2 && (
        <div className="absolute top-full left-0 right-0 mt-2 bg-
slate-800 border border-slate-700 rounded-2xl shadow-2xl z-20 overflow-
hidden">
          {searchResults.map(p => (
            <button
              key={p.id}
              onClick={() => addToCart(p)}
              className="w-full flex items-center justify-between p-
5 hover:bg-slate-700 border-b border-slate-700 last:border-0"
            >
              <div className="text-left">
                <p className="font-bold text-white text-
lg">{p.name}</p>
                <p className="text-sm text-slate-400">SKU: {p.sku} |
Залишок: {p.stockQuantity}</p>
              </div>
              <div className="text-right">

```

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		88

```

                <p className="font-black text-emerald-400 text-
xl">₺{p.price}</p>
            </div>
        </button>
    )})
</div>
})
</div>
<button
    onClick={() => setIsScannerOpen(true)}
    className="p-4 bg-slate-800 border border-slate-700 rounded-
2xl text-slate-400 hover:text-emerald-400 hover:bg-slate-700 transition-
all"
    > <Scan size={32} />
</button>
</div>
<div className="flex-grow bg-slate-800/50 rounded-3xl border
border-slate-700 overflow-hidden flex flex-col">
    <div className="p-6 border-b border-slate-700 bg-slate-800/80
flex justify-between items-center">
        <h3 className="font-bold uppercase tracking-widest text-slate-
400 text-sm">Поточний чек</h3>
        <span className="text-xs font-black bg-slate-700 px-3 py-1
rounded-full text-slate-300">
            {cart.length} позицій
        </span>
    </div>
    <div className="flex-grow overflow-y-auto p-6 space-y-4">
        {cart.map(item => (
            <div key={item.product.id} className="flex items-center
justify-between p-4 bg-slate-800 border border-slate-700 rounded-2xl">
                <div className="flex-grow">
                    <p className="font-bold text-white text-
lg">{item.product.name}</p>

```

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		89


```

hover:border-slate-600"
    })
  >
    <CreditCard size={32} />
    <span className="font-black uppercase tracking-widest
text-xs">Картка</span>
  </button>
</div>
</div>
<div className="bg-slate-900 p-6 rounded-2xl border border-
slate-700 space-y-4">
  <div className="flex justify-between text-slate-400 font-
bold">
    <span>Проміжний підсумок:</span>
    <span>₹{total.toFixed(2)}</span>
  </div>
  <div className="flex justify-between text-slate-400 font-
bold">
    <span>Знижка:</span>
    <span>₹0.00</span>
  </div>
  <div className="pt-4 border-t border-slate-700 flex justify-
between items-end">
    <span className="text-lg font-black text-white uppercase
tracking-tighter">Разом</span>
    <span className="text-5xl font-black text-emerald-400
tracking-tighter">₹{total.toFixed(2)}</span>
  </div>
</div>
</div>
<button
  onClick={handleCheckout}
  disabled={cart.length === 0 || receiptMutation.isPending}
  className="w-full py-6 bg-emerald-500 text-white rounded-2xl

```

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		92

```

font-black text-2xl uppercase tracking-widest hover:bg-emerald-600
disabled:opacity-50 disabled:grayscale transition-all shadow-xl shadow-
emerald-500/20 mt-8" >
    {receiptMutation.isPending ? 'Проведення...' : 'Оплатити (F12)'}
</button> </div>
<BarcodeScanner isOpen={isScannerOpen} onScan={(code) => {
    productApi.getByBarcode(code).then(res => {
        if (res.data.data) addToCart(res.data.data);
    }).catch(() => alert('Товар не знайдено'));
    setIsScannerOpen(false); }} />
{/* Success Modal */}
{isSuccessModalOpen && (
    <div className="fixed inset-0 bg-slate-950/80 backdrop-blur-md z-
[100] flex items-center justify-center p-4">
        <div className="bg-slate-800 p-10 rounded-3xl border border-
slate-700 shadow-2xl text-center max-w-sm w-full animate-in zoom-in-95
duration-200">
            <div className="w-24 h-24 bg-emerald-500 text-white rounded-
full flex items-center justify-center mx-auto mb-6 shadow-lg shadow-
emerald-500/20"> <CheckCircle size={56} />
            </div>
            <h2 className="text-3xl font-black text-white mb-2 uppercase
tracking-tight">Готово!</h2>
            <p className="text-slate-400 mb-8 font-medium">Чек успішно
проведено та збережено в системі.</p>
            <button
                onClick={() => setIsSuccessModalOpen(false)}
                className="w-full py-4 bg-slate-700 text-white rounded-2xl
font-black uppercase tracking-widest hover:bg-slate-600 transition-all"
                >Наступний покупець </button> </div> </div>
        )} </div> );
}

```

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		93

Додаток Г. Лістинг файлу «useCartStore.ts»

```
import { create } from 'zustand';
import { persist } from 'zustand/middleware';

interface CartItem {
  id: number;
  name: string;
  sku: string;
  quantity: number;
  unit: string;
}

interface CartState {
  items: CartItem[];
  addItem: (item: CartItem) => void;
  removeItem: (id: number) => void;
  updateQuantity: (id: number, quantity: number) => void;
  clearCart: () => void;
}

export const useCartStore = create<CartState>()(
  persist(
    (set) => ({
      items: [],
      addItem: (newItem) => set((state) => {
        const existingItem = state.items.find(i => i.id === newItem.id);
        if (existingItem) {
          return {
            items: state.items.map(i =>
              i.id === newItem.id
                ? { ...i, quantity: i.quantity + newItem.quantity }
                : i
            )
          }
        }
      })
    })
  )
)
```

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		94

```

    };
  }
  return { items: [...state.items, newItem] };
}),
removeItem: (id) => set((state) => ({
  items: state.items.filter(i => i.id !== id)
})),
updateQuantity: (id, quantity) => set((state) => ({
  items: state.items.map(i => i.id === id ? { ...i, quantity } : i)
})),
clearCart: () => set({ items: [] }),
}),
{
  name: 'budsklad-cart',
}
)
);

```

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		95

Додаток Д. Лістинг файлу «auth.middleware.ts»

```
import { Request, Response, NextFunction } from 'express';
import jwt from 'jsonwebtoken';
import { sendError } from '../utils/response';
const JWT_SECRET = process.env.JWT_SECRET || 'fallback-secret';
export interface AuthRequest extends Request {
  user?: {
    userId: number;
    role: string;
  };
}
export const authenticate = (req: AuthRequest, res: Response, next:
NextFunction) => {
  const authHeader = req.headers.authorization;

  if (!authHeader || !authHeader.startsWith('Bearer ')) {
    return sendError(res, 'Авторизація обов'язкова', 401);}
  const token = authHeader.split(' ')[1];
  if (!token) {
    return sendError(res, 'Токен відсутній', 401);
  } try {
    const decoded = jwt.verify(token, JWT_SECRET) as any;
    req.user = {
      userId: decoded.userId,
      role: decoded.role,    };
    next();
  } catch (error) {
    return sendError(res, 'Недійсний токен', 401); }};
export const authorize = (roles: string[]) => {
  return (req: AuthRequest, res: Response, next: NextFunction) => {
    if (!req.user || !roles.includes(req.user.role)) {
      return sendError(res, 'Доступ заборонено', 403);
    }    next();  }};
```

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		96

Додаток Е. Лістинг файлу «pos.controller.ts»

```
import { Response, NextFunction } from 'express';
import { POSService } from '../services/pos.service';
import { openShiftSchema, closeShiftSchema, createReceiptSchema } from
'../schemas/pos.schema';
import { AuthRequest } from '../middlewares/auth.middleware';
import { sendSuccess, ApiError } from '../utils/response';

export class POSController {
  static async getActiveShift(req: AuthRequest, res: Response, next:
NextFunction) {
    try {
      const shift = await POSService.getActiveShift(req.user!.userId);
      return sendSuccess(res, shift);
    } catch (error) {
      next(error);
    }
  }

  static async openShift(req: AuthRequest, res: Response, next:
NextFunction) {
    try {
      const validatedData = openShiftSchema.parse(req.body);
      const shift = await POSService.openShift(req.user!.userId,
validatedData);
      return sendSuccess(res, shift, 'Зміну відкрито', 201);
    } catch (error) {
      next(error);
    }
  }

  static async closeShift(req: AuthRequest, res: Response, next:
NextFunction) {
```

					2026.KBP.122.423.09.00.00 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		97

```

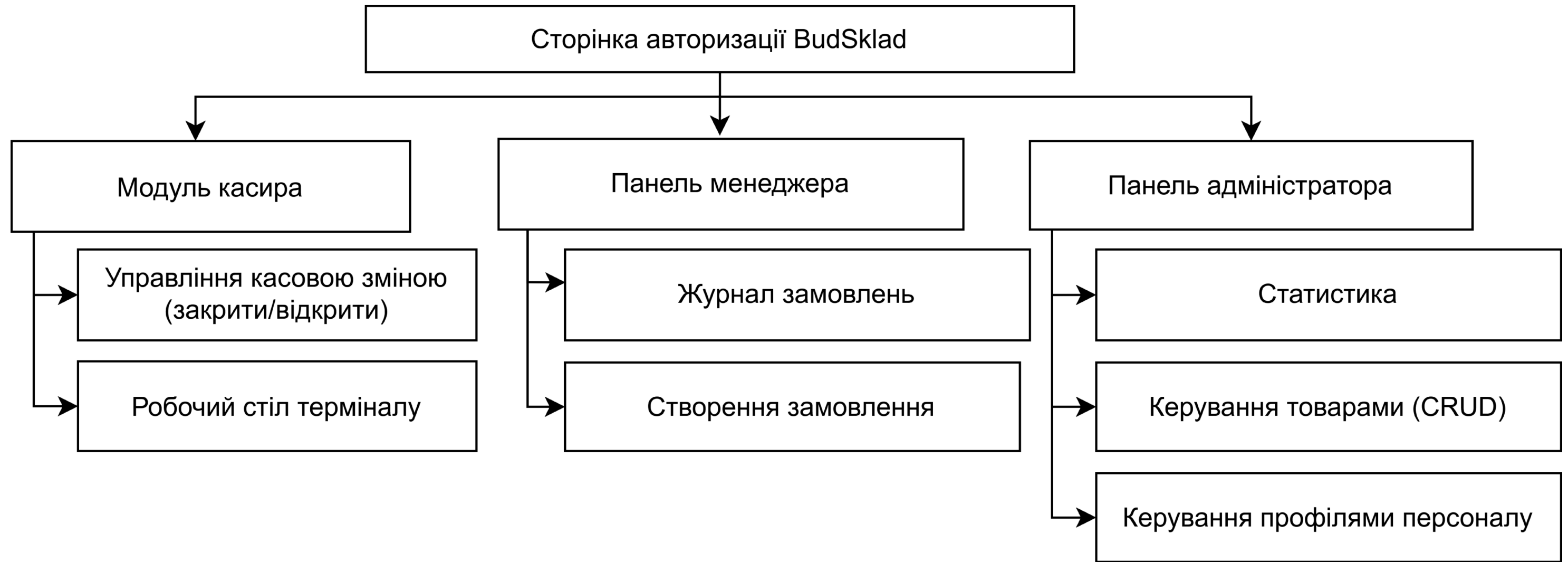
    try {
        const validatedData = closeShiftSchema.parse(req.body);
        const shift = await POSService.closeShift(Number(req.params.id),
validatedData);
        return sendSuccess(res, shift, 'Зміну закрито');
    } catch (error) {
        next(error);
    }
}

static async createReceipt(req: AuthRequest, res: Response, next:
NextFunction) {
    try {
        const validatedData = createReceiptSchema.parse(req.body);
        const receipt = await POSService.createReceipt(validatedData);
        return sendSuccess(res, receipt, 'Чек проведено', 201);
    } catch (error) {
        next(error);
    }
}
}
}

```

					<i>2026.KBP.122.423.09.00.00 ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		98

Схема структурна інформаційної системи



Лист № 1
Лист № 2
Лист № 3
Лист № 4
Лист № 5
Лист № 6
Лист № 7
Лист № 8
Лист № 9
Лист № 10
Лист № 11
Лист № 12
Лист № 13
Лист № 14
Лист № 15
Лист № 16
Лист № 17
Лист № 18
Лист № 19
Лист № 20
Лист № 21
Лист № 22
Лист № 23
Лист № 24
Лист № 25
Лист № 26
Лист № 27
Лист № 28
Лист № 29
Лист № 30
Лист № 31
Лист № 32
Лист № 33
Лист № 34
Лист № 35
Лист № 36
Лист № 37
Лист № 38
Лист № 39
Лист № 40
Лист № 41
Лист № 42
Лист № 43
Лист № 44
Лист № 45
Лист № 46
Лист № 47
Лист № 48
Лист № 49
Лист № 50

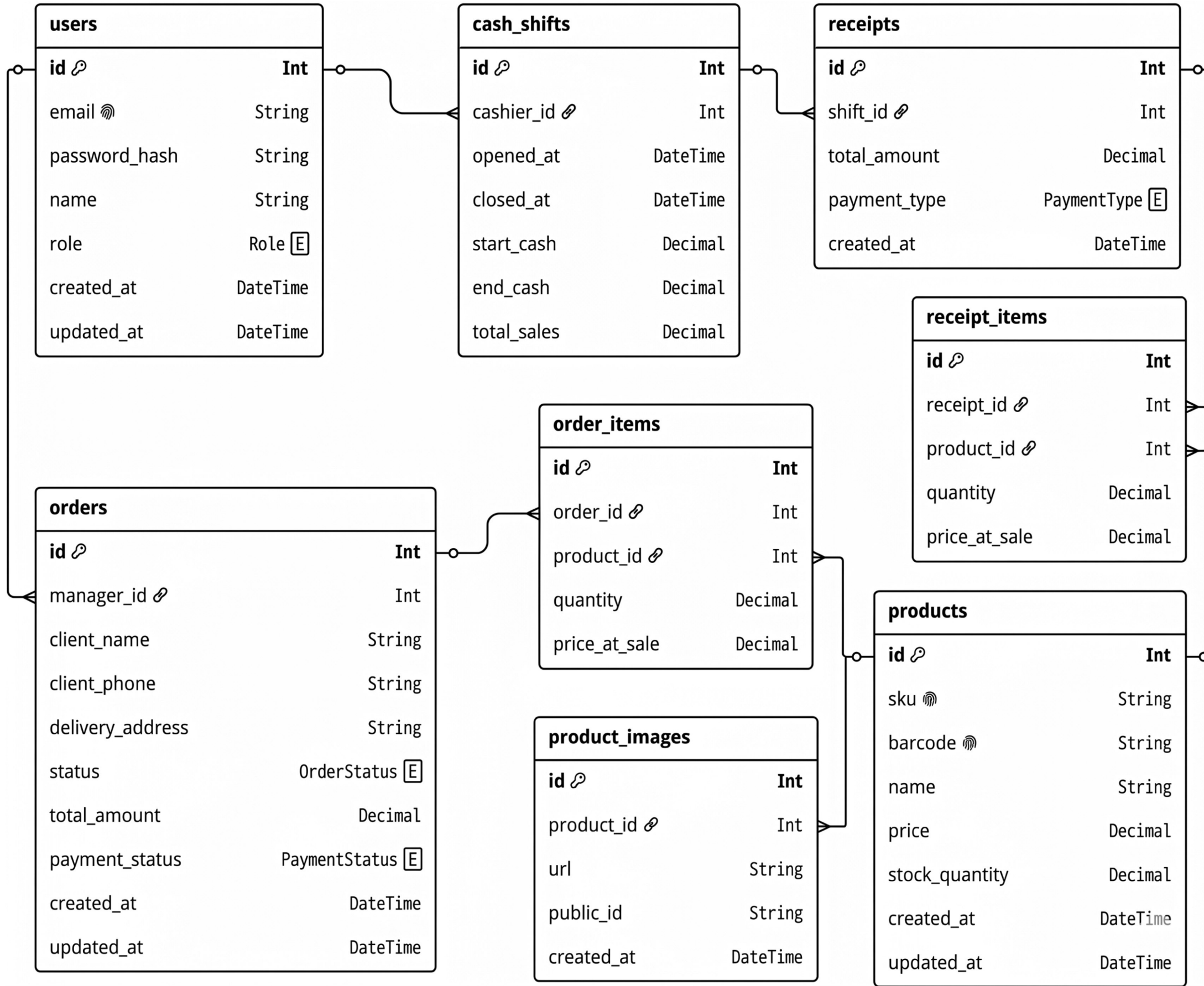
				2026.KBP.122.423.09.00.00 CC				
Изм.	Лист	№ докум.	Подп.	Дата	Разработка информационной системы складского отдела бюджетных материалов «BudSklad»	Лист	Масса	Масштаб
Разработ		Корень М.С.			Схема структурная информационной системы			
Проект		Сердюк В.С.				Лист		Листов 1
Исполнит.								
Рецензент								
Исполнит.		Приймак В.А.						
Чит.								
Копировал						Формат А1		

Таблиця техніко-економічних показників

№	Показник	Одиниця вимірювання	Значення
1	Мова програмування	–	JavaScript (ES6)
2	Фреймворк клієнтської частини	–	React (Vite)
3	Фреймворк серверної частини	–	Express (Node.js)
4	Бібліотека стилізації інтерфейсу	–	Tailwind CSS (v3.4.3)
5	Обсяг програмного коду	рядок	~8 500
6	Кількість таблиць у базі даних	штук	8
7	Содівартість	грн	85 703
8	Плановий грошовий потік	грн	47 994
9	Ціна	грн	133 697
10	Чиста теперішня вартість	грн	33 676
11	Термін окупності	рік	2,05

				2026.KBP.122.423.09.00.00 ТБ			
Изм.	Лист	№ вказ.	Лист	Листа	Розробка інформаційної системи складського обліку відділень матеріалів «ВидSkлад»	Лист	Масштаб
Розроб.		Корень МС			Таблиця техніко-економічних показників		
Проб.		Сердін ВС				Лист	Листа 1
Іконтр.							
Рецензент							
Іконтр.		Приймак В.А.				ВСП ТФК ТНТУ	КН-423
Чиб.						м. Тернопіль	
				Копіював			
				Формат А1			

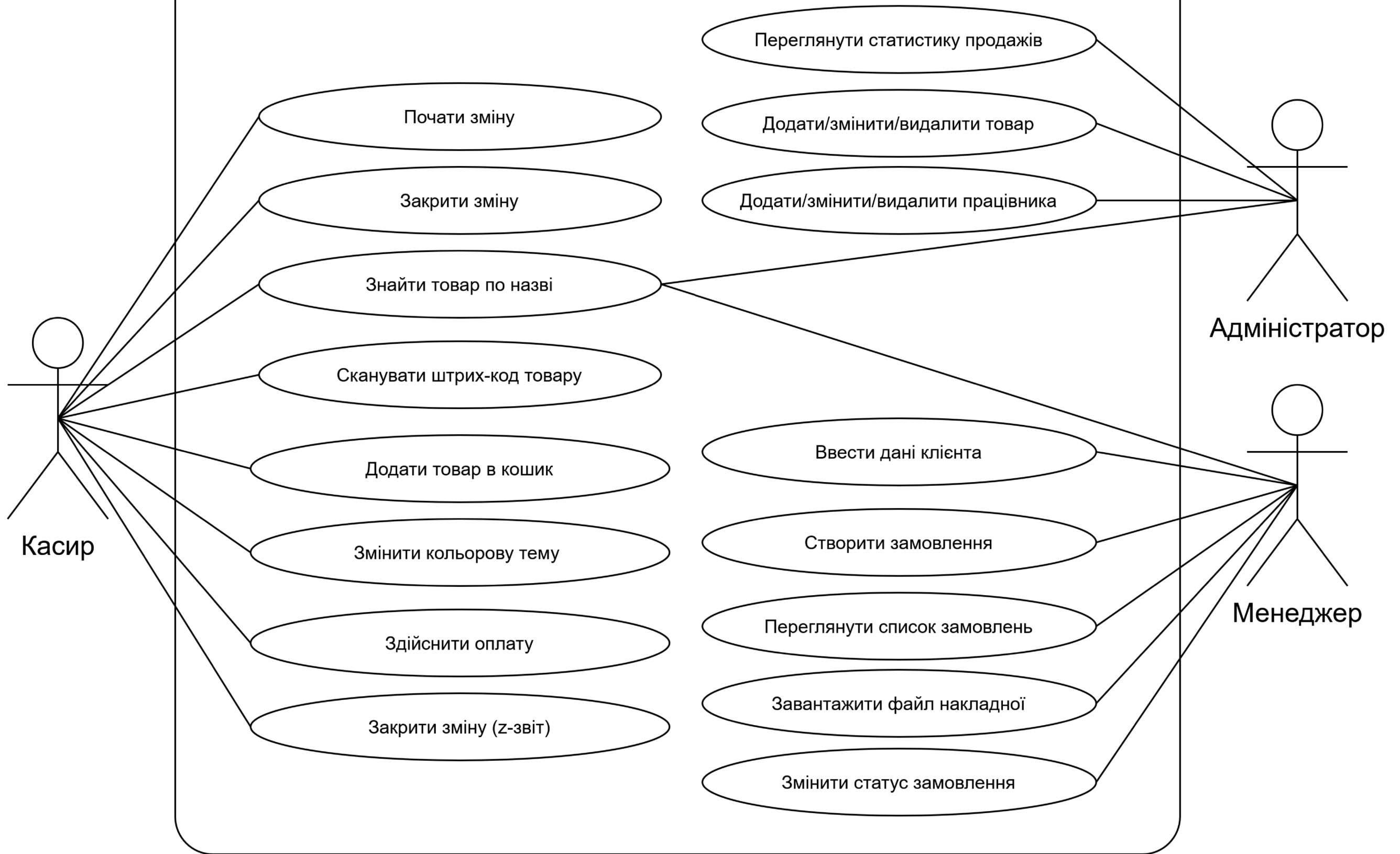
ER-діаграма бази даних



Изм.	Лист	№ докум.	Лист	Дата	Разработчик	Лист	Масса	Масштаб
Разработ		Корень М.С.			Разработка информационной системы складского отдела бюджетных материалов «ВидСклад»			
Проб		Сергей В.С.			ER-диаграмма базы данных	Лист	Листов	1
Исполн								
Рецензент								
Исполн		Проймак В.А.						
Чит								

UML-діаграма варіантів використання

Інформаційна система «BudSklad»



Лист	№	Важк.	Лист	Лист
Розроб.	Корень МС			
Проб.	Сергій В.С.			
Іконтр.				
Рецензент				
Іконтр.	Приймак В.А.			
Члб.				

2026.KBP.122.4.23.09.00.00 ДВ				Лист	Масштаб	Масштаб
Розробка інформаційної системи складського обліку будівельних матеріалів «BudSklad»				Лист	Листов	1
UML-діаграма варіантів використання				Лист	Листов	1
ВСП ТФК ТНТУ				КН-4.23		
м. Тернопіль						
Копіював				Формат		A1