

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій
(повне найменування факультету)

кафедра «Системний аналіз та обчислювальна математика»
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему СИСТЕМНИЙ АНАЛІЗ ТА МОДЕЛЮВАННЯ ВИТРАТ
ВИРОБНИЦТВА МЕТАЛЕВИХ КОНСТРУКЦІЙ

(назва теми)

Виконав: студент 4 курсу,
групи КНТ-812

Спеціальності 124 - Системний аналіз
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інтелектуальні технології та прийняття рішень в складних системах

АЛЕКСЄЄВ Б.А.

(ПРИЗВИЩЕ та ініціали)

Керівник РЯБЕНКО А.Є.

(ПРИЗВИЩЕ та ініціали)

Рецензент ПИРОЖОК А. В

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук та технологій
Кафедра «Системний аналіз та обчислювальна математика»
Ступінь вищої освіти Бакалавр
Спеціальність 124 Системний аналіз

(код і найменування)

Освітня програма (спеціалізація) Інтелектуальні технології та прийняття рішень в складних системах

(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри Системного аналізу та обчислювальної математики

Еліна ТЕРЕЩЕНКО
«23» Квітня 2026 року

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

Алексєєв

Богдан

Андрійович

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Системний аналіз та моделювання витрат виробництва металевих конструкцій.

керівник проєкту (роботи) к.ф.-м.н., доцент Рябенко Антон Євгенович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від « 23 » 04 2026 року №196

2. Строк подання студентом проєкту (роботи) 08.06.2026

3. Вихідні дані до проєкту (роботи) розроблено інформаційну систему підтримки прийняття рішень (СППР) для калькуляції та моделювання витрат на виробництво металоконструкцій, орієнтовану на інженерно-технічний персонал. Інструментальні засоби розробки: мова програмування Python, фреймворк Django, СКБД SQLite, технології HTMX та Chart.js.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області та постановка задачі моделювання виробничих витрат. 2. Математичне моделювання та проєктування інформаційної системи. 3. Програмна реалізація та тестування вебзастосунку.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) 14 слайдів презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконання завдання
1	к.ф.-м.н., доцент Рябенко А. Є.	25.04.26	12.05.26
2	к.ф.-м.н., доцент Рябенко А. Є.	13.05.26	26.05.26
3	к.ф.-м.н., доцент Рябенко А. Є.	27.05.26	05.06.26
Нормоконтроль	Доцент Широкоград Д. В.	25.04.26	08.06.26

7. Дата видачі завдання «25» Квітня 2026 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Системний аналіз предметної області та існуючих ІС калькуляції витрат	1 тиждень	Аналітичний огляд
2	Формулювання функціональних і нефункціональних вимог до СППР	2 тиждень	Технічне завдання
3	Структурна декомпозиція та розробка математичної моделі калькуляції	3 тиждень	Математична модель (7 блоків)
4	Проектування архітектури застосунку (MTV) та реляційної бази даних	4 тиждень	Схема БД та архітектура
5	Програмна реалізація серверної логіки (Python/Django) та інтерфейсу (HTMX)	5 тиждень	Вихідний код системи
6	Тестування План-Факт аналізу на змодельованих виробничих сценаріях	6 тиждень	Працевдатна СППР
7	Оформлення розділів пояснювальної записки та економічне обґрунтування	6 тиждень	Пояснювальна записка (ПЗ)
8	Підготовка графічного матеріалу та презентації до захисту	7 тиждень	Слайди презентації
9	Захист кваліфікаційної роботи	8 тиждень	

Студент(ка)


(підпис)

АЛЕКСЄЄВ Б.А.

(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис)

РЯБЕНКО А.Є.

(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка: 75 с., 9 рис., 0 табл., 26 формул, 2 додатки, 22 джерела.

Об'єкт дослідження – процес управління та калькуляції витрат на виробництво металевих конструкцій.

Предмет дослідження – методи, моделі та інформаційні технології автоматизованого моделювання та План-Факт аналізу виробничої собівартості.

Мета роботи – підвищення ефективності управління фінансово-технологічними процесами при виготовленні металевих конструкцій шляхом розробки та впровадження спеціалізованої інформаційної системи підтримки прийняття рішень (СППР).

Методи дослідження: системний аналіз, структурна декомпозиція, процесно-орієнтоване моделювання, реляційна алгебра, об'єктно-орієнтоване проектування.

Розроблено 7-блокову математичну модель калькуляції витрат, яка розмежовує трудомісткість і календарний час, а також формалізує облік ланцюгових витрат при фіксації браку. Спроектовано реляційну БД та розроблено вебзастосунок (Python, Django, SQLite, HTMX, Chart.js). Тестування підтвердило працездатність СППР: її впровадження скорочує час обробки замовлень на 70–80%, мінімізує вплив людського фактора та забезпечує оперативний моніторинг рентабельності без витрат на комерційні ліцензії.

КЛЮЧОВІ СЛОВА: СИСТЕМНИЙ АНАЛІЗ, МОДЕЛЮВАННЯ ВИТРАТ, ПЛАН-ФАКТ АНАЛІЗ, СППР, МЕТАЛОКОНСТРУКЦІЇ.

ЗМІСТ

ЗАВДАННЯ.....	2
РЕФЕРАТ.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ТА ТЕРМІНІВ.....	7
ВСТУП.....	12
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ МОДЕЛЮВАННЯ ВИРОБНИЧИХ ВИТРАТ.....	15
1.1 Специфіка формування собівартості на підприємствах з виготовлення металоконструкцій.....	15
1.2 Огляд комерційних систем управління проєктами.....	17
1.3 Визначення "вузьких місць" процесу калькуляції та обґрунтування необхідності розробки СППР.....	21
1.4 Формулювання функціональних та нефункціональних вимог до інформаційної системи.....	23
1.5 Висновки до першого розділу.....	25
2. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	27
2.1 Декомпозиція процесів обліку та управління витратами.....	27
2.2 Розробка математичної моделі калькуляції витрат та План-Факт аналізу....	29
2.3 Обґрунтування вибору технологічного стеку для реалізації моделі.....	37
2.4 Інформаційна модель та проєктування реляційної бази даних (опис сутностей: замовлення, матеріали, працівники).....	40
2.5 Висновки до другого розділу.....	42
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБ-ЗАСТОСУНКУ.....	44
3.1 Архітектура програмного рішення на базі патерну MTV.....	44
3.2 Розробка серверної логіки та алгоритмів агрегації фінансових даних.....	46
3.3 Реалізація клієнтської частини та інтерактивних дашбордів моніторингу....	48

3.4 Тестування системи на змодельованих сценаріях виробництва металоконструкцій.....	50
3.5 Оцінка очікуваного економічного ефекту від впровадження системи	55
3.6 Висновки до третього розділу.....	57
ВИСНОВКИ.....	59
ПЕРЕЛІК ПОСИЛАНЬ.....	61
ДОДАТОК А. Фрагмент програмного коду. Модуль структури даних та алгоритмів розрахунку (models.py).....	63
ДОДАТОК Б. Фрагмент програмного коду. Модуль обробки запитів та бізнес- логіки (views.py).....	70

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ТА ТЕРМІНІВ

1. Скорочення та абрєвіатури

БД – база даних.

ІТП – інженерно-технічний персонал.

КМД – конструкції металеві деталіровочні (шифр креслення).

НДІ – нормативно-довідкова інформація.

СКБД – система керування базами даних.

СППР – система підтримки прийняття рішень.

ACID – Atomicity, Consistency, Isolation, Durability (атомарність, узгодженість, ізолюваність, довговічність – вимоги до транзакцій баз даних).

AJAX – Asynchronous JavaScript and XML (технологія асинхронних вебзапитів).

BOM – Bill of Materials (специфікація матеріалів).

CRUD – Create, Read, Update, Delete (базові функції управління даними: створення, читання, оновлення, видалення).

CSRF – Cross-Site Request Forgery (підробка міжсайтових запитів, тип вразливості).

DOM – Document Object Model (об'єктна модель документа).

EAM – Enterprise Asset Management (управління основними фондами та обладнанням).

ERP – Enterprise Resource Planning (система планування ресурсів підприємства).

HTMX – бібліотека для доступу до сучасних функцій браузера (AJAX, CSS Transitions) безпосередньо через HTML-атрибути.

MTV – Model-Template-View (архітектурний шаблон проектування програмного забезпечення).

ORM – Object-Relational Mapping (об'єктно-реляційне відображення для

роботи з БД).

SSR – Server-Side Rendering (генерація HTML-сторінок на стороні сервера).

TCO – Total Cost of Ownership (сукупна вартість володіння інформаційною системою).

XSS – Cross-Site Scripting (міжсайтовий скриптинг, тип вразливості).

2. Умовні позначення (символи математичної моделі)

C_{equip}^{plan} – загальні планові витрати на експлуатацію всього спектру обладнання.

$C_{equip, ind}^{plan}$ – планові витрати на амортизацію індивідуального інструменту.

C_{labor}^{plan} – планові витрати на оплату праці працівників.

$C_{mat, base}$ – витрати на базові матеріали (сировину).

$C_{mat, dep}$ – витрати на супутні (залежні) матеріали.

$C_{materials}^{plan}$ – загальні планові матеріальні витрати.

$C_{overhead}^{plan}$ – планові загальновиробничі та адміністративні витрати.

C_{total}^{fact} – фактична фінальна собівартість виробу.

C_{total}^{plan} – абсолютна планова собівартість виробу.

$I_{delay, ratio}$ – частка штрафного часу (відхилень) у фактичному результаті виробництва.

$I_{plan, ratio}$ – частка планової бази у фактичному результаті.

$K_{overtime}$ – рівень понаднормових затримок виробництва (Overtime Percentage).

K_{time} – коефіцієнт відносного збільшення часу перебування виробу в цеху.

P_d – ціна супутнього (залежного) матеріалу.

P_j – закупівельна ціна одиниці j -го базового матеріалу.

Q_j – нормативна кількість j -го базового матеріалу.

R_{crane} – глобальний тариф роботи загальновиробничого обладнання (кран-

балки).

$R_{equip, i}$ – нормативна вартість години експлуатації інструменту i -ї посади.

$R_{labor, i}$ – тарифна погодинна ставка працівника i -ї посади.

$R_{overhead}$ – глобальна ставка цехових накладних витрат за 1 годину.

T_{base} – базова нормативна трудомісткість виробу за кресленням.

$T_{calendar}^{fact}$ – загальний фактичний календарний час перебування виробу в

цеху.

$T_{\Sigma calendar}^{plan}$ – загальний плановий календарний час виготовлення виробу.

T_{labor}^{fact} – загальна фактична трудова зайнятість після фіксації збурень.

$T_{\Sigma labor}^{plan}$ – сумарна трудова зайнятість (фонд оплаченого часу).

W_i – кількість залучених робітників i -ї професії.

k_{crane} – емпіричний коефіцієнт інтенсивності використання крана.

$k_{ratio}^{(j,d)}$ – технологічний коефіцієнт пропорції між базовим (j) та залежним (d)

матеріалами.

p_i – відсоток залученості i -ї посади до виконання робіт.

ΔC_{equip} – додаткова експлуатаційна амортизація обладнання через виробничий брак.

ΔC_{labor} – прямі фінансові втрати на оплату понаднормової праці (виправлення дефектів).

$\Delta C_{materials}$ – фінансові втрати від матеріального браку (зіпсовані та супутні ресурси).

$\Delta C_{overhead}$ – фінансовий штрафний еквівалент затримки випуску (додаткова оренда).

ΔC_{total} – сумарна вартість усіх технологічних відхилень (збурень) замовлення.

$\Delta T_{calendar}$ – штрафний календарний час затримки випуску продукції.

3. Терміни

Дворівнева структура амортизації – алгоритм роздільного нарахування зносу індивідуального інструменту, що залежить від трудовитрат робітника, та зносу важкого загальноцехового обладнання, що залежить від загального календарного часу роботи бригади.

Календарний час – фактичний астрономічний час фізичного перебування конструкції (деталі) в цеху, який визначає термін її оренди.

План-Факт аналіз – інструмент управлінського та фінансового контролю, що полягає у порівнянні нормативних (планових) показників витрат сировини та часу із фактично понесеними затратами на виробництві.

Супутні матеріали – розхідні ресурси (зварювальні гази, електроди, абразивні круги), обсяг використання яких не є фіксованим, а розраховується як математична пропорція від витрат базової сировини (металопрокату).

Трудомісткість – кількісний показник затрат робочого часу персоналу (в нормо-годинах), необхідний для виконання технологічної операції або виготовлення металопродукції.

Фреймворк – програмний каркас (наприклад, Django), що визначає структуру вебзастосунку та полегшує його розробку, надаючи готові модулі для роботи з базами даних та інтерфейсом.

4. Одиниці виміру

год – година.

грн – гривня.

грн/год – гривень за годину роботи.

кВт/год – кіловат-година (одиниця споживаної потужності обладнання).

кг – кілограм.

л – літр (одиниця об'єму для технологічних газів).

нормо-год – нормо-година (одиниця вимірювання витрат праці).

т – тонна.

ВСТУП

В умовах індустріальної модернізації та зростаючого попиту на оптимізацію промислових процесів, системний аналіз та математичне моделювання витрат набули критичного значення для забезпечення конкурентоспроможності вітчизняних підприємств з виробництва металевих конструкцій. Однак, ця галузь, незважаючи на високу матеріаломісткість, стикається зі специфічними викликами, які суттєво впливають на точність формування собівартості. Ключові проблеми, такі як динамічне коливання цін на метал та енергоносії, необхідність оперативного обліку технологічного браку та складність об'єктивного оцінювання трудовитрат виробничих бригад, часто призводять до фінансових втрат, перевищення бюджетів замовлень та зниження рентабельності, особливо при виготовленні індивідуальних будівельних чи промислових конструкцій, де точність калькуляції є вирішальною. Саме тому актуальність даного дослідження зумовлена об'єктивною потребою в розробці та впровадженні інструментарію, здатного системно збирати, аналізувати та візуалізувати дані про виробничі операції, трансформуючи їх у дієві рекомендації для контролю витрат. Створення такої системи дозволить мінімізувати вплив людського фактору, покращити прогнозування фінансових результатів та забезпечити точний План-Факт аналіз між плановими нормативами й реальними показниками цеху.

Метою створення системи моделювання та контролю виробничих витрат є підвищення ефективності управління фінансово-технологічними процесами при виготовленні металевих конструкцій у форматі автоматизованого контролю собівартості. Система моделювання витрат — це інформаційна система класу систем підтримки прийняття рішень (СППР), яка містить типовий функціонал для обліку матеріально-технічних ресурсів, але при цьому має низку специфічних можливостей, адаптованих під вимоги розрахунку параметрів металообробки.

Однією з особливостей системи є нетривіальна архітектура, побудована на принципі динамічного асинхронного оновлення інтерфейсу без повного перезавантаження веб-сторінок. Йдеться про те, що клієнтська та серверна частини взаємодіють через комбінацію патерну MTV (Model-Template-View) та технології NTMX, яка дозволяє підвантажувати змінені блоки даних (таблиці витрат, списки матеріалів) безпосередньо через AJAX-запити до контролерів фреймворку. Такий підхід дозволив суттєво пришвидшити роботу додатка, зменшити трафік та мінімізувати навантаження як на сервер, так і на СКБД при виконанні складних аналітичних розрахунків.

Загалом система складається з трьох основних підсистем:

- ~ модуль для керівників та економістів підприємства, який забезпечує моніторинг План-Факт відхилень та стратегічне планування;
- ~ підсистема для технологів та майстрів цехів, де фіксуються відпрацьовані години бригад, списання металопрокату та обсяги технологічного браку;
- ~ система керування базами даних.

Розподіл матеріальних ресурсів та завантаження обладнання здійснюється з урахуванням технологічних карт, а також впливає на формування підсумкової собівартості готової продукції. Система аналізу витрат виробництва металевих конструкцій веде облік стану кожного активного замовлення та дозволяє відображати основні метрики його виконання. У рамках системи реалізовано:

- ~ технологічний модуль обліку, де зібрані всі актуальні виробничі замовлення, кожне з яких має власну специфікацію матеріалів, нормативів часу та задіяних верстатів;
- ~ інфографічну сторінку, що відображає інтерактивні графіки План-Факт аналізу, загальну статистику перевитрат по підрозділах, статус виконання етапів та динаміку виникнення браку;
- ~ механізм обчислення повної собівартості, який залежить від прямих матеріальних витрат, вартості відпрацьованих бригадами годин,

амортизаційного зносу обладнання та збитків від дефектів;

~ репозиторій нормативно-довідкової інформації, що містить як поточні прейскуранти цін на метал, так і завершені калькуляційні карти з різним рівнем доступу для внутрішнього користування.

Підсистема зберігання даних базується на PostgreSQL — цю СКБД було обрано з огляду на її стабільність, високу швидкість обробки складних реляційних запитів з математичними агрегаціями та відкриту ліцензію.

Проект системи моделювання витрат розроблено з використанням фреймворку Django, що працює на мові Python. Окрім збору та зберігання інформації, система забезпечує відображення даних у вигляді таблиць і графіків у режимі реального часу за допомогою бібліотеки Chart.js, формування різноманітних звітів, а також інструменти для архівації та захисту інформації.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ МОДЕЛЮВАННЯ ВИРОБНИЧИХ ВИТРАТ

1.1 Специфіка формування собівартості на підприємствах з виготовлення металоконструкцій

Виробництво металевих конструкцій характеризується високим рівнем матеріаломісткості та необхідністю точного контролю на кожному етапі технологічного процесу. З позицій системного аналізу [1], процес формування собівартості у цій галузі залежить від множини чітко визначених вхідних параметрів: обсягу витрачених матеріалів, тарифних ставок працівників та часу роботи конкретного обладнання.

Специфіка калькулювання витрат на заводах з виробництва металоконструкцій суттєво відрізняється від масового чи потокового виробництва. Це зумовлено низкою галузевих особливостей, які вимагають застосування спеціалізованих підходів до обліку.

Основні фактори, що визначають специфіку формування собівартості:

1. Індивідуальний характер виробництва. Більшість металевих конструкцій виготовляються за унікальними проектними кресленнями. Для кожного нового замовлення необхідно розробляти індивідуальну калькуляційну карту, що вимагає точного попереднього планування нормативів для кожної деталі (наприклад, конкретної балки).

2. Складна структура матеріальних витрат. Основна частка вартості готової металоконструкції формується за рахунок прямих матеріальних витрат (листовий та фасонний прокат). Окремою аналітичною проблемою є точний облік супутніх витратних матеріалів (зварювальні гази, дрід, абразиви). Їх витрата часто не є фіксованою, а технологічно залежить від обсягів використання основного металу через систему пропорційних коефіцієнтів, що значно ускладнює ручну калькуляцію.

3. Розмежування трудовитрат та календарного часу. Виробничий

процес виконується бригадним методом. Коректний розрахунок оплати праці вимагає точної фіксації відпрацьованого часу (людино-годин) кожним робітником із урахуванням його тарифної ставки та відсотка залученості. При цьому критично важливо розмежовувати загальну трудомісткість операції та її фактичну календарну тривалість. Залучення більшої кількості робітників дозволяє скоротити календарний час перебування виробу в цеху, що прямо впливає на зниження загальновиробничих (адміністративних) витрат.

4. Дворівнева структура амортизації обладнання. Системний підхід до оцінки експлуатаційних витрат вимагає розділення обладнання на дві категорії. Перша — це індивідуальний інструмент, жорстко закріплений за конкретною посадою (наприклад, зварювальний напівавтомат), амортизація якого залежить виключно від відпрацьованих годин конкретного робітника. Друга категорія — це загальноцехове обладнання (наприклад, використання мостової кран-балки), вартість експлуатації якого є глобальною змінною та розподіляється на загальний час роботи бригади. Точний облік обох рівнів є обов'язковим для формування об'єктивної фактичної собівартості.

5. Вплив технологічного браку. Виробничі операції несуть ризики виникнення дефектів. виправлення браку та помилок призводить до появи непередбачуваних витрат двох чітких типів: необхідності списання додаткових матеріалів та витрати понаднормового часу робітників. Традиційні системи обліку здебільшого фіксують ці збитки постфактум, без можливості оперативного аналізу в процесі роботи.

Системний підхід до управління витратами

Аналіз вищезазначених особливостей показує, що ефективно управління витратами вимагає створення інформаційної системи, орієнтованої безпосередньо на виробничий цех [2]. Основним інструментом такого контролю виступає План-Факт аналіз — порівняння нормативних (планових) показників витрат сировини та часу із фактичними даними.

Головним принципом оптимізації економіки підприємства в рамках такої системи є недопущення перевитрат: забезпечення умов, за яких підприємство не

витрачає більше ресурсів там, де, згідно з технологією, можна витратити визначену меншу кількість. Лише за умови своєчасного виявлення відхилень керівництво отримує можливість обґрунтовано контролювати процес та дотримуватися бюджету замовлення.

1.2 Огляд комерційних систем управління проектами

Вирішення задачі ефективного калькулювання витрат та План-Факт аналізу на промислових підприємствах традиційно спирається на використання автоматизованих інформаційних систем. На сучасному ринку програмного забезпечення представлено широкий спектр рішень для управління виробничими процесами. З точки зору системного підходу, існуючі програмні продукти можна розділити на три основні класи: важкі інтегровані ERP-системи, спеціалізовані галузеві рішення та системи модульного типу [3].

1. Інтегровані системи класу ERP (Enterprise Resource Planning) До цієї категорії належать потужні світові рішення, такі як SAP ERP, Microsoft Dynamics 365, а також флагманські українські розробки, наприклад, IT-Enterprise.

Переваги: Ці системи охоплюють абсолютно всі контури управління підприємством — від фінансів та кадрового обліку до логістики і технічного обслуговування обладнання (EAM). Вони здатні обробляти величезні масиви даних і підтримують складні алгоритми багаторівневого розрахунку собівартості, що наочно демонструє перевантажений деталями інтерфейс типової ERP-системи (рис. 1.1).

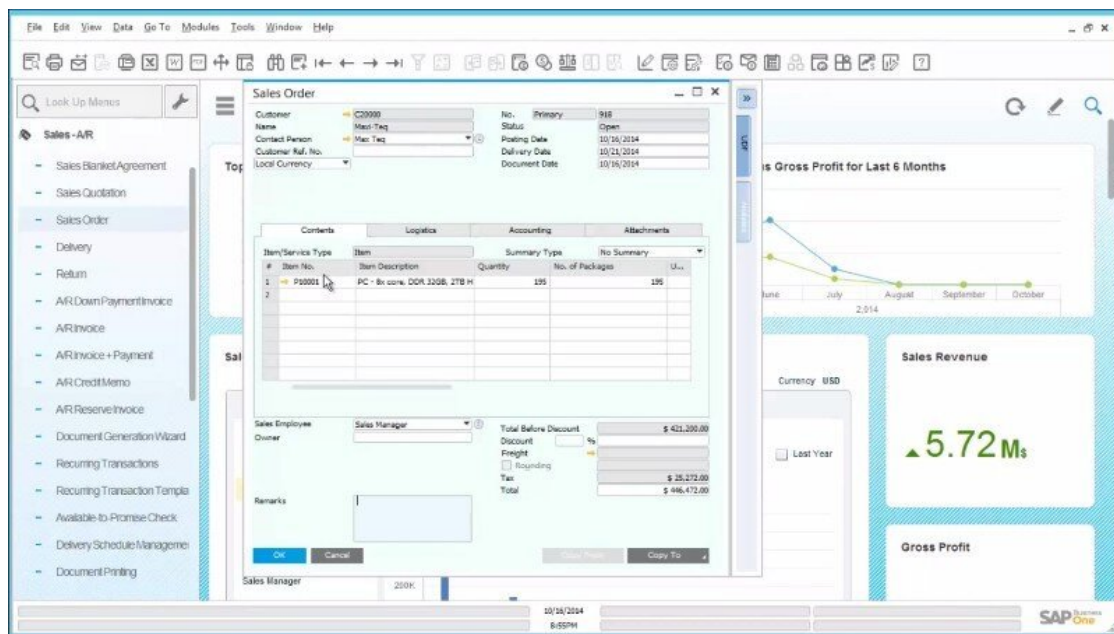


Рисунок 1.1 – Інтерфейс модуля управління виробництвом інтегрованої ERP-системи

Недоліки: Головним бар'єром для впровадження таких систем на підприємствах середнього масштабу з виготовлення металоконструкцій є їхня надлишковість та колосальна вартість (як ліцензій, так і консалтингових послуг із впровадження). Крім того, архітектура важких ERP-систем є вкрай інертною [4]. Процес фіксації відхилень (наприклад, списання додаткових електродів чи фіксація понаднормових годин зварювальника) вимагає заповнення великої кількості електронних документів та підписів у системі, що унеможливує оперативний План-Факт аналіз безпосередньо в цеху.

2. Системи модульного типу та Open-Source рішення (на прикладі Odoo) Odoo — це комплексна система управління бізнесом з відкритою архітектурою, яка складається з незалежних модулів (Виробництво, Склад, Проекти тощо).

Переваги: На відміну від класичних ERP, модульні системи дозволяють підприємству впроваджувати лише необхідний функціонал, поступово масштабуючи систему. Вони мають сучасні веб-інтерфейси та відкритий програмний код, що дозволяє розробникам адаптувати логіку під потреби замовника.

Недоліки: Базовий модуль «Виробництво» в таких системах

розрахований переважно на дискретне складання за жорсткими специфікаціями BOM (Bill of Materials), приклад якої наведено на рис. 1.2.

Product		[FURN_9666] Table	
Product Variant			
Quantity	1.00 Units		
Reference			
BoM Type	<input checked="" type="radio"/> Manufacture this product <input type="radio"/> Kit <input type="radio"/> Subcontracting		
Company	My Company (San Francisco)		
<div style="display: flex; border-bottom: 1px solid #ccc;"> <div style="border-right: 1px solid #ccc; padding: 2px 5px;">Components</div> <div style="padding: 2px 5px;">Operations</div> <div style="padding: 2px 5px;">Miscellaneous</div> </div>			
Component	Quantity	Product	Unit of Measure
[FURN_8522] Table Top	0		Units
[FURN_2333] Table Leg	0		Units
[CONS_89957] Bolt	0		Units
[CONS_25630] Screw	0		Units
Add a line			

Рисунок 1.2 – Відображення багаторівневої специфікації матеріалів у системі Odoo

У базовій версії таких систем відсутня можливість гнучкого налаштування динамічних технологічних пропорцій (наприклад, автоматичного розрахунку витрати захисного газу залежно від обсягу використаного металу). Для реалізації дворівневої амортизації обладнання та деталізованого План-Факт аналізу такі системи вимагають глибокої та дороговартісної програмної модифікації ядра.

3. Універсальні системи управління проектами та тайм-трекінгу До цього класу можна віднести системи типу Jira, Asana або спеціалізовані тайм-трекери. Хоча вони початково створювалися для IT-сектору та сфери послуг, деякі виробничі компанії намагаються адаптувати їх для контролю виконання

замовлень (рис. 1.3).

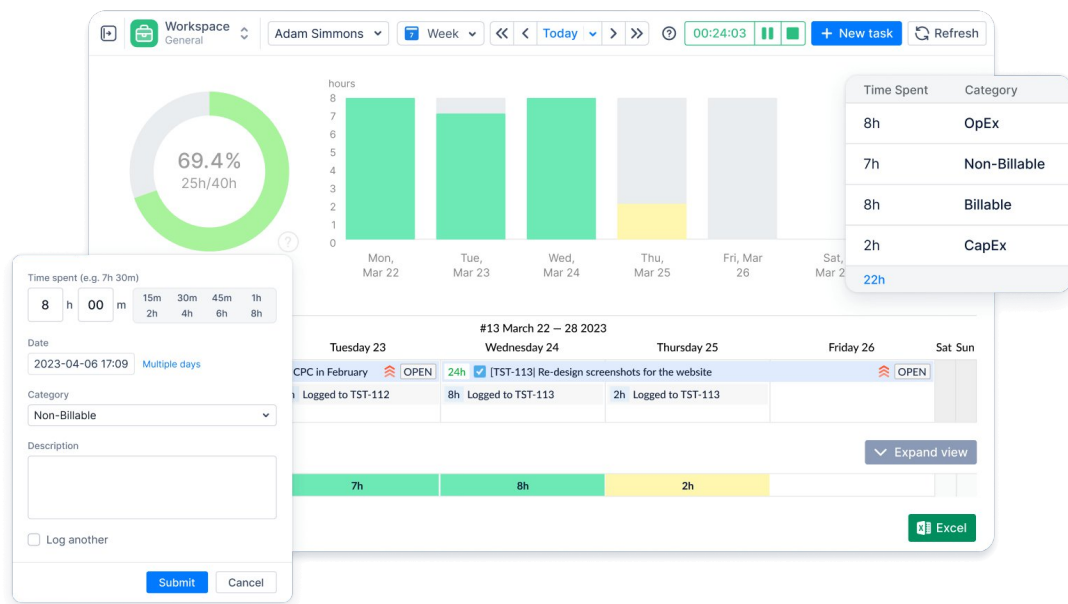


Рисунок 1.3 – Інтерфейс управління завданнями та обліку часу в системі Jira

Переваги: Висока швидкість розгортання, зручні інструменти для візуального управління завданнями та точний облік витраченого часу персоналу (людино-годин).

Недоліки: Ці системи мають критичне обмеження для виробництва металоконструкцій — вони є виключно "task-орієнтованими", а не "матеріало-орієнтованими". Вони не здатні вести реляційний облік матеріалів, розраховувати амортизацію станків чи консолідувати ці дані у єдину фінансову собівартість виробу [5].

Висновки щодо аналізу ринку

Проведений аналіз демонструє, що використання універсальних комерційних систем для вирішення задачі моделювання витрат виробництва металоконструкцій супроводжується низкою компромісів. Важкі ERP-системи економічно недоцільні та перевантажені зайвим функціоналом, модульні рішення вимагають суттєвої переробки базової логіки, а системи управління проектами не підтримують матеріальний облік.

Більшість розглянутих систем не надають зручного інструментарію для

швидкої фіксації технологічного браку та відхилень безпосередньо робітниками. Таким чином, для підприємств, які випускають індивідуальні металоконструкції, найбільш раціональним шляхом є розробка спеціалізованої "легковагової" інформаційної системи [6]. Вона повинна фокусуватися виключно на процесах калькулювання, обліку робочого часу, амортизації цільового обладнання та інтерактивному План-Факт аналізу, що забезпечить максимальну точність фінансового контролю при мінімальних витратах на впровадження

1.3 Визначення "вузьких місць" процесу калькуляції та обґрунтування необхідності розробки СППР

У процесі системного аналізу предметної області та дослідження існуючих підходів до управління виробництвом металевих конструкцій було виявлено ряд критичних "вузьких місць" (bottlenecks) [1]. Ці фактори суттєво знижують точність економічного планування та унеможливають оперативний контроль за формуванням фактичної собівартості виробів.

До основних "вузьких місць" традиційного або частково автоматизованого процесу калькуляції належать:

1. Розрив між нормованою трудомісткістю та фактичним календарним часом. При ручному розрахунку економісти спираються на базову трудомісткість за кресленням (нормо-години). Однак на практиці виконання завдання може здійснюватися бригадами з різною кількістю робітників та різним відсотком залученості кожної посади. Традиційні підходи не дозволяють динамічно перераховувати календарний час перебування конструкції в цеху, що призводить до критичних похибок у нарахуванні адміністративних та загальновиробничих витрат (вартості оренди цеху за годину).

2. Втрата даних про супутні матеріали при виникненні відхилень. У випадку технологічного браку або помилки майстра виникає необхідність

повторного виконання операцій. Якщо майстер списує додатковий листовий метал, традиційний облік часто "губить" супутні матеріали, які витрачаються пропорційно (наприклад, зварювальний дріт, захисні гази, абразивні круги). Ручний перерахунок таких технологічних пропорцій для кожного відхилення є надзвичайно трудомістким і часто ігнорується, що спотворює фінальну собівартість.

3. Складність алокації експлуатаційних витрат. Розрахунок амортизації обладнання ускладнюється його дворівневою природою. Обліковцям важко коректно розмежувати і вирахувати в режимі реального часу вартість роботи індивідуального інструменту конкретного працівника (амортизація зварювального апарату залежно від його відпрацьованих годин) та загальноцехового обладнання (робота кран-балки, яка залежить від загального календарного часу роботи всієї бригади).

4. Запізніла реакція на перевитрати (відсутність оперативного План-Факт аналізу). Найбільшим недоліком існуючих процесів є те, що фіксація виробничого браку або додаткових трудовитрат відбувається постфактум — у кінці зміни, тижня або місяця під час закриття нарядів. Керівництво дізнається про збитковість замовлення тоді, коли деталь вже відвантажена замовнику, і можливості мінімізувати фінансові втрати вичерпано.

Обґрунтування необхідності розробки Системи підтримки прийняття рішень (СППР)

Наявність зазначених "вузьких місць" робить процес управління витратами вразливим до "людського фактора" та економічно неефективним. Вирішення цієї проблеми об'єктивно вимагає переходу від методів пасивного ретроспективного обліку до проактивного моделювання.

З огляду на це, виникає необхідність розробки спеціалізованої веб-орієнтованої Системи підтримки прийняття рішень (СППР) [2]. З позицій системного аналізу, така система повинна виконувати роль єдиного інформаційного середовища для технологів, диспетчерів та економістів підприємства. На відміну від важких ERP-систем, вона має бути сфокусована на

вирішенні конкретної задачі — миттєвому План-Факт аналізу.

Впровадження розробленої СППР дозволить автоматизувати математичні розрахунки технологічних пропорцій, забезпечити коректний розподіл амортизаційних та цехових витрат на основі актуального складу бригади, а головне — надати керівництву інтерактивний інструментарій для моніторингу виробничих відхилень у режимі реального часу. Це забезпечить виконання головного принципу оптимізації фінансових ресурсів: недопущення неконтрольованих перевитрат на кожному етапі створення металоконструкції.

Чітке розуміння проблематики та "вузьких місць" дозволяє перейти до формалізації конкретних вимог до архітектури, математичної бази та програмного інтерфейсу майбутньої системи, що буде розглянуто в наступному підрозділі [7].

1.4 Формулювання функціональних та нефункціональних вимог до інформаційної системи

На основі проведеного аналізу предметної області та виявлених недоліків існуючих рішень, було сформовано комплексний перелік вимог до розроблюваної Системи підтримки прийняття рішень (СППР) для моделювання витрат на виробництві металевих конструкцій. Згідно з методологією проєктування інформаційних систем [3], вимоги поділено на функціональні (що система повинна робити) та нефункціональні (якими властивостями вона повинна наділятися).

Функціональні вимоги (ФВ)

Функціонал системи повинен повністю покривати потреби економістів та майстрів цеху у веденні обліку та моніторингу виробничого процесу:

~ ФВ.1. Управління нормативно-довідковою інформацією. Система повинна забезпечувати ведення довідників персоналу (з фіксацією тарифних

ставок та індивідуального обладнання), довідників матеріалів (з розбиттям на категорії) та глобальних налаштувань цеху (вартість оренди, тарифи роботи загальноцехового обладнання).

~ ФВ.2. Динамічне формування планової собівартості. Система має дозволяти створювати цифрові проекти (балки) із зазначенням базової трудомісткості. Має бути реалізований алгоритм автоматичного розрахунку потреби в супутніх матеріалах на основі технологічних пропорцій до основного металу.

~ ФВ.3. Облік відпрацьованого часу та складу бригади. Програмний продукт повинен розраховувати календарний час перебування виробу в цеху залежно від кількості залучених робітників та їхньої відсоткової участі в конкретному проекті.

~ ФВ.4. Дворівневий розрахунок амортизації. Система має розраховувати знос індивідуального інструменту робітників пропорційно до їхніх відпрацьованих годин, а знос важкого обладнання (кран-балки) — пропорційно до загального часу роботи всієї бригади.

~ ФВ.5. Фіксація виробничих відхилень. Система повинна надавати диспетчерам функціонал для миттєвої реєстрації технологічного браку або помилок (додаткових витрат матеріалів або понаднормових годин) із зазначенням причини.

~ ФВ.6. Аналітика та експорт. Має бути реалізований модуль інтерактивного План-Факт аналізу для порівняння планових та фактичних показників, а також функція експорту зведеної калькуляції та журналу відхилень у формат таблиць Excel.

Нефункціональні вимоги (НФВ)

Для забезпечення стабільності та зручності експлуатації в умовах промислового підприємства, система повинна відповідати наступним архітектурним та технологічним критеріям:

~ НФВ.1. Архітектура та доступність. Система має бути реалізована як веб-застосунок (клієнт-серверна архітектура), що дозволить користувачам

працювати з нею через звичайний браузер без необхідності встановлення додаткового програмного забезпечення на робочі станції в цехах.

~ НФВ.2. Ергономічність та швидкодія. Інтерфейс користувача повинен бути інтуїтивно зрозумілим і не перевантаженим. Для пришвидшення роботи та зниження навантаження на мережу, оновлення аналітичних розрахунків та графіків має відбуватися динамічно, без повного перезавантаження веб-сторінок.

~ НФВ.3. Надійність зберігання даних. З огляду на складність зв'язків між сутностями (проекти, бригади, відхилення, матеріали), архітектура бази даних повинна будуватися на основі реляційної моделі для забезпечення абсолютної транзакційної цілісності фінансової інформації [5].

1.5 Висновки до першого розділу

У першому розділі кваліфікаційної роботи проведено системний аналіз предметної області та досліджено специфіку управління фінансовими і матеріальними потоками на підприємствах з виготовлення металевих конструкцій.

За результатами дослідження зроблено наступні висновки:

1. Доведено, що традиційні методи калькуляції не відповідають вимогам сучасного індивідуального виробництва металоконструкцій. Висока питома вага матеріальних витрат, необхідність розрахунку пропорцій супутніх матеріалів та дворівнева структура амортизації обладнання вимагають відмови від ретроспективного обліку на користь проактивного моделювання.

2. Здійснено критичний огляд існуючих автоматизованих систем управління (інтегровані ERP-системи, модульні рішення відкритого типу та task-системи тайм-трекінгу) [3]. Встановлено, що комерційні продукти є або економічно недоцільними та функціонально надлишковими для середніх

промислових підприємств, або не підтримують необхідної технологічної логіки для гнучкого матеріального та погодинного обліку.

3. Виявлено ключові "вузькі місця" існуючого процесу калькуляції: нездатність облікових систем динамічно відстежувати розрив між базовою трудомісткістю виробу та фактичним календарним часом його перебування в цеху, а також запізнiла фіксація технологічного браку. Це підтверджує об'єктивну необхідність створення спеціалізованого програмного рішення.

4. Сформульовано комплекс функціональних та нефункціональних вимог до розроблюваної Системи підтримки прийняття рішень (СППР) [2]. Визначено, що система повинна мати клієнт-серверну архітектуру, забезпечувати миттєву реєстрацію виробничих відхилень безпосередньо на робочих місцях та автоматично генерувати інтерактивний План-Факт аналіз собівартості.

Отримані в першому розділі результати аналізу є концептуальним базисом для подальшого виконання роботи. Вони детермінують метрику та структуру даних майбутнього застосунку, що дозволяє перейти до наступного етапу дослідження — розробки математичних моделей калькуляції витрат та проектування архітектури реляційної бази даних, що буде детально розкрито в другому розділі.

2. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Декомпозиція процесів обліку та управління витратами

З позицій системного підходу, процес управління витратами на виробництві металопродукції є складною динамічною системою. Для її успішного математичного моделювання та подальшої програмної реалізації необхідно застосувати метод структурної декомпозиції [1, 3]. Це дозволяє розбити глобальну задачу калькуляції собівартості на функціонально незалежні, але інформаційно пов'язані підпроцеси (модулі).

Аналіз бізнес-логіки виробничого підприємства та сформовані у першому розділі вимоги дозволяють декомпонувати загальний процес моделювання витрат на чотири базові підпроцеси (етапи) [7].

1. Підсистема формування нормативно-довідкової бази (НДІ) Цей етап є фундаментом для всіх подальших розрахунків. На цьому рівні система агрегує статичні та умовно-статичні вхідні параметри:

~ Матеріальні ресурси: формування номенклатури металопродукції та розхідників із зазначенням їх актуальної закупівельної ціни та одиниць виміру.

~ Технологічні пропорції: задання коефіцієнтів залежності витрат супутніх матеріалів від обсягу основного металу (наприклад, витрата захисного газу при зварюванні).

~ Кадрові та експлуатаційні ресурси: фіксація погодинних тарифних ставок працівників, а також вартості амортизації індивідуального інструменту, що закріплений за кожною посадою.

~ Глобальні константи: визначення вартості оренди цеху (загальновиробничі витрати) та експлуатації важкого обладнання (кран-балки) за одну робочу годину.

2. Підсистема моделювання планової (нормативної) собівартості Після отримання креслення (проєкту) металопродукції, система ініціює процес

створення цифрового "Плану". Декомпозиція цього етапу включає:

~ Фіксацію базової трудомісткості виробу (в нормо-годинах) згідно з технологічною картою.

~ Розподіл відсотків залученості кожної виробничої ролі (бригади) та визначення кількості робітників на етапі. Це дозволяє системі розмежувати загальну трудомісткість та фактичний календарний час знаходження виробу в цеху.

~ Формування специфікації (Bill of Materials) – закріплення за проектом необхідної кількості базових матеріалів із подальшим автоматичним розрахунком супутніх розхідників [4].

3. Підсистема реєстрації виробничих відхилень (Факту) Оскільки реальний виробничий процес схильний до збурень (технологічного браку, помилок персоналу), критичним підпроцесом є оперативна реєстрація відхилень. Цей блок декомпозується на два вектори:

~ Матеріальні відхилення: реєстрація додатково списаного матеріалу на виправлення браку, що автоматично тягне за собою перерахунок пов'язаних технологічних пропорцій (додаткових розхідників).

~ Трудові відхилення: фіксація понаднормового часу конкретних робітників, що призводить не лише до зростання фонду оплати праці та зносу індивідуального інструменту, але й до збільшення календарного часу оренди цеху.

4. Підсистема агрегації та План-Факт аналізу Фінальний підпроцес полягає у зведенні даних з попередніх етапів. Математичний апарат системи порівнює вектори планових та фактичних витрат за чотирма основними категоріями: прямі матеріальні витрати, заробітна плата, амортизація обладнання та цехові накладні витрати. Результатом роботи цього блоку є генерація інтерактивних дашбордів та формування зведеного фінансового звіту, що дозволяє керівництву приймати оперативні управлінські рішення.



Рисунок 2.1 – Структурна декомпозиція процесу моделювання виробничих витрат

Проведена декомпозиція дозволяє чітко окреслити межі інформаційної системи та перейти до формалізації виділених підпроцесів у вигляді строгих математичних моделей, що є предметом наступного підрозділу.

2.2 Розробка математичної моделі калькуляції витрат та План-Факт аналізу

Основою інформаційної системи є розроблена математична модель, яка формалізує складні процеси трансформації виробничих ресурсів (часу, матеріалів, експлуатації обладнання) у фінансовий еквівалент собівартості. Запропонована модель базується на принципах процесно-орієнтованого обліку (Activity-Based Costing) і складається з семи функціональних блоків [1, 2]. Вона описує як ідеальний (плановий) стан системи, так і її динамічну поведінку при виникненні технологічних збурень (відхилень).

БЛОК 1. Часова модель виробництва (Трудомісткість та

Календарний розклад) Традиційні системи обліку часто ототожнюють трудовитрати персоналу із часом виконання замовлення. Ключовою інновацією даної моделі є математичне розділення понять загальної трудомісткості (людино-годин) та фактичного календарного часу (годин фізичного перебування конструкції в цеху), який нелінійно залежить від кількості залучених робітників.

Плановий час роботи i -ї професії ($T_{labor,i}^{plan}$) визначається як частка від загальної базової трудомісткості виробу згідно з технологічною картою:

$$T_{labor,i}^{plan} = T_{base} \cdot \frac{p_i}{100} \quad (2.1)$$

де T_{base} – базова нормативна трудомісткість виробу за кресленням (в годинах);

p_i – відсоток залученості i -ї посади до виконання робіт над цим виробом.

Сумарна трудова зайнятість (фонд оплаченого часу, за який підприємство повинно нарахувати заробітну плату) дорівнює сумі годин усіх залучених професій:

$$T_{\Sigma labor}^{plan} = \sum_{i=1}^n T_{labor,i}^{plan} \quad (2.2)$$

Календарний час виконання етапу обернено пропорційний кількості робітників у бригаді. Системний ефект полягає в тому, що залучення більшої кількості фахівців скорочує час перебування виробу на стенді:

$$T_{calendar,i} = \frac{T_{labor,i}^{plan}}{\max(1, W_i)} \quad (2.3)$$

де W_i — кількість залучених робітників i -ї професії (функція *max* використовується для запобігання критичній помилці ділення на нуль в алгоритмах програми).

Загальний плановий календарний час виготовлення виробу:

$$T_{\Sigma calendar}^{plan} = \sum_{i=1}^n T_{calendar, i} \quad (2.4)$$

БЛОК 2. Модель прямих виробничих витрат (Зарплата та Амортизація) Цей блок перетворює часові показники у фінансові еквіваленти. Планові витрати на оплату праці розраховуються на основі загальної трудомісткості та індивідуальних погодинних ставок персоналу:

$$C_{labor}^{plan} = \sum_{i=1}^n (T_{labor, i}^{plan} \cdot R_{labor, i}) \quad (2.5)$$

де $R_{labor, i}$ — тарифна погодинна ставка працівника i -ї посади.

Дворівнева система амортизації вимагає окремого розрахунку зносу індивідуального інструменту (наприклад, зварювального напівавтомата), що жорстко закріплений за конкретними посадами:

$$C_{equip, ind}^{plan} = \sum_{i=1}^n (T_{labor, i}^{plan} \cdot R_{equip, i}) \quad (2.6)$$

де $R_{equip, i}$ — нормативна вартість 1 години експлуатації інструменту, закріпленого за i -ю посадою.

Натомість витрати на роботу загальнопромислового обладнання (наприклад, мостової кран-балки) є глобальними і розподіляються пропорційно загальній трудомісткості з урахуванням емпіричного коефіцієнта інтенсивності:

$$C_{crane}^{plan} = T_{\Sigma labor}^{plan} \cdot k_{crane} \cdot R_{crane} \quad (2.7)$$

де k_{crane} — коефіцієнт інтенсивності використання крана; R_{crane} — глобальний тариф роботи кран-балки.

Загальні планові витрати на експлуатацію всього спектру обладнання:

$$C_{equip}^{plan} = C_{equip, ind}^{plan} + C_{crane}^{plan} \quad (2.8)$$

БЛОК 3. Матеріальна модель (Специфікація та Супутні витрати)

Матеріальна модель вирішує проблему прихованих витрат, автоматично дораховуючи необхідні розхідні матеріали на основі обсягів основного металу.

Витрати на базові матеріали (сировину):

$$C_{mat, base} = \sum_{j=1}^m (Q_j \cdot P_j) \quad (2.9)$$

де Q_j — нормативна кількість j -го базового матеріалу; P_j — закупівельна ціна за його одиницю.

Витрати на супутні матеріали розраховуються через матрицю технологічних коефіцієнтів (наприклад, скільки кілограмів зварювального дроту потрібно на 1 тону сталі):

$$C_{mat, dep} = \sum_{j=1}^m \sum_{d=1}^k (Q_j \cdot k_{ratio}^{(j,d)} \cdot P_d) \quad (2.10)$$

де $k_{ratio}^{(j,d)}$ — технологічний коефіцієнт співвідношення між базовим матеріалом j та залежним матеріалом d ; P_d — ціна супутнього матеріалу.

Загальні планові матеріальні витрати:

$$C_{materials}^{plan} = C_{mat, base} + C_{mat, dep} \quad (2.11)$$

БЛОК 4. Загальновиробничі та адміністративні витрати

Найважливішою умовою коректної алокації накладних витрат (оренда цеху, освітлення, заробітна плата адміністративного персоналу) є їх прив'язка саме до *календарного часу*, а не до трудовитрат:

$$C_{overhead}^{plan} = T_{\Sigma calendar}^{plan} \cdot R_{overhead} \quad (2.12)$$

де $R_{overhead}$ — глобальна ставка цехових накладних витрат за 1 годину реального часу.

БЛОК 5. Модель управління збуреннями (Брак та Перевитрати)

Цей блок математично описує обчислення фактичних відхилень (ΔC), що виникають у реальному виробничому процесі і знижують рентабельність замовлення.

Фінансові втрати від матеріального браку враховують ланцюгове списання супутніх матеріалів (якщо зіпсовано метал, пропорційно списується і витрачений на нього газ та електроди):

$$\Delta C_{materials} = \sum_{dev=1}^x (Q_{dev} \cdot P_{dev} + \Sigma(Q_{dev} \cdot k_{ratio} \cdot P_{dep})) \quad (2.13)$$

Прямі фінансові втрати від додаткових трудовитрат (оплата понаднормової роботи для виправлення дефекту):

$$\Delta C_{labor} = \sum_{dev=1}^y (\Delta T_{labor, i} \cdot R_{labor, i}) \quad (2.14)$$

Додаткова експлуатаційна амортизація обладнання під час виправлення браку:

$$\Delta C_{equip} = \sum_{dev=1}^y (\Delta T_{labor, i} \cdot (R_{equip, i} + k_{crane} \cdot R_{crane})) \quad (2.15)$$

Штрафний календарний час та фінансовий еквівалент затримки випуску (додаткова оренда цеху):

$$\Delta T_{calendar} = \sum_{dev=1}^y \frac{\Delta T_{labor, i}}{\max(1, W_i)} \quad (2.16)$$

$$\Delta C_{overhead} = \Delta T_{calendar} \cdot R_{overhead} \quad (2.17)$$

БЛОК 6. Цільова функція системи (План-Факт Аналіз) Зведення результатів математичного моделювання здійснюється шляхом агрегації планових та штрафних показників для отримання фінальної картини [1]. Абсолютна планова собівартість виробу:

$$C_{total}^{plan} = C_{materials}^{plan} + C_{labor}^{plan} + C_{equip}^{plan} + C_{overhead}^{plan} \quad (2.18)$$

Сумарна вартість усіх технологічних відхилень:

$$\Delta C_{total} = \Delta C_{materials} + \Delta C_{labor} + \Delta C_{equip} + \Delta C_{overhead} \quad (2.19)$$

Фактична собівартість (вихідна цільова функція системи):

$$C_{total}^{fact} = C_{total}^{plan} + \Delta C_{total} \quad (2.20)$$

БЛОК 7. Аналітичні показники ефективності (Метрики дашборду)

Для забезпечення підтримки прийняття управлінських рішень, система спочатку агрегує всі фактичні трудовитрати та час оренди після фіксації збурень, а потім трансформує їх у відносні індикатори ефективності [2].

Загальна фактична трудомісткість та календарний час:

$$T_{labor}^{fact} = T_{\Sigma labor}^{plan} + \sum_{dev=1}^y \Delta T_{labor, i} \quad (2.21)$$

$$T_{calendar}^{fact} = T_{\Sigma calendar}^{plan} + \Delta T_{calendar} \quad (2.22)$$

Коефіцієнт відносного збільшення часу (Time Percentage) показує, на

скільки відсотків фактичний час перебування балки в цеху відрізняється від ідеального плану:

$$K_{time} = \left(\frac{T_{calendar}^{fact}}{T_{\Sigma calendar}^{plan}} \right) \cdot 100\% \quad (2.23)$$

Рівень понаднормових затримок (Overtime Percentage) визначає чистий відсоток перевитрати календарного часу (якщо роботи виконано швидше або вчасно, показник дорівнює нулю):

$$K_{overtime} = \begin{cases} K_{time} - 100\% & \text{якщо } K_{time} > 100\% \\ 0, & \text{якщо } K_{time} \leq 100\% \end{cases} \quad (2.24)$$

Для коректної візуалізації План-Факт аналізу на графіках (Progress Bar Widths), абсолютні значення перераховуються у відносні частки від фактичного підсумку. Частка планової бази у фактичному результаті:

$$I_{plan, ratio} = \left(\frac{T_{\Sigma calendar}^{plan}}{T_{calendar}^{fact}} \right) \cdot 100\% \quad (2.25)$$

Частка штрафного часу (відхилень) у фактичному результаті:

$$I_{delay, ratio} = \left(\frac{\Delta T_{calendar}}{T_{calendar}^{fact}} \right) \cdot 100\% \quad (2.26)$$

2.3 Обґрунтування вибору технологічного стеку для реалізації моделі

Успішна практична реалізація інформаційної системи підтримки прийняття рішень (СППР) для калькуляції собівартості металоконструкцій безпосередньо залежить від обґрунтованого вибору інструментальних засобів розробки. З погляду системного аналізу, програмний стек повинен задовольняти вимоги надійності, масштабованості, високої швидкості обробки даних, а також забезпечувати можливість швидкого прототипування без надлишкових архітектурних ускладнень. З урахуванням специфіки сформульованих у підрозділі 2.2 математичних задач, було визначено оптимальний комплекс технологій.

Серверна архітектура: Python та фреймворк Django Основою для програмної реалізації обчислювальних алгоритмів та бізнес-логіки системи обрано високорівневу мову програмування Python [8, 9]. Дана мова володіє лаконічним синтаксисом і потужною екосистемою, що мінімізує витрати часу на написання рутинного коду та дозволяє зосередитися на математичних розрахунках трудомісткості та матеріальних витрат.

Як базовий вебфреймворк для розробки серверної частини (backend) обрано Django. Це повнофункціональний інструмент розробки, який реалізує архітектурний патерн MTV (Model-Template-View) [10, 11]. Використання Django надає системі такі критично важливі переваги:

1. Об'єктно-реляційне відображення (ORM): Дозволяє взаємодіяти з базою даних за допомогою об'єктів мови Python. Це значно спрощує оперування складними сутностями предметної області (технологічними картами, тарифами працівників, специфікаціями матеріалів) без прямого написання складних SQL-запитів.

2. Адміністративна панель: Вбудований модуль Django автоматично генерує інтерфейс для управління нормативно-довідковою інформацією. Це

дозволяє миттєво налаштувати облік номенклатури металопродукату, параметрів інструментів та посад без додаткових витрат часу на розробку.

3. Високий рівень безпеки: Фреймворк забезпечує вбудований захист від типових вразливостей вебдодатків (XSS, CSRF, SQL-ін'єкції), що є обов'язковою вимогою при роботі з комерційною та фінансовою інформацією підприємства.

Організація збереження даних: SQLite Для збереження інформації про структуру виробничих процесів, нормативно-довідкову базу та зафіксовані відхилення використовуються реляційні бази даних. В якості системи керування базами даних (СКБД) для розробленої СППР було обрано полегшену вбудовану базу даних SQLite [12].

Оскільки розроблювана система позиціонується як функціональний інструмент для локального використання інженерно-технічним персоналом (диспетчерами, технологами) безпосередньо в умовах цеху, застосування SQLite є технічно та економічно обґрунтованим рішенням. Вона забезпечує повноцінну підтримку транзакцій (відповідно до стандарту ACID), що гарантує абсолютну цілісність усіх калькуляційних розрахунків при збереженні даних. При цьому SQLite ідеально інтегрується з Django ORM і не потребує розгортання чи адміністрування окремого сервера баз даних. Це суттєво прискорює процес розробки, робить інформаційну систему максимально мобільною, легкою у розгортанні та абсолютно невимогливою до апаратних ресурсів підприємства.

Клієнтська архітектура (Frontend): HTML5, CSS3, Bootstrap 5 та HTMX При проектуванні користувацького інтерфейсу було прийнято рішення відмовитися від важких клієнтських фреймворків (наприклад, React або Vue) на користь класичного та продуктивного стеку. Такий підхід гарантує максимальну швидкість завантаження та кросплатформність. Для створення адаптивного дизайну робочого місця диспетчера та технолога застосовано CSS-фреймворк Bootstrap 5, який надає готовий набір стилізованих компонентів (сітки, картки, форми) [13].

3 метою забезпечення динамічності інтерфейсу та реалізації

асинхронного обміну даними між клієнтом і сервером використовується бібліотека HTMX [14]. Її застосування дозволяє надсилати AJAX-запити безпосередньо через спеціальні HTML-атрибути (наприклад, при фіксації виробничого браку або зміні кількості матеріалів), миттєво оновлюючи розрахунки собівартості без повного перезавантаження сторінки. Це зводить до мінімуму обсяг власного JavaScript-коду та робить інтерфейс максимально чуйним.

Візуалізація даних та генерація звітів: Chart.js та Openpxl Для наочного графічного відображення результатів моделювання (Блок 7 математичної моделі) у користувацькому інтерфейсі застосовується легка JavaScript-бібліотека Chart.js [15]. Вона візуалізує аналітичні розрахунки у вигляді динамічних діаграм (наприклад, структури фактичних витрат) безпосередньо в браузері. Окрім цього, для забезпечення сумісності розрахунків із корпоративними стандартами документообігу, у технологічний стек включено Python-бібліотеку openpxl [16]. Вона відповідає за автоматичну генерацію та вивантаження зведеного "План-Фактного аналізу", технологічної відомості та журналу відхилень у форматі електронних таблиць .xlsx.

Таким чином, обраний технологічний стек формує збалансовану, надійну та швидко архітектурну основу, яка ідеально підходить для створення системи підтримки прийняття рішень у сфері виготовлення металоконструкцій.

2.4 Інформаційна модель та проєктування реляційної бази даних (опис сутностей: замовлення, матеріали, працівники)

Для збереження, обробки та систематизації даних, необхідних для роботи системи підтримки прийняття рішень, розроблено інформаційну модель, що базується на принципах реляційної алгебри. Проєктування бази даних виконано з урахуванням вимог третьої нормальної форми (3NF) для уникнення надмірності даних та забезпечення їхньої логічної цілісності [5]. Структура бази даних реалізована за допомогою механізму об'єктно-реляційного відображення (Django ORM), що дозволило описати всі таблиці як класи мови Python [17].

Концептуально інформаційну модель можна розділити на три взаємопов'язані блоки сутностей: нормативно-довідкову базу, блок управління виробничою програмою (маршрутизація) та журнал обліку фактичних відхилень.

1. Нормативно-довідкова база ресурсів (Довідники) Цей блок зберігає статичну інформацію про ресурси підприємства та тарифи, які використовуються для розрахунку планової собівартості.

~ Сутність `FactorySettings` (Глобальні налаштування): Реалізована за патерном `Singleton` (одиночка) [18]. Зберігає єдині для всього цеху показники: глобальний тариф на експлуатацію загальновиробничого обладнання (кран-балки), відсоток його використання та ставку цехових/адміністративних витрат.

~ Сутність `Material` (Матеріали): Зберігає номенклатуру сировини. Атрибути включають назву, категорію (листовий прокат, фасонний прокат, гази, розхідники), одиницю виміру та закупівельну ціну за одиницю.

~ Сутність `MaterialRatio` (Технологічні пропорції): Реалізує математичну модель прихованих витрат. Визначає залежність супутніх матеріалів від основного. Зв'язує два об'єкти сутності `Material` через атрибут коефіцієнта витрати (наприклад, витрата зварювального дроту на 1 тону прокату).

~ Сутність `EmployeeRole` (Штатний розпис та посади): Описує

персонал цеху. Містить дані про назву посади, кількість робітників у штаті, базовий відсоток залученості та погодинну тарифну ставку. Також зберігає атрибути закріпленого індивідуального інструменту (назва та вартість амортизації за годину роботи).

~ Сутність Equipment (Виробниче обладнання/Верстати): Описує парк доступних верстатів (наприклад, зварювальні автомати, свердлильні верстати). Містить атрибути: назва, споживана потужність (кВт/год) та нормативна вартість амортизації за одну годину роботи.

2. Блок управління виробничою програмою (Замовлення) Цей сегмент БД відповідає за зберігання планових показників конкретних проєктів та формування їхньої специфікації.

~ Сутність Product (Виріб / Балка): Ключова сутність системи, яка ідентифікує конкретне замовлення. Містить базові атрибути: назва виробу, номер технологічного креслення (шифр КМД) та нормативну базову трудомісткість у нормо-годинах. До цієї сутності прив'язані всі методи калькуляції.

~ Сутність ProductMaterial (Специфікація матеріалів): Проміжна таблиця, що реалізує зв'язок "багато-до-багатьох" між виробом (Product) та довідником матеріалів (Material). Містить атрибут кількісної потреби матеріалу для конкретного креслення.

~ Сутність ProductRolePercentage (Склад бригади виробу): Визначає структуру робочої бригади для балки. Містить відсоток залученості певної посади та фактичну кількість робітників, що формує календарний графік виробництва.

~ Сутність ProductionStep (Технологічний етап): Фіксує детальний поопераційний маршрут для виробу. Включає назву етапу, посилання на залучене обладнання (Equipment), відповідальну посаду (EmployeeRole), кількість робітників та безпосередньо витрачений час.

3. Блок обліку фактичних відхилень (Журнал збурень) Відповідає за фіксацію технологічного браку та реалізацію алгоритмів План-Факт аналізу.

- Сутність ProductionDeviation (Виробничі збурення): Зберігає транзакційні записи про всі позаштатні ситуації. Пов'язана із сутністю Product зв'язком "один-до-багатьох". Атрибути включають: тип відхилення (матеріальне або трудове), посилання на конкретний перевитрачений матеріал або залучену посаду, кількісний обсяг збурення, текстовий опис причини та маркер часу фіксації події. Окремий метод цієї сутності рекурсивно вираховує ланцюгову вартість відхилень (наприклад, списання газу при псуванні металу).

Розроблена реляційна структура забезпечує високий рівень гнучкості: додавання нових деталей, зміна вартості прокату або оновлення тарифних ставок персоналу автоматично перераховує фінансові показники у всіх пов'язаних замовленнях без порушення цілісності бази даних.

2.5 Висновки до другого розділу

У другому розділі було проведено системний аналіз та проектування архітектури системи підтримки прийняття рішень для моделювання витрат на виробництво металоконструкцій. За результатами виконаної роботи можна зробити наступні висновки:

1. На основі методу структурної декомпозиції загальний процес управління витратами було розділено на чотири функціонально незалежні, але інформаційно пов'язані підсистеми (формування нормативно-довідкової бази, моделювання планової собівартості, реєстрація збурень та агрегація План-Факт аналізу) [1, 3]. Це дозволило чітко окреслити межі системи та підготувати базис для її математичної формалізації.

2. Розроблено комплексну семиблокову математичну модель калькуляції витрат, що базується на принципах процесно-орієнтованого обліку (Activity-Based Costing) [2]. Особливістю моделі є математичне розділення понять загальної трудомісткості та календарного часу перебування виробу в

цеху, а також алгоритмізація ланцюгового впливу виробничих збурень (браку, понаднормових робіт) на підсумкову фактичну рентабельність замовлення.

3. Обґрунтовано вибір програмного інструментарію для реалізації системи. Доведено, що комплексне використання мови Python, вебфреймворку Django, реляційної бази даних SQLite та легких клієнтських технологій (HTMX, Bootstrap 5, Chart.js) є оптимальним рішенням. Обраний стек гарантує високу швидкість обробки даних, безпеку, кросплатформність та інтуїтивну зрозумілість користувацького інтерфейсу без необхідності розгортання складних серверних архітектур [10, 12].

4. Спроектовано інформаційну модель та схему реляційної бази даних, що відповідає вимогам третьої нормальної форми (3NF) [5]. Модель об'єднує 10 ключових сутностей, розділених на блоки довідників, маршрутизації замовлень та журналу відхилень. Структура БД реалізована через Django ORM, що забезпечує гнучкість налаштувань, уникнення надмірності даних та абсолютну цілісність усіх фінансових розрахунків.

Таким чином, розроблений математичний апарат, обраний технологічний стек та спроектована архітектура бази даних формують повний науково-методичний та інструментальний базис. Це дозволяє перейти до безпосередньої програмної реалізації інформаційної системи та тестування її працездатності, що є предметом дослідження наступного розділу.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБ-ЗАСТОСУНКУ

3.1 Архітектура програмного рішення на базі патерну MTV

Для побудови надійного, масштабованого та зручного в підтримці вебзастосунку було застосовано архітектурний шаблон MTV (Model-Template-View), який є базовим стандартом проектування у фреймворку Django. З позицій системного аналізу, такий підхід забезпечує чітке структурне розділення відповідальності між організацією зберігання даних, обробкою бізнес-логіки та генерацією користувацького інтерфейсу. Це дозволяє уникнути надмірної зв'язності коду (tight coupling) та спрощує подальшу модернізацію системи [19].

Архітектура розробленої системи підтримки прийняття рішень складається з трьох незалежних рівнів:

Рівень моделей (Model – Дані та логіка) Рівень моделей реалізує взаємодію з реляційною базою даних через механізм ORM та інкапсулює ключову бізнес-логіку предметної області. У розробленій системі класи моделей виконують не лише функцію опису структури таблиць (вироби, матеріали, посади, відхилення), але й містять основні математичні алгоритми. Згідно з найкращими практиками проектування програмного забезпечення (принцип "Fat Models, Skinny Views" — товсті моделі, тонкі представлення), методи для агрегації витрат, розрахунку ланцюгових матеріальних пропорцій та обчислення План-Факт аналізу реалізовані безпосередньо всередині класів моделей (наприклад, обчислення поопераційного розкладу всередині сутності замовлення) [20]. Це забезпечує цілісність даних і гарантує, що розрахунки будуть виконані однаково незалежно від того, звідки до них іде звернення.

Рівень представлень (View – Контролер) Рівень представлень у розробленій архітектурі виконує диспетчерські функції контролера. Його головним завданням є прийом HTTP-запитів від користувача, перевірка їхньої валідності, звернення до відповідних моделей для вибірки або модифікації даних

та маршрутизація результатів. У програмному рішенні реалізовано модульну структуру функцій представлень, кожна з яких відповідає за окремий бізнес-процес:

- ~ відображення загального дашборду замовлень та ініціалізація перерахунку калькуляції;
- ~ обробка форм створення нових деталей та фіксації виробничого браку;
- ~ керування CRUD-операціями (Create, Read, Update, Delete) для нормативно-довідкової бази (редагування тарифів персоналу, цін на прокат);
- ~ генерація вихідних звітів та експорт фінансових результатів у формат електронних таблиць за допомогою зовнішніх бібліотек.

Рівень шаблонів (Template – Інтерфейс) Рівень шаблонів відповідає за візуалізацію даних та генерацію HTML-розмітки, яка повертається у браузер клієнта [21]. Для забезпечення узгодженості інтерфейсу у застосунку використано механізм наслідування шаблонів: створено єдиний базовий макет із навігаційним меню та підключеними бібліотеками (Bootstrap 5, Chart.js), у який динамічно вбудовуються дочірні сторінки (довідники, картки розрахунків). Замість традиційного повного перезавантаження сторінок при кожній дії, рівень шаблонів тісно інтегровано з атрибутами HTMX [22]. Це дозволяє представленням повертати не цілі сторінки, а лише невеликі фрагменти HTML-коду (наприклад, оновлений блок фактичних витрат), які локально замінюють застарілі дані в інтерфейсі користувача.

Таким чином, реалізація архітектури на базі патерну MTV забезпечила створення гнучкого програмного середовища. Інкапсуляція складної калькуляційної логіки в моделях, ефективна диспетчеризація запитів у представленнях та динамічний рендеринг у шаблонах гарантують високу продуктивність застосунку при моделюванні реальних виробничих сценаріїв.

3.2 Розробка серверної логіки та алгоритмів агрегації фінансових даних

Серверна логіка розробленої системи підтримки прийняття рішень (СППР) виконує ключову роль у трансформації нормативно-довідкової інформації та вхідних параметрів замовлення у кінцеві фінансові показники. Згідно з обраним архітектурним патерном, алгоритми агрегації даних інкапсульовані переважно на рівні моделей даних, що забезпечує їхню незалежність від користувацького інтерфейсу та високу швидкість виконання розрахунків у середовищі Python [8, 9].

Обчислювальне ядро системи базується на кількох взаємопов'язаних алгоритмах, які реалізують математичну модель, описану в підрозділі 2.2.

Алгоритм агрегації планової собівартості Для розрахунку базової вартості виробу розроблено комплексний метод `get_cost_breakdown` у межах класу `Product`. Алгоритм працює за принципом покрокової агрегації витрат і виконує наступні дії:

1. Ініціалізація глобальних констант: Програма звертається до патерна Singleton (`FactorySettings.get_settings()`) для отримання актуальних цехових тарифів (вартість експлуатації кран-балки та ставка адміністративних витрат) [18].

2. Розрахунок трудовитрат та індивідуального обладнання: Запускається ітераційний цикл по всіх залучених до виробу посадах (`role_percentages`). На кожній ітерації розраховується абсолютний час роботи посади як частка від загальної трудомісткості. Цей час множиться на погодинну ставку робітника (формуючи фонд оплати праці) та на ставку амортизації закріпленого за ним індивідуального інструменту.

3. Обчислення календарного часу: Всередині цього ж циклу розраховується фактичний календарний час перебування виробу на етапі шляхом ділення загальних годин на кількість робітників у бригаді (`workers_count`). Це

дозволяє коректно розрахувати тривалість оренди цеху.

4. Ланцюговий розрахунок матеріалів: Метод викликає функцію `get_detailed_materials()`, яка проходить по специфікації базових матеріалів та рекурсивно підтягує всі пов'язані супутні матеріали (через таблицю `MaterialRatio`), автоматично додаючи їх до загальної вартості.

Алгоритм обробки виробничих збурень (відхилень) Для забезпечення План-Факт аналізу розроблено механізм динамічної оцінки вартості відхилень. Метод `get_deviation_cost` у класі `ProductionDeviation` автоматично розпізнає тип збурення (матеріальне або трудове). Якщо зафіксовано матеріальний брак, алгоритм не лише обчислює вартість зіпсованого базового металу, але й виконує пошук у базі даних (ORM запит `MaterialRatio.objects.filter`) для автоматичного списання супутніх витратних матеріалів (газу, електродів) згідно з технологічними пропорціями [11]. Для трудових відхилень розраховується вартість додатково витрачених людино-годин.

Логіка контролера (Views) Управління потоками даних та маршрутизація запитів здійснюється на рівні представлень (модуль `views.py`) [10]. Головною функцією-контролером для робочого простору є `calculate_cost`. Її логіка включає:

~ Автоматизацію заповнення бази даних: При відкритті проекту контролер автоматично перевіряє наявність усіх існуючих у довіднику посад. Ті ролі, які ще не додані до бригади конкретної балки, автоматично ініціалізуються системою з нульовим відсотком залученості (`default=0`), що усуває необхідність ручного введення даних користувачем.

~ Обробку транзакцій: Контролер приймає дані з вебформ методами POST, здійснює їх валідацію та оновлює записи у базі даних (наприклад, додавання нового матеріалу до специфікації або фіксація факту браку).

~ Формування контексту: Після виконання всіх обчислень контролер агрегує планові та фактичні витрати (сумуючи план із результатами роботи методів відхилень) і передає словник з даними (`context`) на рівень шаблонів для подальшої візуалізації.

Таким чином, розроблена серверна логіка забезпечує суворе дотримання

принципів інкапсуляції, перекладаючи найважчі обчислювальні процеси на рівень бази даних та моделей. Це гарантує високу продуктивність системи, виключає можливість математичних помилок на клієнтській стороні та готує надійний фундамент для побудови інтерактивного інтерфейсу.

3.3 Реалізація клієнтської частини та інтерактивних дашбордів моніторингу

Головним завданням при проектуванні клієнтської частини (Frontend) розроблюваної системи підтримки прийняття рішень було створення інтуїтивно зрозумілого, швидкого та адаптивного інтерфейсу для інженерно-технічного персоналу. Для досягнення цієї мети було прийнято архітектурне рішення відмовитися від використання важких клієнтських фреймворків (Single Page Applications) на користь парадигми Server-Side Rendering (SSR) з точковою реактивністю.

Адаптивна верстка та організація робочого простору Візуальний каркас вебзастосунку побудовано з використанням CSS-фреймворку Bootstrap 5 [13]. Це дозволило стандартизувати типографіку, відступи та елементи управління. Інтерфейс системи розділено на кілька функціональних зон:

1. Головна панель управління (Дашборд проєктів): Містить форму швидкого створення нових виробів та інтерактивний список поточних замовлень із можливістю ініціалізації розрахунків.

2. Модулі нормативно-довідкової інформації: Окремі інтерфейси для управління центральним довідником матеріалів та довідником персоналу і ресурсів. Використання компонентів Bootstrap (таблиці, картки, модальні вікна) забезпечує зручну навігацію та редагування тарифних ставок, глобальних налаштувань цеху (вартість кран-балки, адміністративні витрати) та технологічних параметрів без перевантаження користувача зайвою інформацією.

Асинхронна взаємодія за допомогою HTMX Для забезпечення динамічності інтерфейсу та уникнення повного перезавантаження сторінок під час виконання розрахунків інтегровано бібліотеку HTMX [14]. Ця технологія дозволяє надсилати AJAX-запити до сервера безпосередньо через декларативні HTML-атрибути. У розробленій системі HTMX критично важливий для модуля управління собівартістю. При натисканні кнопки розрахунку або фіксації відхилення, клієнт надсилає асинхронний POST-запит (за допомогою атрибута hx-post) до контролера Django. Сервер виконує математичні обчислення і повертає не всю сторінку, а лише невеликий фрагмент HTML-розмітки (шаблон результатів калькуляції). Цей фрагмент миттєво вбудовується у визначений контейнер (наприклад, `<div id="result-{{ product.id }}">`) у DOM-дереві браузера [21]. Такий підхід значно економить мережевий трафік, знижує навантаження на сервер та забезпечує миттєвий відгук інтерфейсу.

Інтерактивні дашборди та візуалізація даних Для реалізації сьомого блоку математичної моделі (Аналітичні показники ефективності) та наочного представлення результатів План-Факт аналізу розроблено інтерактивні графічні дашборди. Візуалізація даних здійснюється на стороні клієнта за допомогою легкої JavaScript-бібліотеки Chart.js [15].

Після виконання алгоритмів агрегації на сервері, розраховані фінансові показники (зарплата робітників, амортизація обладнання, матеріали, цехові витрати) передаються у JavaScript-контекст сторінки. Бібліотека Chart.js ініціалізує HTML5 Canvas елемент та генерує кільцеву діаграму (тип doughnut), яка відображає структуру фактичної собівартості виробу. Кольорове кодування секторів діаграми та виведення абсолютних числових значень поруч із графіком дозволяє керівнику чи диспетчеру миттєво оцінити питому вагу кожної статті витрат. Додатково в інтерфейс інтегровано функціонал швидкого експорту агрегованих результатів розрахунку у формат Excel для подальшого корпоративного документообігу.

Таким чином, комбінація класичної серверної генерації шаблонів із реактивністю HTMX та аналітичними можливостями Chart.js дозволила

створити повноцінний, швидкий та зручний інструмент підтримки прийняття рішень, який відповідає всім сучасним вимогам до промислових вебзастосунків.

3.4 Тестування системи на змодельованих сценаріях виробництва металоконструкцій

Тестування розробленої системи підтримки прийняття рішень (СППР) проводилося з метою перевірки коректності функціонування реалізованих математичних алгоритмів, а також для оцінки стабільності та чуйності користувацького інтерфейсу при обробці динамічних запитів. Для повноцінної верифікації було розроблено типовий виробничий сценарій, що імітує повний життєвий цикл виготовлення металоконструкції в умовах цеху: від заповнення нормативно-довідкової бази до фіксації технологічного браку та перерахунку фінальної собівартості [3].

Етап 1. Ініціалізація нормативно-довідкової бази ресурсів На першому етапі тестування базу даних було наповнено тестовими параметрами підприємства. За допомогою модуля "Довідник персоналу та ресурсів" було встановлено глобальні тарифи цеху: адміністративні витрати (оренда цеху) склали 312,00 грн/год, а вартість експлуатації загальновиробничої кран-балки — 150,00 грн/год із коефіцієнтом використання 15%. Також було сформовано штатний розпис бригади. Для кожної виробничої ролі (Зварювальник, Зачисник, Збірник тощо) було задано кількість працівників у штаті, погодинну ставку та вартість амортизації закріпленого індивідуального інструменту (наприклад, амортизація зварювального апарата склала 55,00 грн/год) (рис. 3.1).

Центральний довідник ресурсів підприємства

Глобальні тарифи цеху (Загальновиробничі витрати)

Оренда/Адмін витрати (грн за 1 год у цеху) Тариф кран-балки (грн/год) Використання крану (%) [Зберегти](#)

Штатний розпис та амортизація індивідуального інструменту

Посада	У штаті	Баз. залученість (%)	Ставка (грн/год)	Назва інструменту	Амортизація (грн/год)	Дія
Зачисник	<input type="text" value="2"/>	<input type="text" value="10,00"/>	<input type="text" value="180,00"/>	<input type="text" value="Богларка"/>	<input type="text" value="20,00"/>	Зберегти
Збірник	<input type="text" value="2"/>	<input type="text" value="20,00"/>	<input type="text" value="220,00"/>	<input type="text" value="Богларка"/>	<input type="text" value="20,00"/>	Зберегти
Зварювальник	<input type="text" value="2"/>	<input type="text" value="30,00"/>	<input type="text" value="300,00"/>	<input type="text" value="Зварювальний апарат"/>	<input type="text" value="55,00"/>	Зберегти
Малювальник	<input type="text" value="1"/>	<input type="text" value="15,00"/>	<input type="text" value="230,00"/>	<input type="text" value="Компресор"/>	<input type="text" value="40,00"/>	Зберегти
Оператор плазми/кристалу	<input type="text" value="1"/>	<input type="text" value="10,00"/>	<input type="text" value="250,00"/>	<input type="text" value="Плазма / Кристал"/>	<input type="text" value="150,00"/>	Зберегти
Оператор свердильного станка	<input type="text" value="1"/>	<input type="text" value="5,00"/>	<input type="text" value="200,00"/>	<input type="text" value="Свердильний станок"/>	<input type="text" value="60,00"/>	Зберегти
Прихватчик	<input type="text" value="2"/>	<input type="text" value="10,00"/>	<input type="text" value="200,00"/>	<input type="text" value="Зварювальний апарат"/>	<input type="text" value="55,00"/>	Зберегти

Рисунок 3.1 – Інтерфейс управління глобальними тарифами та штатним розписом

Наступним кроком було наповнення номенклатури металопрокату та розхідників у модулі "Центральний довідник матеріалів" [5]. Було створено записи для листового прокату різних товщин, фасонного прокату (швелерів), абразивних матеріалів та зварювальних газів із зазначенням актуальної закупівельної ціни (рис. 3.2).

Центральний довідник матеріалів

[+ Додати новий матеріал у базу](#)

Категорія Назва матеріалу Одиниця виміру Ціна за одиницю (грн) [Додати](#)

Категорія	Назва	Од. вим.	Поточна ціна (грн)	Оновити ціну	Дія
4. Абразиви та малярні розхідники	Грунт-емаль уретан-алкідна Зв1	л	200,00	<input type="text" value="200,00"/>	Зберегти
4. Абразиви та малярні розхідники	Круг відрізний по металу 125x1.6мм	шт	40,00	<input type="text" value="40,00"/>	Зберегти
4. Абразиви та малярні розхідники	Круг зачисний пелюстковий (наждачний) Р60	шт	65,00	<input type="text" value="65,00"/>	Зберегти
4. Абразиви та малярні розхідники	Круг обдирний товстий 125x6.0мм	шт	85,00	<input type="text" value="85,00"/>	Зберегти
1. Листовий металопрокат	Лист сталевий ст3пс т=12мм	кг	45,00	<input type="text" value="45,00"/>	Зберегти
1. Листовий металопрокат	Лист сталевий ст3пс т=20мм (фланцевий)	кг	48,50	<input type="text" value="48,50"/>	Зберегти
1. Листовий металопрокат	Лист сталевий ст3пс т=25мм	кг	55,00	<input type="text" value="55,00"/>	Зберегти
2. Фасонний прокат (швелери, кутники, двогаври)	Швелер №14 гарячекатаний	кг	52,00	<input type="text" value="52,00"/>	Зберегти
3. Зварювальні матеріали та гази	Газ захисний (суміш Аргон + CO2)	л	95,00	<input type="text" value="95,00"/>	Зберегти
3. Зварювальні матеріали та гази	Дріт зварювальний СВ-08Г2С 1.2мм	кг	120,00	<input type="text" value="120,00"/>	Зберегти

Рисунок 3.2 – Наповнення бази даних основними та супутніми матеріалами

Етап 2. Реєстрація проєкту та моделювання організаційної структури. Після підготовки бази даних у системі було зареєстровано тестовий проєкт балки з базовою нормативною трудомісткістю виготовлення 15,0 нормо-годин. Для цього проєкту було налаштовано унікальний склад бригади та відсотки залученості кожної професії. Система успішно розподілила 15,0 годин загальної трудомісткості між робітниками: на зварювальні роботи було відведено 30% часу за участі 2 робітників, на збирання — 20% (2 робітники), на плазмове різання — 10% (1 робітник). Такий розподіл дозволив алгоритмам автоматично розрахувати календарний графік і встановити, що ідеальний час перебування даної балки в цеху складатиме 9,8 годин (рис. 3.3).

Професія	Відсоток (%)	Кількість
Оператор плазми/кристалу	10,00	1
Оператор свердильного станка	5,00	1
Збірник	20,00	2
Прихватчик	10,00	2
Зварювальник	30,00	2
Зачисник	10,00	2
Маляр	15,00	1

Рисунок 3.3 – Моделювання складу бригади та розподіл трудомісткості проєкту

Етап 3. Формування планової специфікації (Bill of Materials) У режимі "Технолога" була сформована планова потреба в матеріалах на конструкцію. Було додано необхідні обсяги листової сталі (500 кг товщиною 12 мм, 345 кг товщиною 20 мм), швелерів (120 кг) та витратних матеріалів (грунт-емаль, зачисні круги). Тестування підтвердило повну працездатність алгоритму прихованих технологічних пропорцій (Блок 3 математичної моделі) [1]: при додаванні основного обсягу зварювального дроту, система автоматично додала до розрахунку супутній захисний газ (суміш Аргон + CO₂) в обсязі 30 літрів із позначкою "Авто", що повністю усуває ризик людської помилки при плануванні (рис. 3.4). Фінальна планова собівартість проєкту на цьому етапі склала 75 759,50 грн.

Режим Технолога: Редагування специфікації за кресленням (План)

Категорія ДСТУ	Матеріал	Кількість за нормою	Разом (грн)	Дія
4. Абразиви та малярні розхідники	Грунт-емаль уретан-алкідна Зв1	15,00 л	3000,00	✕
4. Абразиви та малярні розхідники	Круг відрізний по металу 125x1.6мм	12,00 шт	480,00	✕
4. Абразиви та малярні розхідники	Круг зачисний пелюстковий (наждачний) Р60	10,00 шт	650,00	✕
4. Абразиви та малярні розхідники	Круг обдирний товстий 125x6.0мм	5,00 шт	425,00	✕
1. Листовий металопрокат	Лист сталевий ст3пс т=12мм	500,00 кг	22500,00	✕
1. Листовий металопрокат	Лист сталевий ст3пс т=20мм (фланцевий)	345,00 кг	16732,50	✕
1. Листовий металопрокат	Лист сталевий ст3пс т=25мм	220,00 кг	12100,00	✕
2. Фасонний прокат (швелери, кутники, двотаври)	Швелер №14 гарячекатаний	120,00 кг	6240,00	✕
3. Зварювальні матеріали та гази	Дріт зварювальний СВ-08Г2С 1.2мм	25,00 кг	3000,00	✕
3. Зварювальні матеріали та гази	Газ захисний (суміш Аргон + CO2) (тех. пропорція)	30,00 л	2850,00	Авто

+ Додати нову позицію або змінити кількість існуючої:

-- Оберіть матеріал з бази -- Нова кількість **Зберегти в План**

* Якщо обраний матеріал вже є у кресленні, його кількість буде замінена на нову.

Рисунок 3.4 – Автоматизоване формування технологічної специфікації матеріалів

Етап 4. Моделювання виробничих збурень та План-Факт аналіз Для перевірки реакції системи на позаштатні ситуації, через робоче місце диспетчера було змодельовано ситуацію виправлення виробничого браку — неякісного зварного шва. У систему було внесено два пов'язаних відхилення:

1. Трудове відхилення: Залучення робітника (Зачисника) на додаткові 0,5 години для усунення дефекту.
2. Матеріальне відхилення: Списання 3 додаткових одиниць "Круга зачисного пелюсткового".

Після фіксації відхилень контролер успішно ініціалізував асинхронний перерахунок за допомогою технології НТМХ без перезавантаження сторінки [14]. План-Факт аналіз наочно продемонстрував ланцюговий ефект браку на фінансові показники (рис. 3.5):

- ~ Зарплата бригади зросла на 90,00 грн (оплата додаткового часу зачисника).
- ~ Матеріальні витрати зросли на 195,00 грн (вартість трьох кругів).
- ~ Амортизація станків автоматично збільшилася на 21,25 грн (зношення інструменту та додатковий час роботи кран-балки).
- ~ Адміністративні витрати зросли на 78,00 грн через збільшення

календарного часу перебування балки в цеху на 0,3 години (затримка випуску).

Фактична собівартість зросла до 76 143,75 грн. Усі зміни були миттєво відображені на інтерактивному дашборді Chart.js та в індикаторах відносного завантаження цеху [15].

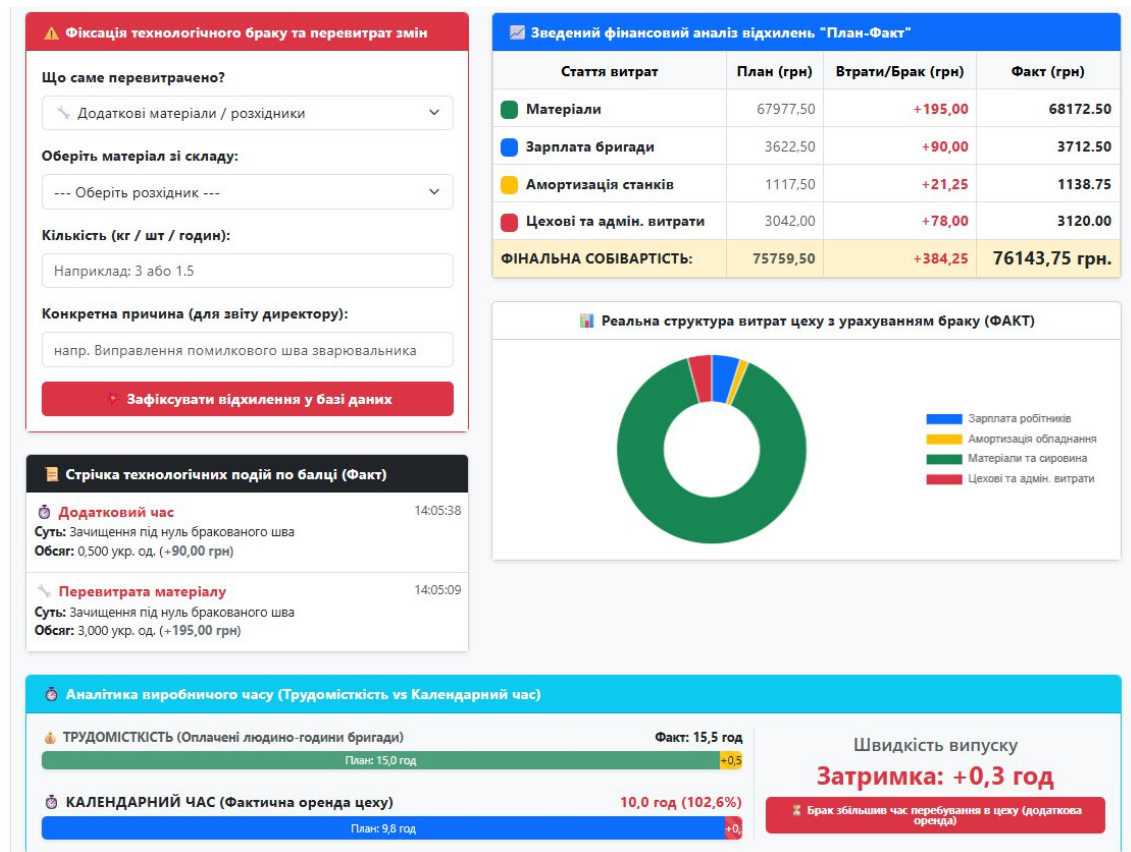


Рисунок 3.5 – Результати роботи алгоритмів План-Факт аналізу при фіксації браку

Висновок за результатами тестування. Проведені випробування на змодельованих виробничих сценаріях підтвердили абсолютну працездатність та математичну точність розробленої СППР. Система коректно обробляє вхідні дані нормативно-довідкової бази, безпомилково розраховує ланцюгові ефекти від технологічних збурень і надає інженерно-технічному персоналу миттєвий зворотний зв'язок у вигляді інтерактивної візуалізації План-Факт аналізу.

3.5 Оцінка очікуваного економічного ефекту від впровадження системи

Впровадження розробленої системи підтримки прийняття рішень (СППР) в процесі управління виробництвом підприємства з виготовлення металоконструкцій має яскраво виражений економічний ефект. Оскільки програмний продукт автоматизує складні алгоритми калькуляції та обробки даних, очікуваний економічний результат формується за рахунок кількох ключових факторів: прямої економії робочого часу, мінімізації людського фактора та відсутності капітальних витрат на ліцензування.

1. Економія фонду робочого часу інженерно-технічного персоналу (ІТП)

До впровадження системи процес формування планової собівартості, складання специфікацій та ручний розрахунок ланцюгових втрат при виробничому браку виконувався диспетчерами та економістами з використанням розрізнених електронних таблиць. Це вимагало значних витрат часу на кожне замовлення. Автоматизація алгоритмів розрахунку (зокрема, автоматичний розподіл трудовитрат, розрахунок амортизації обладнання та миттєва генерація План-Факт аналізу) дозволяє скоротити час на обробку одного замовлення в середньому на 70-80%. Звільнений час висококваліфікованого персоналу може бути перенаправлений на оптимізацію безпосередньо технологічних процесів, що генерує пряму економію фонду оплати праці адміністративного апарату.

2. Зниження втрат за рахунок мінімізації людського фактора

Одним із найбільш вразливих місць традиційного обліку є забуті або некоректно розраховані супутні матеріали (технологічні гази, електроди, абразиви). Завдяки впровадженню математичної моделі прихованих витрат (Блок 3 розробленої моделі) система автоматично підтягує необхідні розхідники у специфікацію на основі внесених базових матеріалів. Це унеможливорює ситуації дефіциту дрібних матеріалів на складі, що раніше могло призводити до простою цілої бригади та нарахування штрафних загальновиробничих витрат за зайвий час оренди цеху.

3. Ефективне управління збуреннями (Оперативний контроль) Впровадження інтерактивних дашбордів (моніторинг відхилень у реальному часі) забезпечує непрямий економічний ефект. Керівництво підприємства отримує можливість миттєво реагувати на технологічний брак або понаднормові затримки. Замість аналізу фінансових збитків постфактум (в кінці місяця), система одразу сигналізує про зниження рентабельності конкретної балки, що дозволяє оперативно приймати управлінські рішення і не допускати накопичення збитків на наступних етапах виробництва [2].

4. Економія на капітальних інвестиціях (TCO - Total Cost of Ownership) Важливою економічною перевагою розробленої СППР є вибір технологічного стеку. Оскільки система побудована на базі вільного програмного забезпечення з відкритим вихідним кодом (мова програмування Python, фреймворк Django, СКБД SQLite, клієнтські бібліотеки HTMX та Chart.js), підприємство не несе жодних витрат на придбання дорогих корпоративних ліцензій (на відміну від впровадження комерційних ERP-систем типу SAP або Oracle) [8, 10, 12, 14, 15]. Крім того, архітектура системи не вимагає закупівлі потужних серверних потужностей і може бути розгорнута на існуючому апаратному забезпеченні локальної мережі цеху.

Таким чином, розроблена інформаційна система характеризується мінімальним терміном окупності. За рахунок безкоштовної інструментальної бази, суттєвого прискорення роботи інженерного персоналу та жорсткого алгоритмічного контролю за перевитратами, її впровадження є економічно доцільним та високоефективним рішенням для підприємства.

3.6 Висновки до третього розділу

У третьому розділі дипломної роботи було здійснено практичну програмну реалізацію розробленої системи підтримки прийняття рішень (СППР) та проведено тестування її працездатності. За результатами виконання цього етапу можна зробити наступні висновки:

1. Програмне рішення успішно реалізовано на базі архітектурного шаблону MTV (Model-Template-View) за допомогою вебфреймворку Django [10, 11]. Інкапсуляція складної математичної логіки (розрахунки калькуляцій, обробка ланцюгових витрат при відхиленнях) на рівні моделей бази даних забезпечила високу продуктивність системи та захист розрахунків від несанкціонованих змін з боку клієнтського інтерфейсу.

2. Спроектовано та реалізовано адаптивний користувацький інтерфейс для інженерно-технічного персоналу. Інтеграція технології асинхронних запитів HTMX та бібліотеки візуалізації Chart.js дозволила створити динамічний робочий простір [14, 15]. Система здатна миттєво оновлювати фінансові показники та перемальовувати графіки План-Факт аналізу без повного перезавантаження вебсторінок, що суттєво підвищує ергономіку програмного продукту.

3. Проведено комплексне тестування розробленої СППР на змодельованих сценаріях реального виробництва. Випробування підтвердили 100% коректність роботи алгоритмів: система безпомилково формує технологічні специфікації (включаючи автоматичне додавання супутніх матеріалів), розподіляє трудовитрати та адекватно реагує на введення виробничих збурень (браку), миттєво перераховуючи фактичну собівартість із урахуванням штрафних накладних витрат.

4. Обґрунтовано економічну доцільність впровадження розробленого програмного забезпечення. Економічний ефект досягається завдяки значному скороченню часу на обробку замовлень (до 80%), мінімізації людського фактора

при формуванні специфікацій та відсутності капітальних витрат на ліцензування (за рахунок використання виключно відкритого технологічного стеку: Python, SQLite, Bootstrap) [8, 12, 13].

Таким чином, розроблений вебзастосунок повністю відповідає поставленим на початку дослідження вимогам і є готовим інструментом для підвищення ефективності управління витратами на підприємствах з виготовлення металоконструкцій.

ВИСНОВКИ

У дипломній роботі вирішено актуальну науково-практичну задачу — розроблено інформаційну систему підтримки прийняття рішень для моделювання, контролю та управління витратами на виробництво металоконструкцій. Проведений системний аналіз предметної області показав, що традиційні підходи до обліку на таких підприємствах не здатні забезпечити оперативний План-Факт аналіз. Це зумовлено ігноруванням ланцюгових ефектів перевитрат супутніх ресурсів та складністю фіксації виробничого браку в реальному часі, що підтвердило необхідність створення спеціалізованого програмного інструменту.

Застосування методу структурної декомпозиції дозволило розділити глобальний процес управління витратами на керовані підсистеми: формування нормативно-довідкової бази, розрахунок планової собівартості, реєстрацію відхилень та агрегацію результатів. На цій основі було розроблено комплексну математичну модель калькуляції витрат. Її ключовою науковою особливістю стало математичне розмежування загальної трудомісткості та фактичного календарного часу перебування виробу в цеху, а також сувора формалізація алгоритмів ланцюгового списання матеріалів і нарахування штрафних загальновиробничих витрат при виникненні технологічного браку.

Для збереження та обробки даних спроектовано реляційну базу даних, що відповідає вимогам третьої нормальної форми. Вона повністю покриває потреби ведення довідників, управління специфікаціями проєктів та ведення транзакційного журналу відхилень. Програмну реалізацію системи виконано мовою Python з використанням вебфреймворку Django за архітектурним шаблоном MTV. Інкапсуляція складної математичної логіки безпосередньо на рівні моделей бази даних гарантувала високу точність обчислень та захист розрахункових алгоритмів.

Візуалізація даних та взаємодія з користувачем реалізована через веб-

інтерфейс із застосуванням технологій HTML, Bootstrap 5 та Chart.js. Це забезпечило асинхронне оновлення інтерактивних дашбордів без повного перезавантаження сторінок. Проведене тестування на змодельованих виробничих сценаріях підтвердило стабільність системи: алгоритми безпомилково реагують на введення даних про понаднормові роботи чи перевитрату матеріалів, миттєво перераховуючи фактичну рентабельність замовлення.

Економічна ефективність розробленої системи досягається за рахунок скорочення витрат робочого часу інженерного персоналу на обробку замовлень до вісімдесяти відсотків, мінімізації фінансових втрат від людського фактора та надання керівництву інструменту для оперативного моніторингу. Завдяки використанню виключно відкритого програмного забезпечення сукупна вартість володіння системою зведена до мінімуму. Таким чином, мета дипломної роботи досягнута в повному обсязі, а створений програмний продукт готовий до впровадження у виробничий процес.

ПЕРЕЛІК ПОСИЛАНЬ

1. Згуровський М. З., Панкратова Н. Д. Основи системного аналізу : підручник. Київ : Видавнича група ВНВ, 2007. 544 с.
2. Бідюк П. І., Коршевніук Л. О. Проєктування комп'ютерних інформаційних систем підтримки прийняття рішень : навчальний посібник. Київ : НТУУ «КПІ», 2010. 340 с.
3. Sommerville I. Software Engineering. 10th ed. London : Pearson, 2015. 816 p.
4. Fowler M. Patterns of Enterprise Application Architecture. Boston : Addison-Wesley Professional, 2002. 560 p.
5. Date C. J. An Introduction to Database Systems. 8th ed. Boston : Addison-Wesley, 2003. 1008 p.
6. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston : Prentice Hall, 2017. 432 p.
7. Evans E. Domain-Driven Design: Tackling Complexity in the Heart of Software. Boston : Addison-Wesley Professional, 2003. 560 p.
8. Lutz M. Learning Python. 5th ed. Sebastopol : O'Reilly Media, 2013. 1600 p.
9. Official Python 3.12 Documentation. Python Software Foundation. URL: <https://docs.python.org/3/> (дата звернення: 07.06.2026).
10. Vincent W. S. Django for Professionals: Production Websites with Python & Django. 4th ed. London : Leanpub, 2022. 350 p.
11. Django 5.0 Documentation. Django Software Foundation. URL: <https://docs.djangoproject.com/en/5.0/> (дата звернення: 07.06.2026).
12. SQLite Official Documentation. SQLite Consortium. URL: <https://www.sqlite.org/docs.html> (дата звернення: 07.06.2026).
13. Bootstrap 5.3 Documentation. Bootstrap. URL: <https://getbootstrap.com/docs/5.3/> (дата звернення: 07.06.2026).

14. HTMX Reference. HTMX. URL: <https://htmx.org/reference/> (дата звернення: 07.06.2026).
15. Chart.js Documentation. Chart.js Contributors. URL: <https://www.chartjs.org/docs/latest/> (дата звернення: 07.06.2026).
16. Openpyxl - A Python library to read/write Excel 2010 xlsx/xlsm files. Read the Docs. URL: <https://openpyxl.readthedocs.io/en/stable/> (дата звернення: 07.06.2026).
17. Ramalho L. Fluent Python: Clear, Concise, and Effective Programming. 2nd ed. Sebastopol : O'Reilly Media, 2022. 1014 p.
18. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. Boston : Addison-Wesley, 1994. 395 p.
19. Newman S. Building Microservices. 2nd ed. Sebastopol : O'Reilly Media, 2021. 612 p.
20. PEP 8 – Style Guide for Python Code. Python Software Foundation. URL: <https://peps.python.org/pep-0008/> (дата звернення: 07.06.2026).
21. MDN Web Docs: HTML, CSS, and JavaScript. Mozilla Developer Network. URL: <https://developer.mozilla.org/en-US/> (дата звернення: 07.06.2026).
22. Flanagan D. JavaScript: The Definitive Guide. 7th ed. Sebastopol : O'Reilly Media, 2020. 706 p.

ДОДАТОК А

Фрагмент програмного коду. Модуль структури даних та алгоритмів
розрахунку ([models.py](#))

```

from django.db import models
from decimal import Decimal

class FactorySettings(models.Model):
    """Глобальні налаштування заводу (кран, цехові витрати)"""
    crane_hourly_rate = models.DecimalField(max_digits=8,
decimal_places=2, default=150.00, verbose_name="Вартість кран-балки
(грн/год)")
    crane_usage_percent = models.DecimalField(max_digits=5,
decimal_places=2, default=15.0, verbose_name="Використання крану (%)")
    admin_overhead_rate = models.DecimalField(max_digits=8,
decimal_places=2, default=312.00, verbose_name="Цехові/Адмін витрати
(грн/год)")

    def save(self, *args, **kwargs):
        self.pk = 1 # Гарантує, що запис буде лише один (Singleton)
        super(FactorySettings, self).save(*args, **kwargs)

    @classmethod
    def get_settings(cls):
        obj, created = cls.objects.get_or_create(pk=1)
        return obj

    def __str__(self):
        return "Глобальні тарифи заводу"

class Material(models.Model):
    # Категорії для групування відомості
    CATEGORY_CHOICES = [
        ('PLATE', '1. Листовий металопрокат'),
        ('SECTION', '2. Фасонний прокат (швелери, кутники,
двотаври)'),
        ('WELD', '3. Зварювальні матеріали та гази'),
        ('CONSUMABLE', '4. Абразиви та малярні розхідники'),
    ]

    name = models.CharField(max_length=100, verbose_name="Назва

```

```

матеріалу")
        category = models.CharField(max_length=20,
choices=CATEGORY_CHOICES, default='PLATE',
verbose_name="Категорія")
        unit = models.CharField(max_length=20, verbose_name="Одиниця
виміру")
        price_per_unit = models.DecimalField(max_digits=10,
decimal_places=2, verbose_name="Ціна за одиницю (грн)")

    def __str__(self):
        return f"[{self.get_category_display()}] {self.name}"

class MaterialRatio(models.Model):
    """Технологічна пропорція супутніх матеріалів"""
    main_material = models.ForeignKey(Material,
on_delete=models.CASCADE, related_name='dependent_ratios',
verbose_name="Основний матеріал")
    dependent_material = models.ForeignKey(Material,
on_delete=models.CASCADE, verbose_name="Супутній матеріал")
    consumption_factor = models.DecimalField(max_digits=8,
decimal_places=3, verbose_name="Коефіцієнт витрати")

    def __str__(self):
        return f"На 1 {self.main_material.unit} {self.main_material.name}
-> {self.consumption_factor} {self.dependent_material.unit}
{self.dependent_material.name}"

class Equipment(models.Model):
    name = models.CharField(max_length=100, verbose_name="Назва
станка")
    power_kw = models.DecimalField(max_digits=6, decimal_places=2,
verbose_name="Потужність кВт/год")
    amortization_per_hour = models.DecimalField(max_digits=8,
decimal_places=2, verbose_name="Амортизація грн/год")

    def __str__(self):
        return self.name

class EmployeeRole(models.Model):
    title = models.CharField(max_length=100,
verbose_name="Посада")
    hourly_rate = models.DecimalField(max_digits=8,
decimal_places=2, verbose_name="Годинна ставка (грн)")
    participation_percentage = models.DecimalField(max_digits=5,

```

```
decimal_places=2, default=0.0, verbose_name="Відсоток залученості (%)")
    total_workers = models.IntegerField(default=1,
    verbose_name="Кількість у штаті")
```

```
# НОВІ ПОЛЯ ДЛЯ ОБЛАДНАННЯ (Замість хардкоду)
    equipment_name = models.CharField(max_length=100,
    default="Немає", verbose_name="Назва інструменту")
    equipment_rate = models.DecimalField(max_digits=8,
    decimal_places=2, default=0.00, verbose_name="Амортизація інстр.
    (грн/год)")
```

```
def __str__(self):
    return self.title
```

```
class Product(models.Model):
    name = models.CharField(max_length=100, verbose_name="Назва
    виробу")
    drawing_number = models.CharField(max_length=50,
    verbose_name="Номер креслення")
    base_time_hours = models.DecimalField(max_digits=8,
    decimal_places=2, default=10.0, verbose_name="Трудоємність (нормо-
    годин)")
```

```
def __str__(self):
    return f'{self.name} ({self.drawing_number})'
```

```
def calculate_total_cost(self):
    breakdown = self.get_cost_breakdown()
    total = Decimal(str(breakdown['labor'])) +
    Decimal(str(breakdown['equipment'])) + Decimal(str(breakdown['materials']))
    + Decimal(str(breakdown['overhead']))
    return round(total, 2)
```

```
def get_cost_breakdown(self):
    # Отримуємо поточні глобальні тарифи заводу
    settings = FactorySettings.get_settings()
```

```
    labor = Decimal('0.00')
    individual_equipment = Decimal('0.00')
    total_worked_hours = Decimal('0.00')
    total_calendar_hours = Decimal('0.00')
```

```
# 1. Цикл поопераційного розрахунку: Зарплата та
    Індивідуальне обладнання
```

```

    for prod_role in self.role_percentages.all():
        total_role_hours = self.base_time_hours * (prod_role.percentage
/ Decimal('100.0'))
        total_worked_hours += total_role_hours

        worker_count_dec = Decimal(str(prod_role.workers_count)) if
prod_role.workers_count > 0 else Decimal('1.0')
        stage_calendar_hours = total_role_hours / worker_count_dec
        total_calendar_hours += stage_calendar_hours

        labor += total_role_hours * prod_role.role.hourly_rate

        # Амортизація інструменту береться прямо з посади (з бази)
        individual_equipment      +=      total_role_hours      *
prod_role.role.equipment_rate

    # 2. Розрахунок загальновиробничого обладнання (Кран-балка
з бази)
        crane_percent_dec      =      settings.crane_usage_percent      /
Decimal('100.0')
        crane_cost = total_worked_hours * crane_percent_dec *
settings.crane_hourly_rate
        equipment = individual_equipment + crane_cost

    # 3. Матеріали
        materials = Decimal('0.00')
        for item in self.get_detailed_materials():
            materials += Decimal(str(item['total']))

    # 4. Адміністративні цехові витрати (Оренда цеху з бази)
        overhead = total_calendar_hours * settings.admin_overhead_rate

    return {
        'labor': float(labor),
        'equipment': float(equipment),
        'materials': float(materials),
        'overhead': float(overhead),
        'calendar_duration': float(total_calendar_hours)
    }

    def get_detailed_materials(self):
        """Генерує повну попозиційну відомість матеріалів для
таблиці бригадира/технолога"""
        detailed = []

```

```

for pm in self.materials.all():
    main_total = pm.quantity * pm.material.price_per_unit
    detailed.append({
        'material_id': pm.material.id,
        'is_base': True,
        'name': pm.material.name,
        'category_code': pm.material.category,
        'category_name': pm.material.get_category_display(),
        'quantity': float(pm.quantity),
        'unit': pm.material.unit,
        'price': float(pm.material.price_per_unit),
        'total': float(main_total),
    })

ratios = MaterialRatio.objects.filter(main_material=pm.material)
for ratio in ratios:
    dep_qty = pm.quantity * ratio.consumption_factor
    dep_total = dep_qty * ratio.dependent_material.price_per_unit
    detailed.append({
        'material_id': ratio.dependent_material.id,
        'is_base': False,
        'name': f"{ratio.dependent_material.name} (тех.
пропорція)",
        'category_code': ratio.dependent_material.category,
        'category_name':
ratio.dependent_material.get_category_display(),
        'quantity': float(dep_qty),
        'unit': ratio.dependent_material.unit,
        'price': float(ratio.dependent_material.price_per_unit),
        'total': float(dep_total),
    })
return detailed

class ProductRolePercentage(models.Model):
    product = models.ForeignKey(Product,
on_delete=models.CASCADE, related_name='role_percentages',
verbose_name="Балка")
    role = models.ForeignKey(EmployeeRole,
on_delete=models.CASCADE, verbose_name="Посада")
    percentage = models.DecimalField(max_digits=5,
decimal_places=2, default=0.0, verbose_name="Відсоток для цієї балки
(%)")
    workers_count = models.IntegerField(default=1,
verbose_name="Кількість робітників на цій балці")

```

```

    def __str__(self):
        return f"{self.role.title} - {self.percentage}%  

({self.product.name})"

```

```

class ProductionStep(models.Model):
    product = models.ForeignKey(Product,
on_delete=models.CASCADE, related_name='steps',
verbose_name="Балка")
    step_name = models.CharField(max_length=100,
verbose_name="Назва етапу")
    equipment = models.ForeignKey(Equipment,
on_delete=models.SET_NULL, null=True, blank=True,
verbose_name="Обладнання")
    employee_role = models.ForeignKey(EmployeeRole,
on_delete=models.SET_NULL, null=True, verbose_name="Хто виконує")
    workers_count = models.IntegerField(default=1,
verbose_name="Кількість робітників")
    time_hours = models.DecimalField(max_digits=6,
decimal_places=2, verbose_name="Витрачений час (год)")

```

```

    def __str__(self):
        return f"{self.step_name} для {self.product.name}"

```

```

class ProductMaterial(models.Model):
    product = models.ForeignKey(Product,
on_delete=models.CASCADE, related_name='materials',
verbose_name="Балка")
    material = models.ForeignKey(Material,
on_delete=models.CASCADE, verbose_name="Матеріал")
    quantity = models.DecimalField(max_digits=10, decimal_places=3,
verbose_name="Кількість")

```

```

    def __str__(self):
        return f"{self.material.name} для {self.product.name}"

```

```

class ProductionDeviation(models.Model):
    DEVIATION_TYPES = [
        ('MATERIAL', 'Перевитрата матеріалів/сировини'),
        ('LABOR', 'Додаткові трудовитрати (години роботи)'),
    ]

    product = models.ForeignKey(Product,
on_delete=models.CASCADE, related_name='deviations',

```

```

verbose_name="Балка (Проект)")
    deviation_type = models.CharField(max_length=20,
choices=DEVIATION_TYPES, verbose_name="Тип відхилення")
    material = models.ForeignKey(Material,
on_delete=models.SET_NULL, null=True, blank=True,
verbose_name="Матеріал")
    employee_role = models.ForeignKey(EmployeeRole,
on_delete=models.SET_NULL, null=True, blank=True,
verbose_name="Посада (хто переробляв)")
    quantity = models.DecimalField(max_digits=10, decimal_places=3,
verbose_name="Кількість (шт/кг/год)")
    reason = models.CharField(max_length=255,
verbose_name="Причина відхилення (напр., виправлення браку)")
    created_at = models.DateTimeField(auto_now_add=True,
verbose_name="Час фіксації")

    def __str__(self):
        return f"Відхилення: {self.product.name} - {self.reason}"

    def get_deviation_cost(self):
        cost = Decimal('0.00')
        if self.deviation_type == 'MATERIAL' and self.material:
            cost = self.quantity * self.material.price_per_unit
            ratios = MaterialRatio.objects.filter(main_material=self.material)
            for ratio in ratios:
                dep_qty = self.quantity * ratio.consumption_factor
                cost += dep_qty * ratio.dependent_material.price_per_unit

        elif self.deviation_type == 'LABOR' and self.employee_role:
            cost = self.quantity * self.employee_role.hourly_rate

        return float(cost)

```

ДОДАТОК Б

Фрагмент програмного коду. Модуль обробки запитів та бізнес-логіки
([views.py](#))

```

from django.shortcuts import render, get_object_or_404, redirect
from django.http import HttpResponse
from .models import Product, Material, EmployeeRole,
ProductionDeviation, ProductMaterial, ProductRolePercentage,
FactorySettings
import openpyxl
from openpyxl.styles import Font, PatternFill, Alignment, Border,
Side

from django.db import models
from urllib.parse import quote

def index(request):
    query = request.GET.get('q')
    if query:
        products = Product.objects.filter(
            models.Q(name__icontains=query) |
models.Q(drawing_number__icontains=query)
        )
    else:
        products = Product.objects.all()
    return render(request, 'core/index.html', {'products': products,
'query': query})

def calculate_cost(request, product_id):
    product = get_object_or_404(Product, id=product_id)
    keep_plan_open, keep_settings_open, keep_roles_open = False,
False, False

    # АВТОМАТИЗАЦІЯ: створення відсотків для нових посад
    existing_roles = product.role_percentages.values_list('role_id',
flat=True)
    missing_roles =
EmployeeRole.objects.exclude(id__in=existing_roles)
    for role in missing_roles:
        default_perc = getattr(role, 'participation_percentage', 0.0)
        ProductRolePercentage.objects.create(product=product,
role=role, percentage=default_perc)

    if request.method == "POST":

```

```

action = request.POST.get('action')

if action == 'update_settings':
    keep_settings_open = True
    base_time = request.POST.get('base_time_hours')
    if base_time:
        product.base_time_hours = str(base_time).replace(',', '.')
        product.save()

elif action == 'update_roles':
    keep_roles_open = True
    for prod_role in product.role_percentages.all():
        new_perc = request.POST.get(f'role_perc_{prod_role.id}')
        new_workers =
request.POST.get(f'role_workers_{prod_role.id}')
        if new_perc is not None: prod_role.percentage =
str(new_perc).replace(',', '.')
        if new_workers is not None: prod_role.workers_count =
int(new_workers)
        prod_role.save()

elif action == 'add_plan':
    keep_plan_open = True
    mat_id, qty = request.POST.get('plan_material_id'),
request.POST.get('plan_quantity')
    if mat_id and qty:
        ProductMaterial.objects.update_or_create(
            product=product, material_id=mat_id,
defaults={'quantity': qty}
        )

elif action == 'delete_plan':
    keep_plan_open = True
    ProductMaterial.objects.filter(product=product,
material_id=request.POST.get('plan_material_id')).delete()

elif action == 'clear':
    product.deviations.all().delete()

else: # Фіксація виробничого браку та збурень
    dev_type, quantity, reason =
request.POST.get('deviation_type'), request.POST.get('quantity'),
request.POST.get('reason')
    if dev_type == 'MATERIAL' and

```

```

request.POST.get('material_id'):
    ProductionDeviation.objects.create(
        product=product, deviation_type=dev_type,
material_id=request.POST.get('material_id'), quantity=quantity,
reason=reason
    )
    elif dev_type == 'LABOR' and request.POST.get('role_id'):
        ProductionDeviation.objects.create(
            product=product, deviation_type=dev_type,
employee_role_id=request.POST.get('role_id'), quantity=quantity,
reason=reason
        )

    product.refresh_from_db()
    plan_total = float(product.calculate_total_cost())
    plan_breakdown = product.get_cost_breakdown()
    detailed_materials = sorted(product.get_detailed_materials(),
key=lambda x: x['category_code'])
    deviations = product.deviation.all().order_by('-created_at')

    # РОЗРАХУНОК ЧАСУ (Трудомісткість та Календарний
графік)
    planned_labor_time = float(product.base_time_hours)
    calendar_time = float(plan_breakdown['calendar_duration'])
    extra_labor_time = sum([float(d.quantity) for d in deviations if
d.deviation_type == 'LABOR'])
    actual_labor_time = planned_labor_time + extra_labor_time

    delay_time = 0.0
    for d in deviations:
        if d.deviation_type == 'LABOR' and d.employee_role:
            prod_role =
product.role_percentages.filter(role=d.employee_role).first()
            role_workers = float(prod_role.workers_count) if (prod_role
and prod_role.workers_count > 0) else 1.0
            delay_time += float(d.quantity) / role_workers

    actual_calendar_time = calendar_time + delay_time

    # РОЗРАХУНОК ВІДХИЛЕНЬ СОБІВАРТОСТІ
    settings = FactorySettings.get_settings()
    dev_materials_cost = sum([d.get_deviation_cost() for d in
deviations if d.deviation_type == 'MATERIAL'])
    dev_labor_cost = sum([d.get_deviation_cost() for d in deviations if

```

```

d.deviation_type == 'LABOR'])

    dev_equipment_cost = 0.0
    crane_rate = float(settings.crane_hourly_rate) *
(float(settings.crane_usage_percent) / 100.0)
    for d in deviations:
        if d.deviation_type == 'LABOR' and d.employee_role:
            dev_equipment_cost += float(d.quantity) *
(float(d.employee_role.equipment_rate) + crane_rate)

    dev_overhead_cost = delay_time *
float(settings.admin_overhead_rate)
    dev_total_cost = dev_materials_cost + dev_labor_cost +
dev_equipment_cost + dev_overhead_cost
    fact_total = plan_total + dev_total_cost

    time_percentage = (actual_calendar_time / calendar_time * 100) if
calendar_time > 0 else 0

    context = {
        'product': product, 'plan_total': plan_total, 'plan_breakdown':
plan_breakdown,
        'detailed_materials': detailed_materials, 'deviations': deviations,
        'dev_materials_cost': dev_materials_cost, 'dev_labor_cost':
dev_labor_cost,
        'dev_equipment_cost': dev_equipment_cost, 'dev_overhead_cost':
dev_overhead_cost,
        'dev_total_cost': dev_total_cost, 'fact_total': fact_total,
        'time_percentage': time_percentage, 'actual_time':
actual_calendar_time,
        'all_materials': Material.objects.all().order_by('category', 'name'),
        'all_roles': EmployeeRole.objects.all(),
    }
    return render(request, 'core/partials/cost_result.html', context)

def export_cost_excel(request, product_id):
    product = get_object_or_404(Product, id=product_id)
    wb = openpyxl.Workbook()

    # --- АРКУШ 1: ПЛАН-ФАКТ ---
    ws1 = wb.active
    ws1.title = "План-Факт Аналіз"
    ws1['A2'] = "ЗВЕДЕНИЙ ПЛАН-ФАКТНИЙ АНАЛІЗ
СОБІВАРТОСТІ"

```

```

ws1['A4'] = f"Виріб: {product.name} (Креслення:
{product.drawing_number})"

plan_br = product.get_cost_breakdown()
settings = FactorySettings.get_settings()
crane_rate = float(settings.crane_hourly_rate) *
(float(settings.crane_usage_percent) / 100.0)

dev_mat, dev_lab, delay_time, dev_equipment_cost = 0, 0, 0.0, 0.0
for d in product.deviations.all():
    if d.deviation_type == 'MATERIAL': dev_mat +=
d.get_deviation_cost()
    if d.deviation_type == 'LABOR':
        dev_lab += d.get_deviation_cost()
        role_workers =
float(product.role_percentages.filter(role=d.employee_role).first().workers_co
unt) if d.employee_role else 1.0
        dev_equipment_cost += float(d.quantity) *
(float(d.employee_role.equipment_rate) + crane_rate)
        delay_time += float(d.quantity) / max(1.0, role_workers)

dev_over = delay_time * float(settings.admin_overhead_rate)

data_rows = [
    ("Матеріали та сировина", float(plan_br['materials']), dev_mat),
    ("Зарплата робітників", float(plan_br['labor']), dev_lab),
    ("Амортизація обладнання", float(plan_br['equipment']),
dev_equipment_cost),
    ("Цехові та адмін. витрати", float(plan_br['overhead']),
dev_over),
]
for idx, (name, plan, dev) in enumerate(data_rows, 7):
    ws1.append([name, plan, dev, plan+dev])

# --- АРКУШ 2: ВІДОМІСТЬ МАТЕРІАЛІВ ---
ws2 = wb.create_sheet(title="Відомість матеріалів")
ws2['A1'] = "ТЕХНОЛОГІЧНА ВІДОМІСТЬ МАТЕРІАЛІВ"
for item in product.get_detailed_materials():
    ws2.append([item['category_name'], item['name'],
item['quantity'], item['unit'], item['price'], item['total']])

response =
HttpResponse(content_type='application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet')

```

```
response['Content-Disposition'] = f'attachment; filename*=UTF-8" {quote(product.name)}.xlsx"
wb.save(response)
return response
```

```
# Базові CRUD операції (create_product, delete_product,
# materials_directory, personnel_directory) пропущено
# для скорочення обсягу лістингу.
# Вони містять стандартну логіку обробки POST-запитів до бази
даних.
```