

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

«Вебдодаток для складання кошторису»

Виконала студентка групи ІПЗк-23-1
ГОРБАЧЕВСЬКА Діана Андріївна

Керівник роботи:
ГОДЛЕВСЬКИЙ Юрій Олександрович

Рецензент:
МОРОЗОВ Андрій Васильович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення

Тетяна Вакалюк

«26» лютого 2025 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ГОРБАЧЕВСЬКА Діана Андріївна**

Керівник роботи: **ГОДЛЕВСЬКИЙ Юрій Олександрович**

Тема роботи: **«Вебдодаток для складання кошторису»,**

затверджена наказом закладу вищої освіти від **«14» лютого 2025 р., №61/с**

Вихідні дані для роботи: архітектура, інтерфейс та функціональні можливості вебдодатку для створення будівельних кошторисів, зокрема механізми формування проєктів, управління ресурсами, автоматизований розрахунок вартості, експорт документації, а також процес організації взаємодії користувачів і адміністраторів із системою за допомогою сучасних вебтехнологій.

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Годлевський Ю.О.	27.02.2025р.	27.02.2025р.
2	Годлевський Ю.О.	27.02.2025р.	27.02.2025р.
3	Годлевський Ю.О.	27.02.2025р.	27.02.2025р.

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з вебдодатку для складання та управління кошторисами, а також пояснювальної записки. Пояснювальна записка до випускної роботи містить 101 сторінку, 53 ілюстрації та 23 таблиці.

Мета роботи – автоматизація процесу складання кошторисів, підвищення точності фінансових розрахунків та оптимізація управління ресурсами для покращення ефективності планування проєктів.

Методи дослідження включають аналіз існуючих програмних рішень, проєктування та реалізацію модульної монолітної архітектури backend-частини з використанням Node.js та Express.js, frontend-розробку на основі React з бібліотекою Ant Design, використання реляційної бази даних PostgreSQL з ORM Sequelize, а також інтеграцію функцій експорту у формати Excel та PDF. Додатково реалізовано калькулятор для розрахунку фарбування труб та модуль користувацьких формул з підтримкою математичних виразів.

Результати – створено ефективний вебдодаток, який автоматизує розрахунки вартості матеріалів, робочої сили та загальних витрат, забезпечує зручний інтерфейс для створення, редагування, експорту та аналізу кошторисів, а також надає функції для точних інженерних обчислень.

Рекомендації щодо використання – система може бути впроваджена у будівельних компаніях, проєктних бюро та організаціях, що займаються фінансовим плануванням, для підвищення продуктивності, зменшення кількості помилок і покращення управління проєктними ресурсами.

КЛЮЧОВІ СЛОВА: КОШТОРИС, АВТОМАТИЗАЦІЯ РОЗРАХУНКІВ, УПРАВЛІННЯ РЕСУРСАМИ, ФІНАНСОВИЙ ЗВІТ, ПРОЕКТНЕ ПЛАНУВАННЯ, ВЕБДОДАТОК, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

					ДУ «Житомирська політехніка».25.121.8.000 – ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Горбачевська Д.А.			Веб-додаток для складання кошторисів	Літ.	Арк.	Аркушів	
Керівник.		Годлевський Ю.О.					3	42	
						ФІКТ Гр. ІПЗк-23-1[1]			
Н. контр.									
Зав. каф.									

ABSTRACT

The bachelor's qualification thesis consists of a web application for creating and managing cost estimates, as well as an explanatory report. The explanatory report contains 101 pages, 53 illustrations, and 23 tables.

The aim of the thesis is to automate the process of estimate generation, improve the accuracy of financial calculations, and optimize resource management to enhance the efficiency of project planning.

Research methods include the analysis of existing software solutions; the design and implementation of a modular monolithic backend architecture using Node.js and Express.js; frontend development based on React with the Ant Design library; the use of a PostgreSQL relational database with the Sequelize ORM; and integration of export features to Excel and PDF formats. Additionally, a calculator for pipe painting calculations and a custom formula module with mathematical expression support were implemented.

Results – an efficient web application has been developed that automates the calculation of material, labor, and general costs; provides a user-friendly interface for creating, editing, exporting, and analyzing estimates; and supports precise engineering calculations.

Recommendations for use – the system can be implemented in construction companies, design bureaus, and organizations involved in financial planning to improve productivity, reduce errors, and enhance project resource management.

KEYWORDS: ESTIMATE, AUTOMATED CALCULATIONS, RESOURCE MANAGEMENT, FINANCIAL REPORT, PROJECT PLANNING, WEB APPLICATION, SOFTWARE.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
ПЕРЕЛІК УМОВНИХ ЗНАЧЕНЬ	6
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ВЕБДОДАТКУ ДЛЯ СКЛАДАННЯ КОШТОРИСІВ.....	8
1.1 ПОСТАНОВКА ЗАДАЧІ	8
1.2 АНАЛІЗ АНАЛОГІВ ПРОГРАМНОГО ПРОДУКТУ	10
1.3 ВИБІР АРХІТЕКТУРИ У ВЕБДОДАТКУ ДЛЯ СКЛАДАННЯ КОШТОРИСІВ.....	16
1.4. ОБҐРУНТУВАННЯ ВИБОРУ ІНСТРУМЕНТАЛЬНИХ ЗАСОБІВ ТА ВИМОГИ ДО АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ.....	19
Висновки до першого розділу	22
РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБДОДАТКУ ДЛЯ СКЛАДАННЯ КОШТОРИСІВ.....	23
2.1 Визначення варіантів використання та об’єктно-орієнтованої структури СИСТЕМИ	23
2.2 Розробка бази даних системи.....	37
2.3 ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ АЛГОРИТМІВ РОБОТИ СИСТЕМИ	41
2.4 РЕАЛІЗАЦІЯ ФУНКЦІОНАЛУ ВЕБДОДАТКУ ДЛЯ СКЛАДАННЯ КОШТОРИСУ	49
Висновки до другого розділу	58
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБДОДАТКОМ ДЛЯ СКЛАДАННЯ КОШТОРИСІВ.....	60
3.1 Порядок встановлення та налаштування параметрів системи	60
3.2 СТРУКТУРА ІНТЕРФЕЙСУ ВЕБДОДАТКУ ДЛЯ СКЛАДАННЯ КОШТОРИСУ	62
3.3 ТЕСТУВАННЯ РОБОТИ ПРОГРАМНОГО КОМПЛЕКСУ	76
Висновки до третього розділу	78
ВИСНОВКИ	79
ДЖЕРЕЛА ІНФОРМАЦІЇ.....	80
ДОДАТКИ	82

ПЕРЕЛІК УМОВНИХ ЗНАЧЕНЬ

API – прикладний програмний інтерфейс

БД – база даних

ПЗ – програмне забезпечення

СУБД – система управління базами даних

SDK – набір для розробки програмного забезпечення (software development kit)

MVC – схема розділення даних додатків і керуючої логіки на три окремі компоненти Model-View-Controller

CRUD – основні операції, що використовуються для управління даними (Create , Read , Update , Delete)

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

Актуальність теми полягає в необхідності створення ефективного програмного забезпечення для складання кошторисів, що особливо важливо в умовах післявоєнного відновлення країни. Значна частина інфраструктури, житлових і промислових об'єктів потребує реконструкції, що зумовлює постійний попит на точні та зручні у використанні кошторисні інструменти. Програмний продукт дозволяє проводити розрахунки вартості будівництва, ремонту, очищення територій від завалів та інших відновлювальних робіт. Висока актуальність теми зумовлена також потребою в цифровій трансформації будівельної галузі.

Метою випускної кваліфікаційної роботи бакалавра є створення сучасного вебзастосунку, який автоматизує процеси розробки, аналізу та управління кошторисною документацією в будівництві та суміжних сферах.

Для досягнення мети проєкту насамперед потрібно було визначити ключові функції, які будуть корисними користувачам, що працюють із кошторисами. На основі цього аналізу обиралися найефективніші технології для створення продуктивного й зручного вебдодатку. Особливу увагу приділено архітектурі системи — вона мала бути масштабованою, відповідати сучасним стандартам і бути зручною для підтримки. Для збереження всієї необхідної інформації створено продуману базу даних, що охоплює дані про користувачів, проєкти та фінансові розрахунки. Також реалізовано об'єктно-орієнтовану модель, яка впорядковує логіку системи та взаємозв'язки її компонентів.

Завданням є створити вебдодаток для автоматизації кошторисних розрахунків шляхом аналізу потреб користувачів, вибору технологій, проєктування архітектури, розробки бази даних і побудови об'єктно-орієнтованої моделі системи.

Об'єктом дослідження є процес автоматизації кошторисних розрахунків у сфері будівництва та інфраструктурного відновлення.

Предметом дослідження є розробка програмного забезпечення для складання кошторисів, яке дозволяє оптимізувати оцінку вартості будівництва, ремонтних робіт та очищення територій.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ВЕБДОДАТКУ ДЛЯ СКЛАДАННЯ КОШТОРИСІВ

1.1 Постановка задачі

Однією з ключових проблем у сфері будівництва та ремонту є складність процесу складання кошторисів. Традиційні методи ведення кошторисної документації часто включають ручні розрахунки, що призводить до високої ймовірності помилок, неточностей у підрахунках та значних часових витрат. Крім того, відсутність централізованої системи управління кошторисами ускладнює їх редагування та оновлення, особливо при роботі з великими обсягами даних.

Неефективне управління кошторисною документацією у сфері будівництва породжує низку суттєвих проблем, що впливають як на якість проєктного планування, так і на фінансову стабільність виконавців робіт. Однією з основних труднощів є часті помилки під час розрахунків вартості матеріалів, обсягу трудозатрат та залучення механізмів, що може призводити до суттєвих перевитрат бюджету або недооцінки реальної вартості виконання робіт. У випадках, коли ціни на будівельні матеріали змінюються практично щотижня, особливо в умовах нестабільної економічної ситуації, процес оновлення даних у кошторисах стає складним і трудомістким, що затримує ухвалення рішень та корегування проєктів.

Інтеграція з іншими цифровими інструментами також стає складнішою, особливо коли йдеться про обмін даними з бухгалтерськими або аналітичними системами, які використовуються на підприємствах. Зокрема, часто виникають труднощі з експортом кошторисної інформації у стандартизовані формати, такі як Excel або PDF, що є необхідними для подальшої обробки, звітності чи передачі в зовнішні облікові програми. Відсутність підтримки таких форматів або обмежені можливості експорту значно ускладнюють робочий процес, змушуючи користувачів вручну переносити дані, що збільшує ризик помилок і втрату даних.

Ще однією слабкою ланкою сучасних кошторисних процесів є відсутність вбудованих спеціалізованих калькуляторів для оперативного обрахунку окремих статей витрат. Для деяких видів робіт — таких як фарбування труб, що потребує

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

специфічних формул та інженерних підходів — користувачам доводиться вдаватися до сторонніх інструментів або виконувати обрахунки вручну, що знову ж таки створює простір для помилок і знижує продуктивність праці.

Усі ці фактори вказують на нагальну потребу у впровадженні сучасного програмного забезпечення, яке здатне централізовано вирішити згадані проблеми та забезпечити прозоре, гнучке і швидке управління кошторисною інформацією в реальному часі. Таке рішення має підтримувати автоматизацію розрахунків, інтеграцію з іншими системами підприємства та зручний інтерфейс для користувачів з різними рівнями технічної підготовки.

Для подолання вищезгаданих проблем необхідно створити сучасну інформаційну систему, яка забезпечить повноцінну автоматизацію процесу складання та редагування кошторисів. Такий підхід дозволить значно зменшити кількість помилок завдяки впровадженню цифрових розрахунків та вбудованих алгоритмів верифікації даних. Крім того, інтеграція функціоналу експорту до формату Excel дозволить використовувати сформовані дані у зовнішніх системах та підвищить зручність звітності. Окрему увагу слід приділити реалізації вбудованого калькулятора, здатного швидко обчислювати витрати матеріалів та трудозатрати, зокрема для специфічних задач, що потребують точних інженерних розрахунків.

Обґрунтування методів вирішення.

Для досягнення поставлених цілей у процесі розробки програмного забезпечення передбачається використання низки сучасних методів та підходів. Зокрема, застосовуватимуться методи автоматизації, що забезпечать високоточний розрахунок кошторисної вартості із врахуванням динаміки цін та ресурсних обмежень. Для підвищення ефективності управління фінансами буде впроваджено оптимізаційні алгоритми, які дозволять аналізувати витрати та виявляти можливості для їх коригування. Важливу роль у реалізації функціоналу відіграватимуть веб-технології, які забезпечать створення зручного, інтерактивного інтерфейсу для повноцінного управління кошторисами з будь-якого пристрою та в будь-якому середовищі.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

Для реалізації даного проекту будуть використані такі засоби (Табл. 1.1):

Таблиця 1.1

Компонент	Засіб / Технологія
Backend	
Платформа	Node.js
Фреймворк	Express.js
Архітектура	Модульна монолітна
База даних	PostgreSQL
Мова запитів до БД	SQL
ORM	Sequelize
Аутентифікація / Авторизація	JWT (JSON Web Tokens)
Тестування	Postman
Експорт	ExcelJS, PDFKit, Puppeteer
Обробка математичних формул	Math.js
Frontend	
Бібліотека	React
Мова	JavaScript
Система авторизації	JWT
Компонентна бібліотека	Ant Design
Маршрутизація	React Router
Зовнішні бібліотеки	Axios

Розробка даного програмного забезпечення дозволить значно спростити та пришвидшити процес управління кошторисами, підвищити точність розрахунків та забезпечити зручність роботи.

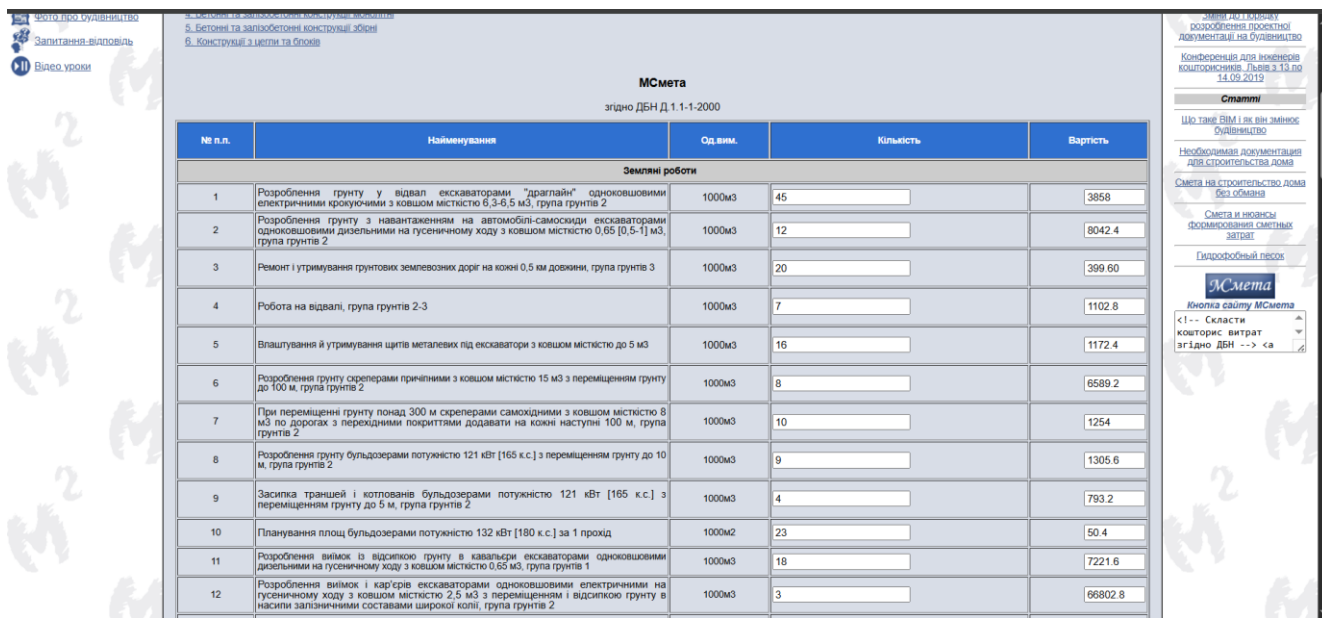
1.2 Аналіз аналогів програмного продукту

У процесі розробки ефективної системи для складання кошторисів важливо враховувати досвід існуючих програмних рішень у цій галузі. Аналіз аналогів дає змогу не лише виявити успішні підходи до реалізації функціоналу, але й

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

зрозуміти недоліки, які варто уникнути при створенні власного програмного забезпечення.

Одним із поширених рішень є онлайн-сервіс «Скласти кошторис» (Рис. 1.1) на платформі Meta є прикладом максимально спрощеного вебінструменту, орієнтованого на одноразове формування кошторисної таблиці без реєстрації або додаткових сервісів.



№ п.п.	Найменування	Од.вим.	Кількість	Вартість
1	Розроблення ґрунту у відвал екскаваторами "драглайн" одноковшовими електричними крокуючими з ковшем місткістю 6,3-6,5 м ³ , група ґрунтів 2	1000м ³	45	3858
2	Розроблення ґрунту з навантаженнями на автомобіль-самоскиди екскаваторами одноковшовими дисковими на пусиничному ході з ковшем місткістю 0,65 [0,5-1] м ³ , група ґрунтів 2	1000м ³	12	8042.4
3	Ремонт і утримування ґрунтових землевозних доріг на кожен 0,5 км довжини, група ґрунтів 3	1000м ³	20	399.60
4	Робота на відвалі, група ґрунтів 2-3	1000м ³	7	1102.8
5	Влаштування й утримування цитів металевих під екскаватори з ковшем місткістю до 5 м ³	1000м ³	16	1172.4
6	Розроблення ґрунту скреперами причіпними з ковшем місткістю 15 м ³ з переміщенням ґрунту до 100 м, група ґрунтів 2	1000м ³	8	6589.2
7	При переміщенні ґрунту понад 300 м скреперами самохідними з ковшем місткістю 8 м ³ по дорожках з періодичними покріттями додавати на кожен наступні 100 м, група ґрунтів 2	1000м ³	10	1254
8	Розроблення ґрунту бульдозерами потужністю 121 кВт [165 к.с.] з переміщенням ґрунту до 10 м, група ґрунтів 2	1000м ³	9	1305.6
9	Засипка траншей і котлованів бульдозерами потужністю 121 кВт [165 к.с.] з переміщенням ґрунту до 5 м, група ґрунтів 2	1000м ³	4	793.2
10	Планування площ бульдозерами потужністю 132 кВт [180 к.с.] за 1 прохід	1000м ²	23	50.4
11	Розроблення вишок із відсікою ґрунту в кавальєри екскаваторами одноковшовими дисковими на пусиничному ході з ковшем місткістю 0,65 м ³ , група ґрунтів 1	1000м ³	18	7221.6
12	Розроблення вишок і кар'єрів екскаваторами одноковшовими електричними на пусиничному ході з ковшем місткістю 2,5 м ³ з переміщенням і відсікою ґрунту в напісні залізними складами широкої колії, група ґрунтів 2	1000м ³	3	66802.8

Рис. 1.1 Онлайн-сервіс «Скласти кошторис» початок таблиці [1]

Основна мета цього ресурсу — дозволити користувачеві швидко порахувати загальну вартість виконаних робіт або послуг, ввівши відповідні дані у стандартну таблицю прямо в браузері.

Функціональність сервісу є мінімальною. Користувач має змогу ввести вручну назви позицій, їх кількість, ціну за одиницю та одиниці виміру. На основі цих значень система обраховує загальну суму по кожній позиції, а також підсумкову суму всього кошторису (Рис. 1.2).

Це відбувається в режимі реального часу, без додаткових підтверджень чи збережень. Жодної складної логіки, умовних обчислень або збереження введених даних на сервері не передбачено.

Окремим блоком формується службова інформація: назва об'єкта, замовник, виконавець, дата тощо. Однак ця інформація також вводиться лише один раз у межах поточної сесії. Після оновлення сторінки або випадкового закриття браузера всі дані буде втрачено — функція автоматичного або ручного збереження відсутня.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		
						11

13	Улаштування хвіртки з установленням стовпів бетонних	шт		543.6
14	Установлення в спорудах кутових лотків перерізом до 0,2 м2	м3		3270
15	Улаштування в житлових будинках механічних перегородок з одношаровою облицовкою цементно-стружковими плитами товщиною 10 мм по дерев'яному каркасу з ізоляційною прокладкою	100м2		27804
Конструкції з цегли та блоків				
1	Улаштування фундаментів стрічкових із бутового каменю	м3		583.2
2	Улаштування основи під фундаменти цеглені	м3		273.6
3	Гідроізоляція стін, фундаментів горизонтальна облицювання в 2 шари	100м2		5852.4
4	Мурування зовнішніх середньої складності стін з цегли керамичної при висоті поверху до 4 м	м3		984
5	Мурування перегородок армованих з цегли керамичної товщиною в 1/2 цегли при висоті поверху до 4 м	100м2		18205.2
6	Мурування з цегли керамичної стовпів прямокутних неармованих при висоті поверху понад 4 м	м3		1124.4
7	Розширення швів мурування з цегли	100м2		1520.4
8	Мурування з великих вапнякових блоків масою до 0,5 т	м3		662.4
9	Чисте мурування стін із каменів вапнякових при висоті поверху понад 4 м	м3		636
10	Установлення перегородок із гіпсових плит товщиною 100 мм в 2 шари при висоті поверху до 4 м	100м2		14055.6
11	Укладання підлогових залізобетонних плит із мозаїчним покриттям	100м2		10482
Всього: 716322				грн.
				Підрахувати Очистити

Рис. 1.2 Онлайн-сервіс «Скласти кошторис» вигляд таблиці знизу

На відміну від більш функціональних сервісів, Meta не надає можливості експортувати сформований кошторис у вигляді PDF-документа або в інших цифрових форматах (наприклад, Excel). Користувач може лише роздрукувати вміст сторінки браузера за допомогою стандартної команди «Друк», що, однак, не гарантує якісного вигляду готового документа. Немає жодної системи шаблонів або попереднього перегляду результату в форматі, придатному для офіційного використання.

Також повністю відсутні такі важливі для професійного використання елементи, як облік клієнтів, історія замовлень, повторне використання даних, збереження або управління шаблонами робіт. Інтеграція з платіжними платформами, системами аналітики або можливість відправки кошторису по посиланню теж відсутні.

Зважаючи на це, Meta «Скласти кошторис» не можна розглядати як повноцінний вебдодаток. Це радше статична вебсторінка із калькулятором, що імітує структуру кошторису. Вона придатна лише для найпростіших завдань, які не потребують збереження історії, співпраці з клієнтами або подальшої автоматизації документообігу.

Наступним аналогом є OneB Invoice — це сучасна SaaS-платформа, яка пропонує комплексне рішення для створення рахунків, інвойсів, комерційних пропозицій та кошторисів. Основна мета сервісу — надати підприємцям, фрілансерам та малому бізнесу інструмент, який поєднує зручність, функціональність і автоматизацію процесів, пов'язаних з фінансовими документами.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		12

Рис. 1.3 Платформа «OneV Invoice» форма для створення кошторису [2]

Головна особливість OneV — це інтуїтивно зрозумілий інтерфейс, у якому користувач може створити інвойс буквально за кілька хвилин (Рис. 1.3). У формі доступно заповнення реквізитів компанії, логотипу, дати, номеру документа, клієнтських даних та позицій: назви товарів або послуг, ціни, податків, знижок тощо. Позиції можна зберігати до каталогу, щоб повторно використовувати в інших документах(Рис. 1.4).

Рис. 1.4 Платформа «OneV Invoice» форма для створення кошторису(продовження)

Крім базових функцій, платформа має розширені можливості: документ можна не лише завантажити у форматі PDF, але й надіслати клієнтові по електронній пошті або створити унікальне публічне посилання (Рис. 1.5). Завдяки цьому, процес

передачі документів значно спрощується. Особливо зручно, що за посиланням клієнт бачить документ у вебформаті, а також має змогу одразу сплатити рахунок.

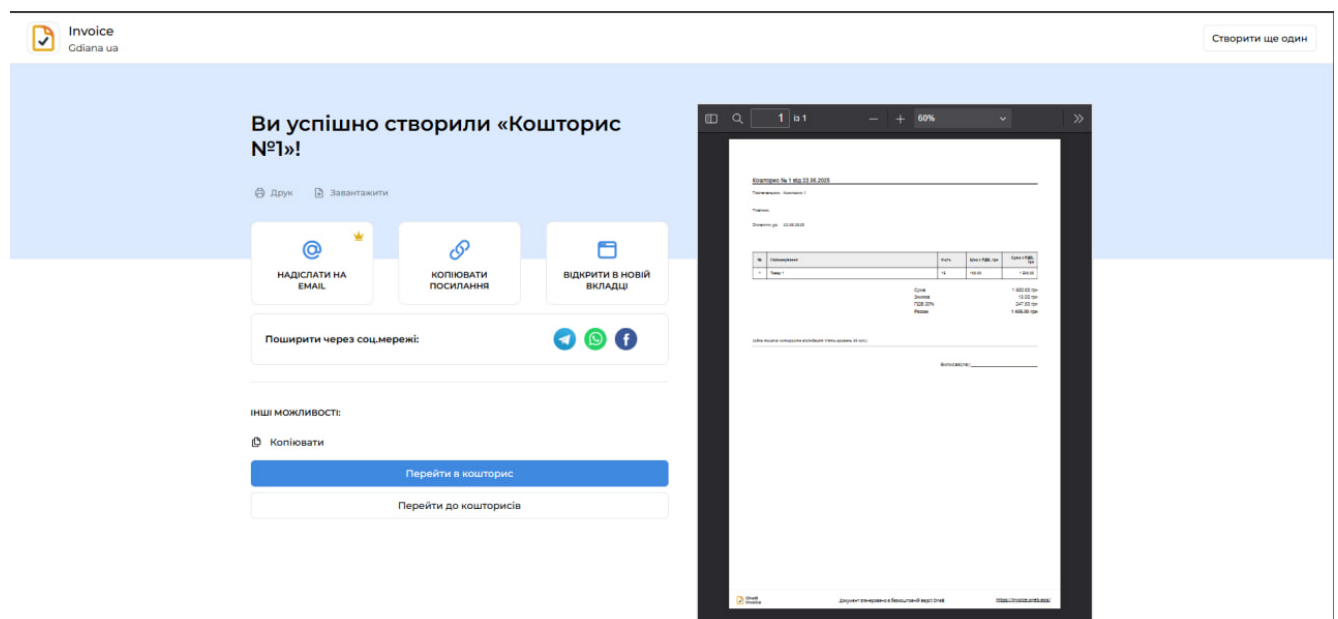


Рис. 1.5 Платформа «OneV Invoice» отриманий кошторис з розширеними можливостями

Сервіс підтримує інтеграцію з популярними платіжними платформами, такими як LiqPay, PayPal, MonoPlata, а також можливість оплати через банківські реквізити (IBAN) з автоматичним генеруванням QR-коду для оплати. Це робить OneV Invoice не просто генератором документів, а повноцінним інструментом комунікації та взаємодії з клієнтом у режимі онлайн.

Ще однією сильною стороною є аналітика. Користувач бачить статус документів (оплачено, в очікуванні, протерміновано тощо), може налаштувати нагадування для клієнтів, отримувати push-повідомлення. Для користувачів, які працюють з клієнтами з різних країн, доступна мультимовна підтримка: англійська, українська, польська тощо.

OneV Invoice працює на умовах безкоштовного періоду з подальшою оплатою за розширений функціонал. Загалом цей сервіс виглядає як високоякісний аналог професійного CRM-рішення, який орієнтований не лише на створення, а й на повний цикл взаємодії з клієнтом — від кошторису до оплати й аналітики.

		Горбачевська Д.А.			ДУ «Житомирська політехніка». 25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		14

Оцінка наведених рішень дозволяє сформувати уявлення про ключові функціональні можливості, які мають бути реалізовані в сучасному програмному продукті для складання кошторисів. Водночас аналіз недоліків виявляє ті аспекти, на які слід звернути особливу увагу при розробці власної системи. Для кращої наочності результати порівняння основних характеристик аналогічних програмних продуктів подано в таблиці нижче.

Таблиця 1.2

Характеристика	Онлайн-сервіс «Скласти кошторис»	Платформа «OneB Invoice»	Вебдодаток для складання кошторисів
Створення кошторисів	+	+	+
Обрахунок загальної вартості позицій	+	+	+
Введення довільних позицій (назва, ціна, кількість)	+	+	+
Каталог товарів/послуг з можливістю вибору	-	+	+
Управління клієнтами/замовниками	-	+	+
Збереження історії створених кошторисів	-	+	+
Формування документу для друку та експорт в PDF	+/-	+	+
Експорт в Excel	-	-	+
Онлайн оплата (IBAN, QR-код, LiqPay, PayPal, тощо)	-	+	-
Мультимовна підтримка	-	+	-
Аутентифікація та управління користувачами (адмін-панель)	-	+	+
Калькулятор спеціалізованих розрахунків (фарбування труб тощо)	-	-	+
Можливість створення та збереження власних математичних формул	-	-	+
Збереження формул у боковій панелі для повторного використання	-	-	+
Редагування та оновлення вже створених кошторисів	-	+	+
Ролі користувачів (адмін, користувач)	-	+	+
Доступ без реєстрації	+	-	-

Переглянувши аналоги програмного продукту для складання кошторисів, можна виокремити основні відмінності між ними. Онлайн-сервіс «Скласти кошторис» є надзвичайно простим інструментом із мінімальним функціоналом — він дозволяє лише вручну заповнювати таблицю позицій з автоматичним підрахунком загальної вартості, без можливості збереження, повторного використання чи експорту у структурованому вигляді. Платформа «OneB Invoice» має розширений функціонал, включаючи збереження клієнтів, онлайн-оплату та генерацію PDF-файлів, однак орієнтована переважно на створення інвойсів, а не на комплексну роботу з будівельними чи технічними кошторисами.

Враховуючи результати аналізу аналогів, було прийнято рішення створити власний вебдодаток, який буде зосереджений саме на роботі з кошторисною документацією. На відміну від наявних рішень, він підтримує створення та збереження кошторисів з можливістю редагування, включає систему користувачів із поділом ролей (адміністратор/користувач), а також надає унікальні функції, серед яких — вбудований калькулятор технічних розрахунків (наприклад, фарбування труб) і модулів для створення та повторного використання власних математичних формул.

Таким чином, запропонований вебдодаток має бути зручним у користуванні, орієнтованим на практичні потреби будівельної сфери, підтримувати персоналізацію розрахунків, а також забезпечувати сучасний рівень керування даними за допомогою авторизації, збереження історії та адміністрування. Це дозволить повністю задовольнити запити цільової аудиторії й перевершити наявні рішення за функціональністю та гнучкістю.

1.3 Вибір архітектури у вебдодатку для складання кошторисів

Розробка вебдодатку для складання кошторисів базується на клієнт-серверній архітектурі, яка забезпечує чітке розділення між логікою клієнтської частини (інтерфейс користувача) та серверною частиною (обробка даних, авторизація, генерація документів тощо). Такий підхід дозволяє досягти масштабованості, модульності та зручності супроводу проєкту.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		16

Серверна частина реалізована за допомогою платформи Node.js у поєднанні з фреймворком Express.js, який забезпечує просту та гнучку розробку REST API. Архітектура проєкту є модульною монолітною, що дозволяє зберігати логіку в ізольованих функціональних модулях і водночас спрощує розгортання.

Для взаємодії з базою даних використовується PostgreSQL — потужна реляційна СУБД із відкритим вихідним кодом, яка відзначається стабільністю, підтримкою SQL-стандарту, високою продуктивністю та великою кількістю додаткових модулів. Робота з базою даних на рівні коду реалізована через ORM-бібліотеку Sequelize, що дозволяє будувати SQL-запити у вигляді JavaScript-структур, а також забезпечує автоматичне мігрування схем, валідацію, зв'язки між таблицями та інші корисні інструменти для ефективної розробки.

Для реалізації аутентифікації та авторизації застосовується технологія JWT (JSON Web Tokens), яка дозволяє надійно зберігати токени доступу на клієнтській стороні та захищати маршрути, забезпечуючи конфіденційність даних. Це дозволяє ефективно розмежовувати права доступу між користувачами різних ролей та обмежувати доступ до важливих ресурсів.

Окрему увагу приділено реалізації експорту даних. Для створення документів у форматі Excel використовується бібліотека ExcelJS, а для генерації PDF — PDFKit у поєднанні з Puppeteer, що дозволяє зберігати сторінки у форматі PDF з повною підтримкою стилів і макетів. Такий підхід забезпечує збереження структури та дизайну документа при друку або передачі у зовнішні системи.

Для підтримки унікального функціоналу — створення й обчислення власних математичних формул — інтегровано бібліотеку Math.js. Вона дозволяє інтерпретувати користувацькі вирази, проводити складні математичні обрахунки та створювати динамічні розрахункові модулі, зокрема калькулятор фарбування труб та інші інженерні задачі, характерні для кошторисної документації. Це значно розширює можливості системи та робить її зручною для технічних спеціалістів, що потребують точних і гнучких обчислень.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

На фронтенді використовується бібліотека React, яка забезпечує розробку компонентного, динамічного та масштабованого інтерфейсу. Реалізація маршрутизації виконується з використанням React Router, що дає змогу будувати зручну структуру сторінок. Для візуальної частини застосовується UI-бібліотека Ant Design, яка забезпечує привабливий вигляд інтерфейсу та підтримує велику кількість готових компонентів.

Клієнтська частина вебдодатку реалізована за допомогою мови програмування JavaScript, яка є стандартною для роботи у браузері. Саме завдяки JavaScript стає можливою динамічна взаємодія з елементами інтерфейсу — наприклад, оновлення вмісту сторінки без повного перезавантаження, валідація форм у реальному часі та миттєве реагування на дії користувача. Для обміну даними з серверною частиною використовується бібліотека Axios — зручний інструмент для надсилання HTTP-запитів. Вона забезпечує просту інтеграцію з API, дозволяє налаштовувати заголовки запитів (наприклад, для передачі токенів авторизації), обробляти помилки та оптимізувати роботу з асинхронними запитами. Використання Axios сприяє зручнішій побудові логіки взаємодії з бекендом і покращує загальну читабельність коду.

Для перевірки працездатності API вебдодатку використовується Postman — спеціалізований інструмент для тестування серверних запитів. За його допомогою розробник може вручну сформулювати різноманітні запити (GET, POST, PUT, DELETE), задати параметри, тіла запитів та заголовки, після чого миттєво переглянути відповіді від сервера. Це дозволяє виявляти помилки, тестувати логіку обробки даних, перевіряти систему на надійність та відповідність очікуваній поведінці. Регулярне використання Postman значно полегшує підтримку і розвиток проєкту, особливо при внесенні змін у backend або інтеграції нових функцій.

Таким чином, вибір інструментів був зумовлений необхідністю створення швидкого, надійного та розширюваного вебдодатку з підтримкою зручного інтерфейсу, обробкою складних розрахунків і можливістю генерування кошторисної документації. Всі обрані технології є сучасними, мають активну спільноту, документацію та добре масштабуються під потреби реального бізнесу.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

1.4. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

У процесі створення вебдодатку для складання кошторисів було обрано перевірений і сучасний стек технологій, що дозволяє забезпечити стабільність, безпечну обробку даних, а також гнучкість у розширенні функціоналу відповідно до зростаючих потреб користувачів. Основною вимогою до архітектури системи була можливість централізованого керування кошторисами, обліку користувачів, математичних обчислень і експорту документів у зручному форматі.

Архітектурно система реалізована як модульний моноліт, що дає змогу ефективно підтримувати кодову базу, легко впроваджувати нові модулі без необхідності масштабування в окремі мікросервіси на ранніх етапах. Цей підхід дозволяє підтримувати внутрішню структурованість, гнучке розділення відповідальності між частинами системи, а також дає змогу в майбутньому масштабувати її до мікросервісної моделі за потреби.

На серверному боці обрана платформа Node.js, яка забезпечує асинхронне виконання операцій та високу продуктивність при роботі з великою кількістю одночасних запитів. Для побудови REST API використовується Express.js — мінімалістичний фреймворк, що надає повну свободу у формуванні структури проєкту, при цьому підтримує інтеграцію сторонніх middleware-компонентів. Уся логіка роботи з базою даних реалізується за допомогою ORM Sequelize, що забезпечує зручну роботу з таблицями, валідацію, створення зв'язків між сутностями та міграції.

На рівні збереження інформації використовується PostgreSQL — стабільна та надійна СУБД, яка чудово підходить для структурування складних кошторисних даних, зв'язків між користувачами, розрахунковими формулами та шаблонами. Крім того, PostgreSQL легко розширюється за допомогою вбудованих типів даних і дозволяє виконувати складні SQL-запити високої продуктивності.

Важливою частиною функціоналу вебдодатку є обчислення технічних формул, зокрема — у спеціалізованому калькуляторі фарбування труб, який використовується для інженерних розрахунків. Для реалізації цієї можливості було інтег-

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

ровано бібліотеку Math.js, що підтримує парсинг (синтаксичний розбір) та обчислення користувацьких виразів безпосередньо в браузері. Це забезпечує гнучкість — користувачі можуть створювати власні математичні моделі з довільною кількістю змінних, проводити розрахунки у зручному візуальному середовищі, а також зберігати створені формули у бічній панелі для повторного використання у майбутньому. Такий підхід дозволяє автоматизувати специфічні розрахунки, які часто зустрічаються у практиці кошторисування, і уникнути постійного ручного введення.

На клієнтському боці застосовується React — сучасна JavaScript-бібліотека, що дозволяє створювати ефективні, динамічні та інтерактивні інтерфейси користувача на основі компонентно-орієнтованої архітектури. Це дозволяє розробникам створювати повторно використовувані блоки UI, легко оновлювати частину контенту без повного перезавантаження сторінки, а також покращує продуктивність роботи застосунку. Для організації навігації використовується React Router — бібліотека, що реалізує маршрутизацію у межах односторінкової архітектури (SPA), дозволяючи користувачам перемикатися між різними функціональними модулями (наприклад, створення кошторису, управління формулами, перегляд звітів) без втрати стану сторінки. З метою створення привабливого й зручного дизайну у проєкт інтегровано бібліотеку Ant Design, яка надає велику кількість готових, стилізованих компонентів — таких як кнопки, форми, модальні вікна, таблиці тощо — що дозволяє значно пришвидшити розробку та забезпечити послідовність інтерфейсу.

Для забезпечення захисту персональних даних та контрольованого доступу до системи впроваджена авторизація на базі JWT (JSON Web Tokens). Користувачі мають відповідні ролі (адмін або звичайний користувач), що регулює рівень доступу до функцій системи. Всі запити до захищених маршрутів обробляються із перевіркою токенів, що запобігає несанкціонованому доступу.

Додатково в систему інтегровано функціонал експорту документів у форматі PDF та Excel, що є критично важливим для подальшого використання кошторисної документації в офіційному або діловому обігу. На серверній частині для

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

цього застосовуються бібліотеки PDFKit та Puppeteer: перша відповідає за побудову структури PDF-документів, а друга — за рендеринг сторінки з урахуванням усіх стилів, макетів і навіть динамічного контенту, що забезпечує точну відповідність вигляду документів оригінальному вебінтерфейсу. Для створення Excel-документів використовується ExcelJS, яка дозволяє будувати таблиці з підтримкою кольорів, шрифтів, меж, форматів чисел і навіть формул, що робить результуючі файли максимально адаптованими до вимог бухгалтерських чи інженерних стандартів.

Для підтримки стабільної якості розробки застосовується регулярне тестування API за допомогою Postman — інструменту, що дозволяє створювати, зберігати та виконувати HTTP-запити до серверної частини. Це дає змогу перевіряти не лише коректність отриманих відповідей, але й поведінку системи при помилках, неправильних параметрах або неавторизованому доступі. Такий підхід сприяє швидкому виявленню дефектів, полегшує налагодження та забезпечує стабільність функціоналу при впровадженні нових змін або оновлень.

Вимоги до апаратного забезпечення залежать від масштабу розгортання системи — тестове середовище, внутрішнє використання або продакшн для великої кількості користувачів. У базовій конфігурації для розробницького або локального тестування достатньо комп'ютера з 4 ядрами CPU, 8 ГБ оперативної пам'яті, SSD від 50 ГБ, та стабільним інтернет-з'єднанням не нижче 100 Мбіт/с. Такий рівень ресурсів дозволяє запускати сервер, базу даних та клієнт одночасно, виконувати основні функції, проводити тестування та налагодження.

Для розгортання у продуктивному середовищі, де передбачається робота з великою кількістю даних і одночасних користувачів, рекомендовано використовувати сервер із 8–16 vCPU, 32 ГБ оперативної пам'яті та NVMe SSD на 500 ГБ або більше. Така конфігурація забезпечує високу швидкість обробки запитів, стабільну роботу бази даних та експорту документів. Сервер бажано запускати на Linux-дистрибутиві Ubuntu 22.04 LTS або Windows Server 2022, що забезпечують стабільність та оновлення системної безпеки.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

Користувачі взаємодіють із системою через браузер, тому клієнтські вимоги не є високими. Достатньо звичайного ноутбука або ПК із 4-ядерним процесором (Intel Core i5 або AMD Ryzen 5), 8 ГБ ОЗП та підтримкою сучасних браузерів (Google Chrome, Mozilla Firefox, Microsoft Edge). Рекомендована швидкість з'єднання — від 10 Мбіт/с, що забезпечує комфортну роботу з вебінтерфейсом та завантаженням PDF/Excel-документів.

Висновки до першого розділу

У першому розділі дипломної роботи було проведено детальний аналіз предметної області, що дало змогу визначити цільову аудиторію, ключові функціональні та нефункціональні вимоги до вебдодатку для складання кошторисів. Основне призначення системи полягає у створенні зручного, гнучкого та доступного інструменту для роботи з кошторисами в онлайн-середовищі.

Проаналізовано два актуальні аналоги: онлайн-сервіс «Скласти кошторис» та платформа OneB Invoice. Вони надають базову функціональність створення інвойсів та кошторисів, але мають обмеження у гнучкості, обробці даних, аналітиці та автоматизації. Це дозволило сформулювати бачення покращеного рішення, яке охоплює роботу з каталогами, управління користувачами, збереження історії, можливість експорту, а також впровадження спеціалізованих інструментів, таких як калькулятор фарбування труб та створення користувацьких формул.

На основі аналізу визначено доцільний стек технологій: Node.js з Express.js і PostgreSQL для серверної частини, React для фронтенду, а також набір додаткових бібліотек і засобів для обчислень, тестування та експорту. Сформульовано вимоги до апаратного забезпечення, що забезпечать стабільну та масштабовану роботу системи як у тестовому, так і в продуктивному середовищі.

Таким чином, розділ заклав ґрунтовну основу для реалізації вебдодатку, орієнтованого на ефективну, зручну та розширювану роботу з кошторисною документацією.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБДОДАТКУ ДЛЯ СКЛАДАННЯ КОШТОРИСІВ

2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи

Розроблений вебдодаток призначений для автоматизації процесу створення, редагування та управління кошторисами, що особливо актуально для малого бізнесу, приватних виконавців та компаній, які регулярно готують кошторисну документацію. Система реалізована як клієнт-серверний додаток, що надає зручний вебінтерфейс для взаємодії з користувачем через браузер. Для наглядної взаємодії між елементами вебдодатку було створено Діаграму варіантів використання (Рис. 2.1).

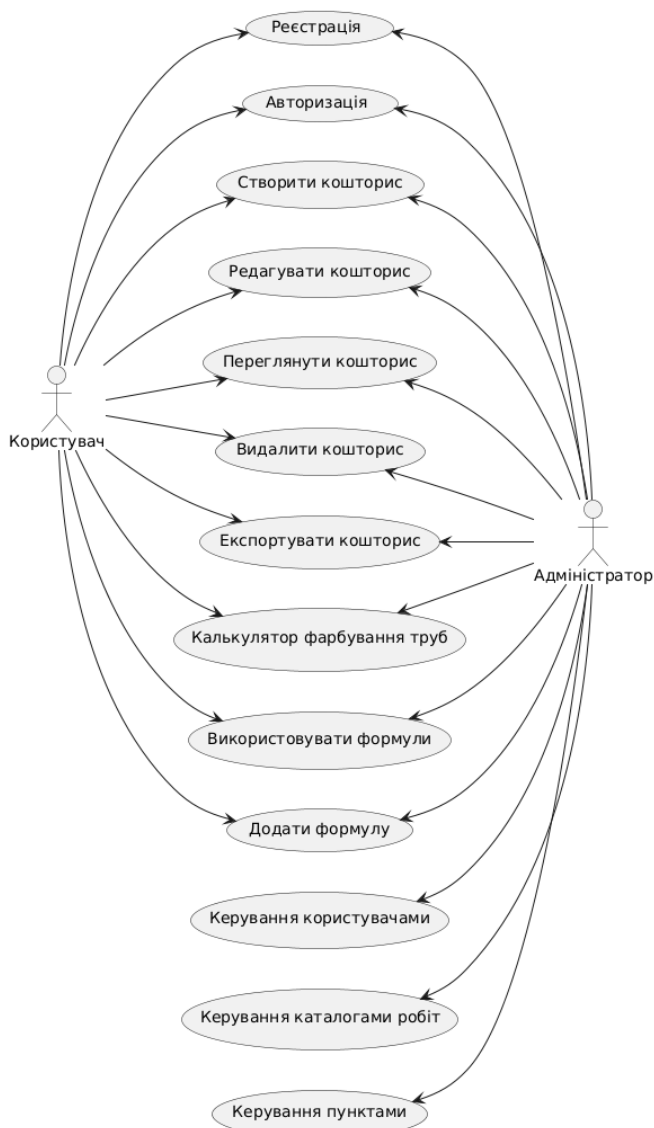


Рис. 2.1 Діаграма варіантів використання

Користувачі системи поділяються на дві основні ролі: звичайний користувач та адміністратор. Звичайні користувачі мають доступ до створення кошторисів, додавання позицій з каталогу, збереження історії, а також до спеціалізованих інструментів, таких як калькулятор фарбування труб чи редактор власних математичних формул. Адміністратор, у свою чергу, має можливість керувати користувачами, переглядати створені кошториси та здійснювати контроль за даними у системі.

Система включає широкий набір функціональних можливостей, що забезпечують ефективну та зручну роботу з кошторисною документацією, оптимізуючи процеси складання, редагування і контролю фінансових даних. Зокрема, серед основних функцій вебдодатку варто виділити кілька ключових напрямків, які сприяють підвищенню продуктивності та точності роботи користувачів.

Перш за все, система дозволяє створювати, редагувати та зберігати кошториси у зручному інтерфейсі. Користувачі мають змогу формувати нові документи, вносити зміни до існуючих кошторисів, додавати або видаляти позиції з урахуванням актуальних потреб проекту. Всі дані надійно зберігаються у базі даних, що дає можливість у будь-який момент повернутися до раніше створених кошторисів, продовжити їх редагування або експортувати для зовнішнього використання.

Особливу увагу приділено автоматичному обрахунку вартості. Система сама підсумовує загальну суму кошторису, враховуючи кількість і ціну кожної позиції, що значно знижує ризик помилок при ручних розрахунках. Такий підхід гарантує високу точність фінансових підсумків, що є важливим аспектом для прийняття управлінських рішень і подальшого бюджетування.

Важливою складовою є інтеграція з каталогом матеріалів і робіт. Користувачі мають доступ до упорядкованого переліку типових ресурсів — будівельних матеріалів, оздоблювальних робіт, інструментів та інших елементів, які можна швидко додати до кошторису. Це не лише спрощує стандартизацію документів, але й пришвидшує процес їх створення, особливо при роботі з великими проектами.

Серед ключових можливостей також реалізовано експорт кошторисів у популярні формати PDF та Excel. Такий функціонал дозволяє генерувати документи,

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

готові до друку, презентації або подальшої обробки в інших програмах. Експортовані файли зберігають всі розрахунки, табличні дані та службову інформацію у професійному вигляді, що значно полегшує передачу кошторисів між відділами або зовнішніми контрагентами.

Не менш важливою є адміністративна складова системи. Через спеціальну адмін-панель адміністратори можуть створювати і редагувати облікові записи користувачів, надавати або обмежувати права доступу, блокувати підозрілі акаунти та контролювати активність у системі. Це забезпечує високий рівень безпеки, захист даних та ефективне управління користувацькою базою, що особливо актуально для корпоративних клієнтів.

Підтримка зовнішніх математичних формул — користувачі можуть створювати власні формули для обчислень, які зберігаються в системі та можуть використовуватись у кошторисах, дозволяючи гнучко адаптувати систему до специфічних потреб. Використання інструментів спеціалізованих розрахунків — наприклад, реалізовано калькулятор фарбування труб, що дозволяє точно визначати обсяг і вартість робіт, базуючись на технічних параметрах об'єкта. Ці функції разом формують цілісну, зручну та масштабовану систему для створення, управління та аналізу кошторисної документації в будівельних або ремонтних проєктах.

Основна мета вебдодатку — надати універсальний, гнучкий та розширюваний інструмент для створення і супроводу кошторисної документації в онлайн-середовищі. Система враховує потреби як окремих виконавців, так і малих компаній, забезпечуючи простоту користування, адаптивність до конкретних бізнес-процесів та можливість масштабування.

Інтерфейс вебдодатку орієнтований на інтуїтивно зрозумілу логіку дій, що дозволяє швидко освоїти функціонал навіть без спеціальної підготовки. Гнучка структура системи передбачає подальше розширення функцій залежно від потреб користувача або специфіки проєкту.

Для візуалізації взаємодії користувачів із системою було побудовано діаграму варіантів використання, а також розроблено відповідні таблиці, які описують прецеденти.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				25
Змн.	Арк.	№ докум.	Підпис	Дата		

Опис діаграми варіантів використання наведено в таблицях 2.1 – 2.15.

Таблиця 2.1

Опис прецеденту «Авторизація / Реєстрація»

Назва варіанту використання	Авторизація / Реєстрація
Основні актори	Користувач, Адміністратор
Короткий опис	Користувач може створити обліковий запис або увійти в систему. Адміністратор також проходить авторизацію для доступу до панелі управління.

Таблиця 2.2

Опис прецеденту «Створення кошторису»

Назва варіанту використання	Створення кошторису
Основні актори	Користувач, Адміністратор
Короткий опис	Користувач або Адміністратор вводить інформацію щодо робіт, цін, кількості, формує кошторис із можливістю використання готового каталогу чи ручного вводу.

Таблиця 2.3

Опис прецеденту «Управління користувачами»

Назва варіанту використання	Управління користувачами
Основні актори	Адміністратор
Короткий опис	Адміністратор має змогу переглядати список користувачів, редагувати їхні ролі, блокувати чи активувувати облікові записи.

Таблиця 2.4

Опис прецеденту «Розрахунок за формулою»

Назва варіанту використання	Розрахунок за формулою
Основні актори	Користувач, Адміністратор
Короткий опис	Користувач та Адміністратор створює математичну формулу, вводить змінні та отримує результат. Формули зберігаються в обліковому записі для повторного використання.

Таблиця 2.5

Опис прецеденту «Експорт документів»

Назва варіанту використання	Експорт документів
Основні актори	Користувач, Адміністратор
Короткий опис	Користувач та Адміністратор має змогу експортувати кошторис у форматах PDF або Excel з відповідним оформленням (дата, логотип, підпис, таблиця).

Таблиця 2.6

Опис прецеденту «Створення каталогу»

Назва варіанту використання	Створення каталогу
Основні актори	Адміністратор
Короткий опис	Адміністратор може створювати нові каталоги, які містять тематично згруповані позиції (ресурси або роботи), що згодом використовуються при формуванні кошторисів. Каталоги доступні лише для перегляду та вибору користувачами

Таблиця 2.7

Опис прецеденту «Додавання пункту / позиції до каталогу»

Назва варіанту використання	Додавання пункту / позиції до каталогу
Основні актори	Адміністратор
Короткий опис	Адміністратор додає до наявного каталогу нові пункти з визначеними характеристиками: назва, одиниця виміру, вартість, опис. Ці пункти використовуються кошторисником при складанні кошторису.

Таблиця 2.8

Опис прецеденту «Редагування каталогу»

Назва варіанту використання	Редагування каталогу
Основні актори	Адміністратор
Короткий опис	Адміністратор може змінювати назву каталогу, його опис або інші параметри. Зміни автоматично застосовуються для користувачів, які використовують каталог у процесі створення кошторису.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

Таблиця 2.9

Опис прецеденту «Редагування пункту каталогу»

Назва варіанту використання	Редагування пункту каталогу
Основні актори	Адміністратор
Короткий опис	Адміністратор редагує дані пункту — змінює назву, одиницю виміру, ціну або опис. Змінені дані доступні користувачам під час створення нових кошторисів.

Таблиця 2.10

Опис прецеденту «Видалення каталогу»

Назва варіанту використання	Видалення каталогу
Основні актори	Адміністратор
Короткий опис	Адміністратор має можливість видалити каталог, якщо він більше не використовується. При цьому також видаляються всі пункти, що належать до цього каталогу. Якщо каталог уже застосований у створених кошторисах, його видалення може бути обмежене.

Таблиця 2.11

Опис прецеденту «Редагування пункту каталогу»

Назва варіанту використання	Редагування пункту каталогу
Основні актори	Адміністратор
Короткий опис	Адміністратор редагує дані пункту — змінює назву, одиницю виміру, ціну або опис. Змінені дані доступні користувачам під час створення нових кошторисів.

Таблиця 2.12

Опис прецеденту «Видалення пункту з каталогу»

Назва варіанту використання	Видалення пункту з каталогу
Основні актори	Адміністратор
Короткий опис	Адміністратор може видалити окремий пункт із каталогу. Якщо пункт вже використаний у кошторисах, його видалення може бути заборонено або обмежено відповідною перевіркою.

Таблиця 2.12

Опис прецеденту «Створення користувацької формули»

Назва варіанту використання	Створення користувацької формули
Основні актори	Користувач, Адміністратор
Короткий опис	Користувач та Адміністратор може створити власну математичну формулу, використовуючи змінні та оператори. Формула зберігається в особистому кабінеті та доступна лише відповідному користувачу.

Таблиця 2.13

Опис прецеденту «Редагування або видалення формули»

Назва варіанту використання	Редагування або видалення формули
Основні актори	Користувач, Адміністратор
Короткий опис	Користувач та Адміністратор може змінити або видалити раніше збережену формулу. Це дозволяє адаптувати розрахунки під нові умови або очистити список від непотрібних формул.

Таблиця 2.14

Опис прецеденту «Обчислення результату формули»

Назва варіанту використання	Обчислення результату формули
Основні актори	Користувач, Адміністратор
Короткий опис	Користувач та Адміністратор вводить значення змінних, формула автоматично обчислюється та повертає результат. Можливо використовувати цей механізм як частину створення кошторису або окремо.

Таблиця 2.15

Опис прецеденту «Розрахунок фарбування труб»

Назва варіанту використання	Розрахунок фарбування труб
Основні актори	Користувач, Адміністратор

Короткий опис	Користувач та Адміністратор може перейти до окремої сторінки калькулятора фарбування труб, ввести параметри (діаметр, довжину, кількість шарів тощо) та отримати результат розрахунку необхідної кількості фарби.
---------------	---

Адміністратор виконує ключові завдання з управління системою: має доступ до списку користувачів, списку каталогів та пунктів, керує правами доступу, створює та редагує каталоги видів робіт та їх пункти, а також контролює створені кошториси. Це забезпечує централізований контроль за функціональністю системи та захистом даних.

Користувач відповідає за створення кошторисів: додає позиції з доступного каталогу, виконує розрахунки, використовує власні або збережені формули, а також експортує кошториси у формати PDF чи Excel.

З точки зору програмної архітектури, система побудована за об'єктно-орієнтованим підходом, що дозволяє досягти високої гнучкості, масштабованості та спрощує подальший розвиток функціоналу. Основними об'єктами системи є: користувач, позиція кошторису, каталог, формула, калькулятор та експортний документ. Кожен із цих об'єктів реалізовано як окрему сутність із чітко визначеними атрибутами та методами, що описують їхню поведінку й взаємозв'язки.

У вебдодатку кожен користувач має доступ до власного облікового запису, де зберігаються його авторизаційні дані, роль (користувач або адміністратор), історія створених кошторисів та особисті формули для розрахунків. Під час створення кошторису користувач заповнює загальну інформацію про проєкт, додає необхідні позиції, вказує дату та отримує підсумкові обчислення. Кожна позиція — це окрема одиниця роботи або товару з вказаними параметрами, такими як назва, ціна, кількість і належність до певного каталогу. Каталоги, які структурують ці пункти за типами, доступні лише для редагування адміністратором.

Формули створюються користувачем і містять змінні, які можна використовувати у власних обчисленнях за допомогою калькулятора. У калькуляторі можна вводити значення змінних для обраної формули і миттєво отримувати результат.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

Після завершення роботи над кошторисом його можна експортувати у вигляді фінального документа — PDF або Excel-файлу.

Ця модель дозволяє ефективно розділяти обов'язки між компонентами системи, спрощує адаптацію до нових вимог і забезпечує зручність у тестуванні та підтримці коду.

Визначення вимог до системи є важливим етапом, під час якого сформульовано вимоги до функціональності, інтерфейсу, безпеки та продуктивності вебдодатку, що автоматизує процес створення кошторисів у будівельній сфері. Основна мета системи — надати зручний інструмент для швидкого формування кошторисів, розрахунків та експорту документації, з урахуванням актуальних цін та нормативів. Система повинна бути безпечною, доступною онлайн та придатною для використання на різних пристроях.

Бізнес-вимоги полягають у створенні вебзастосунку, що спрощує процес підготовки кошторисної документації. Інтерфейс повинен бути зрозумілим навіть для користувачів без глибоких технічних знань. Додаток має гарантувати надійний зберігання даних, централізоване керування адміністраторами та зручний доступ до інструментів для створення, редагування й експорту кошторисів.

Вимоги користувачів передбачають наявність у системі низки важливих можливостей для кошторисників, які значно полегшують процес створення та ведення кошторисної документації. Користувач може формувати кошториси, додаючи необхідні позиції з попередньо створеного каталогу послуг та матеріалів. Усі елементи документа доступні для редагування — зокрема, можна змінювати ціни, кількість одиниць або інші параметри, а обрахунок загальної вартості відбувається миттєво в режимі реального часу.

Також система дозволяє використовувати збережені користувачем формули або вбудовані інструменти для спеціалізованих розрахунків, наприклад, калькулятор фарбування труб. Готові кошториси можна не лише переглядати у вебінтерфейсі, а й зберігати або експортувати у зручних форматах — PDF або Excel — для

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

подальшої передачі, друку або архівування. Інтерфейс вебдодатку розроблений таким чином, щоб бути максимально зрозумілим і доступним, із підтримкою автоматичних розрахунків.

Вимоги адміністратора передбачають надання розширених прав доступу, що дозволяють повноцінно керувати користувачами та ключовими параметрами вебдодатку. Зокрема, він має можливість переглядати список зареєстрованих користувачів, змінювати їхні ролі та обмеження доступу відповідно до рівня повноважень. Крім того, адміністратор керує каталогами робіт і пов'язаними з ними пунктами — він може додавати нові елементи, редагувати або видаляти наявні, що забезпечує актуальність і повноту довідкової бази для формування кошторисів. Також доступна зміна прав доступу до різних частин системи, що дозволяє гнучко налаштовувати взаємодію між ролями користувачів. Окрему увагу адміністратор приділяє загальним параметрам системи, зокрема налаштуванню одиниць виміру, податкових ставок (ПДВ) та валюти розрахунків — ці значення мають значення для точного відображення фінансових даних у кошторисах. Окрім управлінських функцій, адміністратор має доступ до всіх кошторисів, створених користувачами, що дозволяє здійснювати контроль, аудит і аналіз даних у межах всієї системи..

Характеристика об'єкта комп'ютеризації полягає в тому, що розроблювана система є сучасним вебдодатком, який дозволяє створювати, редагувати, зберігати та експортувати кошториси. Усі дані зберігаються в базі даних, а доступ до функціоналу регулюється ролями користувачів. Застосунок взаємодіє з користувачами через браузер, не потребуючи встановлення локального ПЗ. Архітектура системи підтримує подальше розширення — наприклад, додавання нових формул чи шаблонів.

Функціональні вимоги передбачають реалізацію низки основних функцій, які забезпечують повноцінну роботу з кошторисною інформацією та її управління. Користувачі можуть проходити процедуру реєстрації та авторизації, при цьому система підтримує розмежування ролей — наприклад, звичайний користувач і адміністратор мають різні рівні доступу до функцій додатку. Після входу в систему користувач має змогу створювати кошториси, додаючи до них позиції з каталогу робіт

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				32
Змн.	Арк.	№ докум.	Підпис	Дата		

та матеріалів. Автоматичний розрахунок вартості виконується на основі введених даних, що дозволяє миттєво отримувати підсумкову суму. За потреби можна використовувати математичні формули — як загальні, доступні всім, так і індивідуальні, створені самим користувачем. Готові кошториси можна експортувати у форматі PDF або Excel для подальшого використання, зберігання чи друку. Також реалізовано можливість пошуку та сортування кошторисів за різними параметрами, що дозволяє швидко знаходити потрібні документи навіть у великій базі даних. Для спеціалізованих розрахунків передбачено інтегрований калькулятор, наприклад, для обчислення об'єму фарбування труб. Кожен користувач має доступ до власного кабінету, де може переглядати створені документи та персональні налаштування. Адміністратори, у свою чергу, мають додаткові функції — зокрема, управління каталогами, позиціями, користувачами та іншими довідниками, що забезпечує стабільне функціонування системи та її постійне оновлення..

Нефункціональні вимоги передбачають відповідність системи ключовим критеріям, що забезпечують її стабільну, безпечну та ефективну роботу в реальних умовах експлуатації. Зокрема, важливим критерієм є безпека — усі дані мають передаватися у зашифрованому вигляді, автентифікація користувачів здійснюється за допомогою токенів JWT, а система передбачає чітке розмежування прав доступу відповідно до ролі кожного користувача. З точки зору продуктивності, додаток повинен забезпечувати обробку більшості запитів у межах 1–2 секунд, що дозволяє підтримувати комфортну швидкість взаємодії навіть при значному обсязі даних. Масштабованість системи дозволяє працювати одночасно з десятками активних користувачів, не втрачаючи у продуктивності, що важливо для розширення або впровадження у невеликі організації чи команди. Також система повинна демонструвати кросплатформеність, тобто стабільну роботу у найпоширеніших сучасних браузерах, таких як Google Chrome, Mozilla Firefox та Microsoft Edge. Надійність — ще одна обов'язкова вимога: система має бути доступною цілодобово, без частих збоїв чи втрат доступу до функціоналу. Окрему увагу приділено юзабіліті — інтерфейс має бути інтуїтивно зрозумілим, простим у використанні навіть для новачків, із ло-

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				33
Змн.	Арк.	№ докум.	Підпис	Дата		

гічною навігацією та швидким доступом до основних функцій. Незважаючи на обмеження щодо повної адаптивності, базова підтримка різних розмірів екранів також врахована при розробці.

Аналіз функціональних вимог дозволив визначити ключові об'єкти системи: користувач, кошторис, позиція кошторису, формула, каталог, експортний документ та калькулятор. Їхня взаємодія та структура подані на діаграмі класів (Рис. 2.2), яка ілюструє реалізацію архітектури системи на основі об'єктно-орієнтованого підходу.

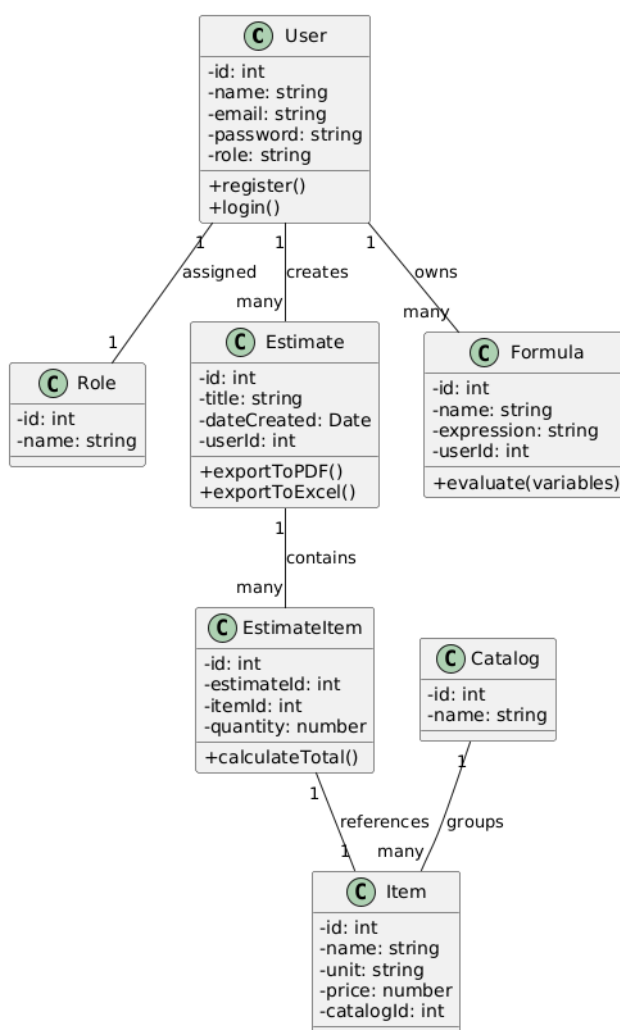


Рис. 2.2 Діаграма класів

Опис діаграми класів полягає в тому, що вебдодаток для створення та управління кошторисами реалізовано за об'єктно-орієнтованою моделлю. Основні класи системи описують ключові сутності: користувачів, їхні ролі, кошториси, пункти

кошторису, каталоги, позиції, формули та функціональність експорту. Така структура дозволяє чітко організувати логіку взаємодії та підтримувати масштабованість системи.

Клас «Користувач» (User) відповідає за повну ідентифікацію та управління доступом до системи. Основними обов'язками цього класу є автентифікація (перевірка особи користувача під час входу) та авторизація (надання доступу до функцій відповідно до ролі). До атрибутів належать: `id` — унікальний ідентифікатор користувача в системі; `name` — ім'я користувача; `email` — електронна адреса, що використовується для входу; `password` — хешований пароль; `role` — роль користувача (звичайний користувач або адміністратор). Користувачі можуть реєструватися та входити в систему через методи `register()` та `login()`. Після входу користувач має можливість створювати кошториси, зберігати власні формули для обчислень і працювати з інтерфейсом, адаптованим до його ролі в системі (наприклад, адміністратор має розширені можливості керування контентом і користувачами).

Клас «Роль» (Role) визначає рівень доступу користувача до функціоналу вебдодатку. Основними ролями є звичайний користувач та адміністратор. Атрибути включають `id` — ідентифікатор ролі та `name` — назва ролі. Клас застосовується для контролю прав доступу, наприклад, лише адміністратор може редагувати або створювати нові каталоги та мати повний доступ до адміністративної панелі. Система перевіряє роль кожного користувача під час виконання дій, які потребують особливих прав.

Клас «Кошторис» (Estimate) зберігає ключову інформацію про конкретний кошторис, який створює користувач. Основні атрибути — `id` (ідентифікатор), `title` (назва кошторису), `dateCreated` (дата створення), `userId` (ідентифікатор користувача, що його створив). Кошторис об'єднує кілька пов'язаних об'єктів `EstimateItem`, які описують окремі пункти робіт або товарів. Крім того, цей клас має методи `exportToPDF()` та `exportToExcel()`, які дозволяють вивантажити дані кошторису у відповідних форматах для подальшого використання, друку або збереження. Завдяки цьому користувач може формувати повноцінні документи безпосередньо з інтерфейсу програми.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				35
Змн.	Арк.	№ докум.	Підпис	Дата		

Клас «Пункт кошторису» (EstimateItem) виступає проміжною ланкою між кошторисом і конкретною позицією (Item). Його завдання — зберігати інформацію про те, скільки одиниць певного товару чи послуги включено до кошторису. Атрибути: id — ідентифікатор пункту, estimateId — зв'язок із кошторисом, itemId — зв'язок із позицією, quantity — кількість. Метод calculateTotal() дозволяє обчислити повну вартість позиції на основі ціни за одиницю та вказаної кількості. Цей клас формує основу підрахунків у кошторисі та забезпечує точність фінансових результатів.

Клас «Каталог» (Catalog) дозволяє структурувати всі доступні позиції в системі за категоріями. Прикладами можуть бути «Оздоблювальні роботи», «Будівельні матеріали» тощо. Атрибути включають id — ідентифікатор каталогу та name — його назву. Кожен каталог включає в себе багато Item, тобто позицій. Управління каталогами здійснюється лише адміністратором, що гарантує централізований контроль над структурою даних та запобігає дублюванню або хаотичному додаванню пунктів.

Клас «Пункт / Позиція» (Item) описує конкретну одиницю товару чи послуги, яку можна додати до кошторису. Основні атрибути: id — ідентифікатор позиції, name — назва, unit — одиниця виміру (наприклад, м², м.п., шт.), price — ціна за одиницю, catalogId — зв'язок із певним каталогом. Item може бути використано в одному або кількох кошторисах завдяки зв'язку з EstimateItem. Цей клас дає змогу легко оперувати базою типових елементів для кошторисів.

Клас «Формула» (Formula) надає можливість користувачу зберігати власні математичні вирази для подальших обчислень. Формула може містити змінні, які задаються під час використання. Атрибути: id — унікальний ідентифікатор, name — назва формули, expression — сам вираз (наприклад, $2*a + b^2$), userId — зв'язок з користувачем. Метод evaluate(variables) дозволяє обчислювати результат, підставляючи значення змінних. Формули активно застосовуються в калькуляторі системи або можуть бути частиною складніших розрахунків у кошторисах. Вони розширюють функціональність додатку та забезпечують персоналізовані сценарії розрахунків.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

У системі реалізовані ключові зв'язки між основними сутностями, які відображають логіку взаємодії користувача з кошторисами, формулами та довідковими даними. Кожен користувач (User) може створювати кілька кошторисів (Estimate), тобто між цими класами встановлено зв'язок "один до багатьох". У свою чергу, кожен кошторис (Estimate) містить перелік позицій (EstimateItem), що також утворює зв'язок "один до багатьох". Кожна позиція в кошторисі (EstimateItem) прив'язана до конкретного елемента з довідника (Item), що дозволяє зберігати актуальні дані про ціну, одиниці виміру та опис послуг або матеріалів. Самі ж елементи (Item) згруповані у каталоги (Catalog), де один каталог може включати багато таких елементів. Користувачі також мають можливість створювати власні формули для обчислень (Formula), що реалізується через зв'язок "один до багатьох" між класами User і Formula. Крім того, кожен користувач має певну роль (Role), яка визначає рівень доступу до функціоналу системи — цей зв'язок є "один до одного".

Переваги такої структури — чіткий розподіл відповідальностей, легкість масштабування (наприклад, додавання нових типів ресурсів або ролей), а також можливість розширення функціоналу без значних змін у логіці взаємодії об'єктів.

2.2 Розробка бази даних системи

У вебдодатку для складання кошторисів використовується система управління базами даних PostgreSQL, що базується на реляційній моделі та забезпечує високу продуктивність, надійність і гнучкість роботи з даними. Усі основні сутності вебдодатку зберігаються у відповідних таблицях, що дозволяє ефективно організувати взаємозв'язки між користувачами, кошторисами, позиціями, каталогами тощо. База даних системи побудована на основі шести основних таблиць, які забезпечують збереження, структурування та взаємозв'язок усіх ключових даних вебдодатку.

Таблиця users призначена для зберігання інформації про зареєстрованих користувачів. У ній містяться такі дані, як ім'я користувача, електронна пошта, хешований пароль, роль (наприклад, звичайний користувач або адміністратор), а також службова інформація — дата реєстрації, статус активності тощо.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		37

Таблиця *estimates* містить відомості про всі створені кошториси. Кожен запис прив'язаний до конкретного користувача, що дозволяє фіксувати, хто саме створив документ. Тут зберігаються назва кошторису, дата створення, загальна сума, примітки або опис.

Таблиця *estimate_items* реалізує зв'язок між кошторисами та окремими пунктами, які до них додані. Кожен запис у цій таблиці відповідає одній позиції у конкретному кошторисі. Зберігаються дані про кількість, ціну, а також посилання на елемент з таблиці *items*, що дозволяє динамічно формувати зміст кожного кошторису.

Таблиця *items* містить перелік пунктів, які можуть бути використані під час складання кошторису. До кожного елемента додаються назва, одиниця виміру, базова ціна та опис. Ці пункти є типовими елементами будівельних або ремонтних робіт і можуть бути багаторазово використані в різних кошторисах.

Таблиця *catalogs* структурує пункти з *items* за категоріями. Це дозволяє логічно організовувати інформацію, спрощуючи навігацію користувача під час створення кошторису. Наприклад, можна розділити елементи за напрямками робіт: електромонтаж, оздоблення, сантехніка тощо.

Таблиця *user_formulas* відповідає за зберігання індивідуальних формул, створених користувачами для виконання спеціалізованих обчислень. Тут зберігаються назви формул, математичні вирази у вигляді рядків, змінні, а також посилання на користувача, якому належить ця формула.

Ця структура дозволяє ефективно зберігати дані, підтримує логіку взаємодії між об'єктами системи та забезпечує масштабованість для подальшого розширення функціоналу. Вона також сприяє підвищенню надійності та спрощує процес супроводу програмного забезпечення.

Опис кожної з таблиць бази даних наведений у таблицях 2.16 – 2.21.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.16

Опис таблиці Users

Users		
Поле	Тип	Призначення
id	int	Унікальний ідентифікатор користувача
username	varchar	Ім'я користувача
email	varchar	Електронна пошта
password	varchar	Хешований пароль
role	varchar	Роль користувача (admin, user)

Таблиця 2.17

Опис таблиці Estimates

Estimates		
Поле	Тип	Призначення
id	int	Унікальний ідентифікатор кошторису
user_id	int	Зовнішній ключ на автора кошторису
title	varchar	Назва кошторису
description	varchar	Короткий опис або пояснення до кошторису
total	decimal	Загальна сума кошторису з урахуванням усіх позицій і накладних витрат
created_at	timestamp	Дата створення кошторису
overhead_percentage	decimal	Відсоток накладних витрат (як частина загальної суми)
overhead_amount	decimal	Обчислена сума накладних витрат на основі відсотка

Таблиця 2.18

Опис таблиці Estimate_items

Estimate_items		
Поле	Тип	Призначення
id	int	Унікальний ідентифікатор запису позиції у кошторисі

Estimate_items		
Поле	Тип	Призначення
estimateId	int	Зовнішній ключ на таблицю estimates, що вказує, до якого кошторису належить позиція
itemId	int	Зовнішній ключ на таблицю items, що вказує, який саме пункт використовується
quantity	decimal	Кількість одиниць для даного пункту
total	decimal	Загальна сума за цю позицію (кількість × ціна або custom_price)
custom_price	decimal	Індивідуальна ціна за одиницю, яку можна задати вручну

Таблиця 2.19

Опис таблиці Items

Items		
Поле	Тип	Призначення
id	int	Ідентифікатор пункту кошторису
name	varchar	Назва пункту
unit	varchar	Одиниця виміру
price	decimal	Ціна за одиницю
catalogId	int	Зовнішній ключ на каталог

Таблиця 2.20

Опис таблиці Catalogs

Catalogs		
Поле	Тип	Призначення
id	int	Ідентифікатор каталогу
name	varchar	Назва категорії робіт

Опис таблиці User_formulas

User_formulas		
Поле	Тип	Призначення
id	int	Ідентифікатор формули
name	varchar	Назва формули
expression	text	Вираз з математичними операціями
userId	int	Зовнішній ключ на власника формули

У контексті реалізації даного вебдодатку, всі дані зберігаються у централізованій реляційній базі даних PostgreSQL. Це дозволяє ефективно зберігати, обробляти та оновлювати інформацію, пов'язану з користувачами, кошторисами, позиціями, каталогами та збереженими формулами.

Кожна таблиця в базі даних має чітко визначену структуру, з ключовими полями та зв'язками між сутностями, що забезпечує цілісність і логічну узгодженість даних. Наприклад, таблиця estimate_items використовує зовнішні ключі до таблиць estimates та items, що дозволяє точно ідентифікувати, які пункти включено до конкретного кошторису.

Система підтримує механізми створення, оновлення та видалення даних відповідно до ролі користувача. Звичайні користувачі можуть створювати та редагувати кошториси, використовувати калькулятор і формули, тоді як адміністратор має розширені повноваження: керує списком користувачів, редагує каталоги та контролює повноту і коректність внесених даних.

Реляційна модель дозволяє реалізувати гнучкі запити до даних, забезпечити надійне збереження інформації та масштабованість додатку для подальшого розвитку. Завдяки використанню SQL та ORM-бібліотеки Sequelize, взаємодія між додатком і базою даних є безпечною, структурованою та зручною для підтримки.

2.3 Проектування та реалізація алгоритмів роботи системи

Проектування алгоритмів функціонування вебдодатку для складання кошторисів базується на аналізі типових сценаріїв використання. З метою формалізації

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		41

поведінки користувачів і взаємодії з функціональними модулями системи, було розроблено низку діаграм активностей. Вони дозволяють структуровано відобразити послідовність дій, логіку переходів між станами, а також визначити відповідальність за ті чи інші дії (користувач або система). Це є важливою складовою інженерного підходу до розробки, оскільки забезпечує прозору логіку функціонування й оптимізує подальшу реалізацію програмного коду.

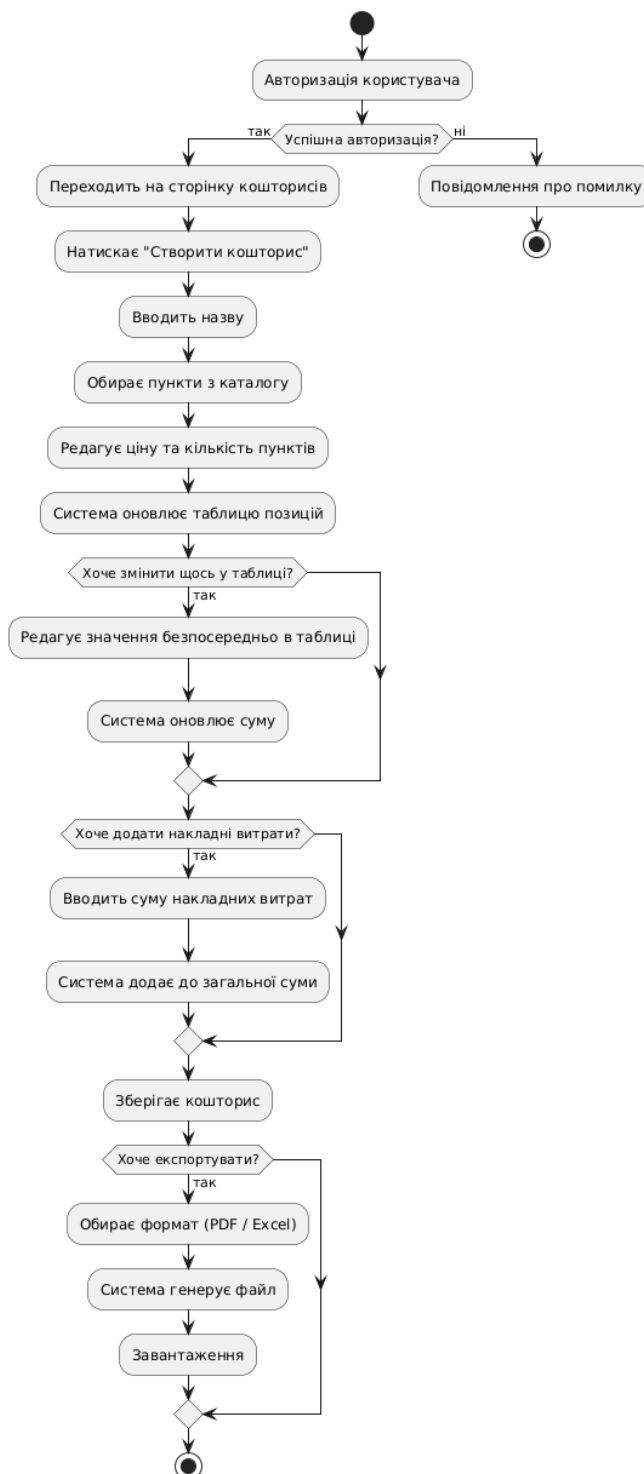


Рис. 2.3 Діаграма активності «Створення нового кошторису»

На діаграмі активностей (Рис. 2.3), представлено алгоритм створення кошторису звичайним користувачем. Процес розпочинається з успішної авторизації — тільки після неї користувач отримує доступ до функціоналу системи.

Далі користувач переходить до розділу створення кошторису, де заповнює основну інформацію: назву, опис тощо. Наступним кроком є вибір позицій із каталогу: система надає доступ до переліку доступних елементів, згрупованих за категоріями. Користувач має змогу вказати кількість одиниць кожної позиції та, за потреби, задати власну вартість, яка може відрізнятись від стандартної.

Система на основі введених даних виконує автоматичний розрахунок підсумкової вартості з урахуванням накладних витрат. Після перевірки введених даних користувач натискає кнопку підтвердження, і кошторис зберігається в базі даних, стаючи доступним для подальшої роботи.

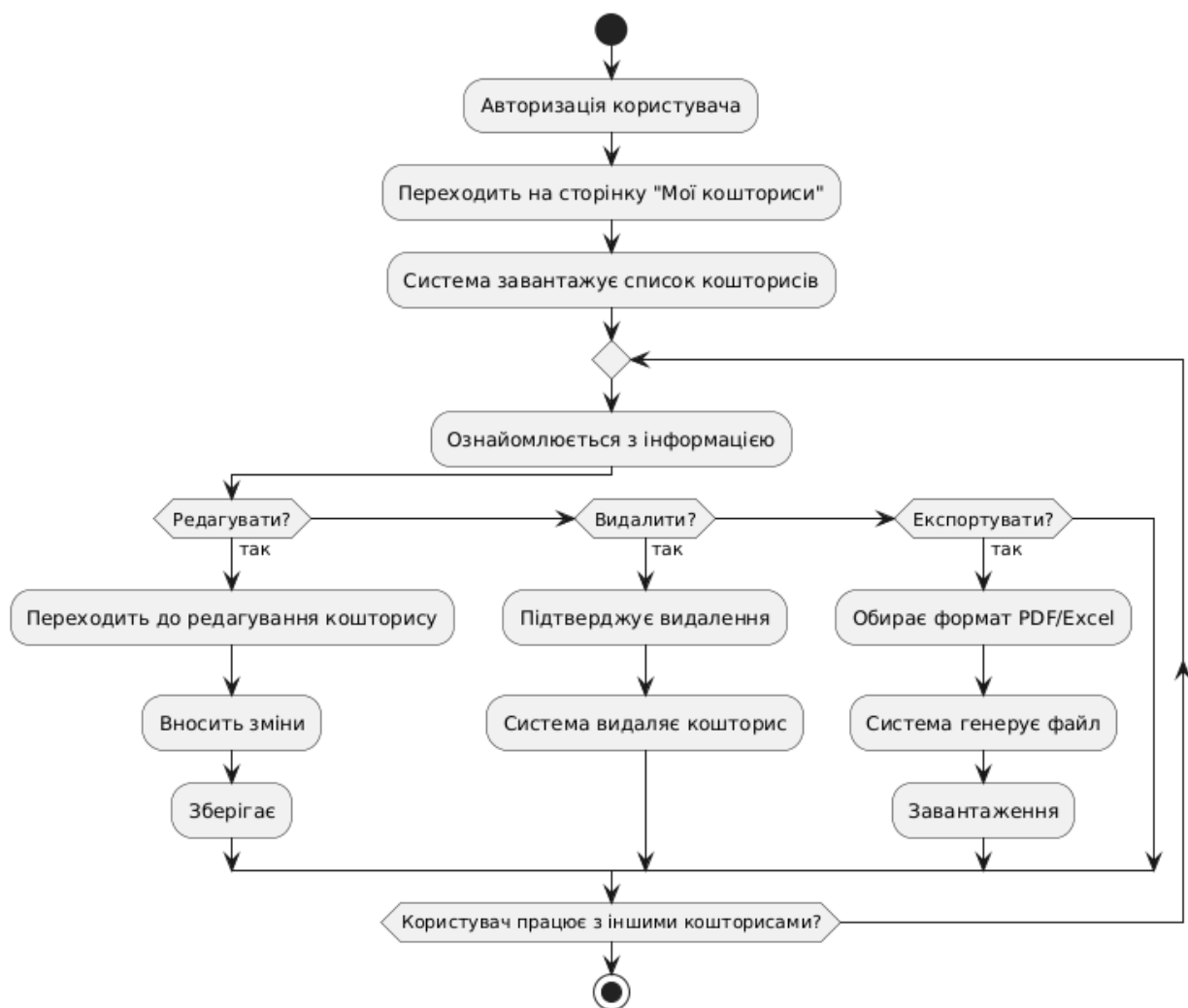


Рис. 2.4 Діаграма активності «Керування раніше створеними кошторисами»

Діаграма (Рис. 2.4) відображає повну послідовність взаємодії користувача з функціональністю сторінки «Мої кошториси», яка слугує особистим робочим простором для перегляду та керування всіма створеними кошторисами. Після того як користувач авторизується у системі та переходить на цю сторінку, система автоматично надсилає запит до бази даних, щоб отримати всю релевантну інформацію. У відповідь користувачеві відображається повний перелік кошторисів, які були створені саме ним. Кожен запис включає основні дані — назву, дату створення, загальну суму тощо.

У межах інтерфейсу користувач має змогу працювати з кожним кошторисом детальніше. При натисканні на конкретний запис відкривається сторінка, де відображається структура вибраного кошторису: перераховані усі додані позиції, їх кількість, вартість, а також підсумкова вартість із урахуванням накладних витрат. За необхідності користувач може внести зміни до вже наявного кошторису — наприклад, оновити кількість одиниць певної позиції, замінити її на іншу або скоригувати індивідуальну ціну. Усі зміни автоматично враховуються системою, яка перераховує загальну вартість кошторису в режимі реального часу.

Крім редагування, передбачена можливість повного видалення кошторису. Якщо певний документ більше не актуальний, користувач може ініціювати його вилучення, після чого система остаточно видаляє відповідний запис із бази даних. Окремо реалізовано функцію експорту: користувач може обрати формат, у якому він бажає зберегти кошторис — це може бути як PDF-документ, зручний для друку або надсилання, так і таблиця Excel для подальших розрахунків чи аналізу. Система автоматично формує файл за обраним шаблоном і пропонує його до завантаження.

Діаграма (Рис. 2.5) демонструє процес створення користувачем власної математичної формули, яка в подальшому може використовуватись для автоматизованих обчислень у межах кошторисів. Ця функціональність реалізована як окремий модуль у вигляді калькулятора формул, доступ до якого користувач отримує після авторизації.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		44

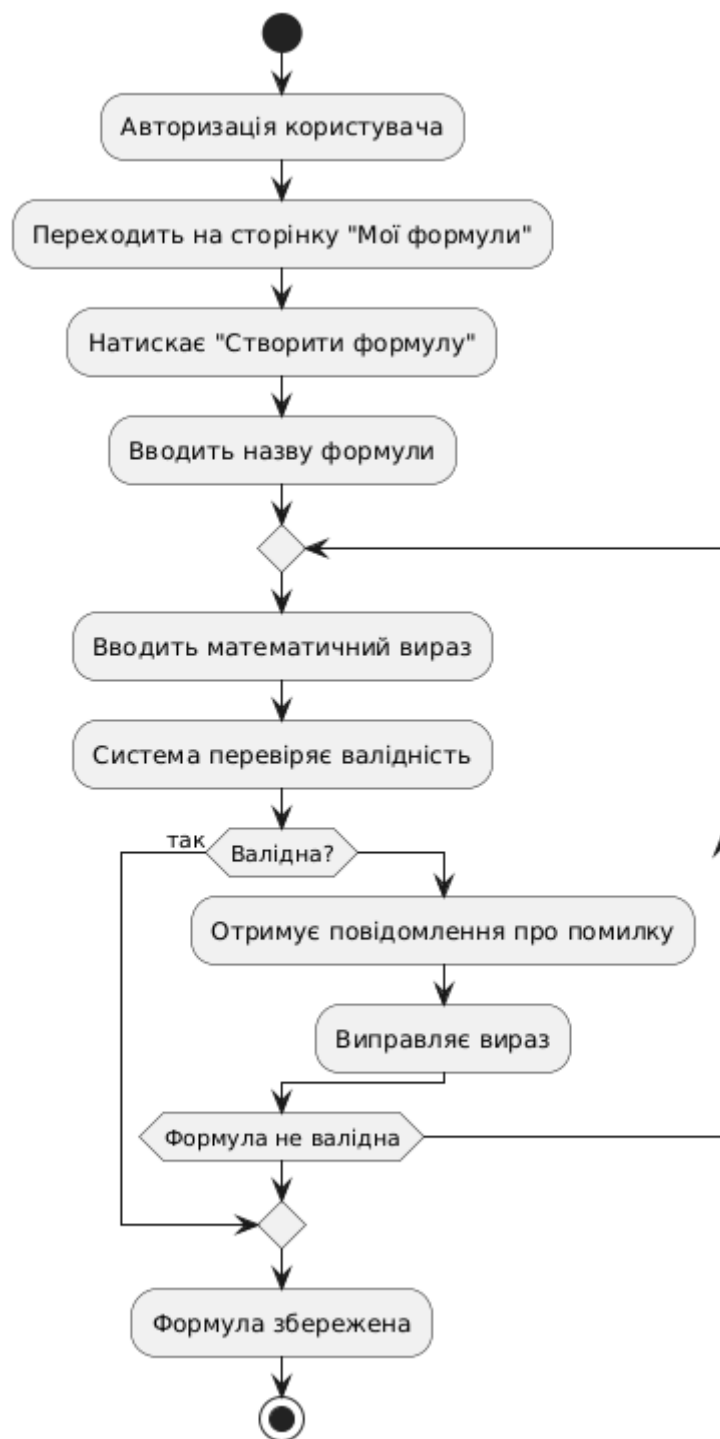


Рис. 2.5 Діаграма активності «Створення користувацької формули»

Опинившись у розділі «Мої формули», користувач має змогу розпочати створення нового математичного виразу. Інтерфейс пропонує йому ввести назву формули, що слугуватиме її ідентифікатором у списку, а також записати сам вираз у математичному форматі із застосуванням змінних. Наприклад, користувач може ввести щось на кшталт $L * D * 3.14$, де L і D — це умовні змінні, значення яких

буде підставлено під час обчислення. Після введення даних система виконує процес валідації, що включає перевірку правильності синтаксису, допустимості використаних математичних операторів і коректності структури виразу загалом. Якщо перевірка проходить успішно, введена формула зберігається у базі даних і автоматично додається до списку доступних обчислень у калькуляторі.

У подальшому користувач може обирати створену формулу, підставляти значення змінних вручну через зручну форму введення і миттєво отримувати результат розрахунку. Такий підхід значно спрощує роботу з типовими чи повторюваними математичними операціями, які часто використовуються під час складання кошторисів. Таким чином, система дає змогу не лише зберігати важливі обчислювальні шаблони, але й забезпечує їх повторне застосування без необхідності ручного підрахунку щоразу.

На (Рис.2.6) представлено алгоритм взаємодії адміністратора з панеллю керування системою, яка виконує ключову роль у підтримці структури та актуальності інформації, що використовується всіма користувачами вебдодатку. Доступ до цього інтерфейсу відкривається лише після проходження авторизації користувачем, який має відповідну роль адміністратора. Після успішного входу адміністратор потрапляє до централізованого інтерфейсу адміністрування, що реалізований у вигляді інтерактивної панелі з кількома функціональними розділами.

Перший розділ присвячений роботі з користувачами. Тут адміністратор бачить список усіх зареєстрованих у системі облікових записів із можливістю перегляду детальної інформації щодо кожного з них. У межах цього розділу реалізована можливість змінювати ролі користувачів — наприклад, надати адміністративні права або навпаки, обмежити доступ до певного функціоналу. За потреби адміністратор також може деактивувати або повністю видалити обліковий запис, якщо користувач порушив правила використання системи або більше не є активним.

Другий розділ інтерфейсу зосереджений на керуванні каталогами. Каталоги є своєрідними категоріями робіт, до яких належать різні позиції кошторису. У цьому блоці адміністратор має змогу створювати нові каталоги, які можуть відо-

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

бражати додаткові напрямки робіт чи послуг, перейменовувати вже наявні для кращої структуризації або видаляти ті, що більше не використовуються. Це дозволяє підтримувати зручну й логічну організацію елементів кошторису відповідно до потреб проєкту.

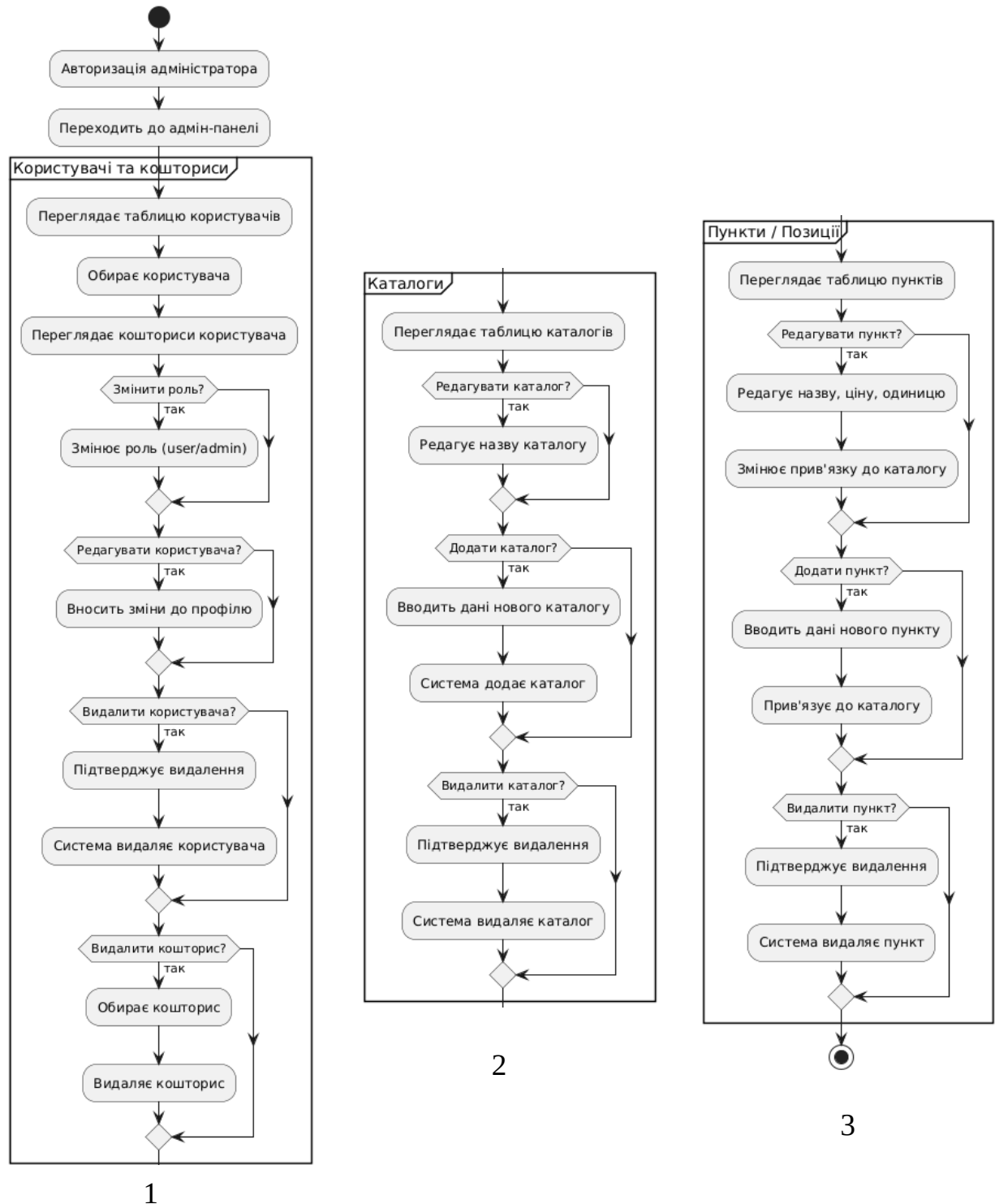


Рис. 2.6 Діаграма активності «Алгоритм взаємодії з адміністративною панеллю»

Третій ключовий розділ — керування позиціями, тобто конкретними елементами, які входять до складу кошторисів. Адміністратор може переглядати список усіх доступних позицій, вносити зміни до назв, оновлювати одиниці виміру або коригувати вартість відповідно до поточної економічної ситуації. Така можливість є надзвичайно важливою для підтримання системи в актуальному стані та забезпечення достовірності фінансових розрахунків.

Усі дії, які виконує адміністратор у межах панелі керування, супроводжуються надсиленням відповідних запитів до серверної частини системи. Після виконання запиту зміни автоматично відображаються в інтерфейсі без потреби в ручному оновленні сторінки, що забезпечується за рахунок використання сучасної технології React. Завдяки цьому адміністратор може оперативно керувати великим обсягом інформації та забезпечувати безперервну роботу системи без затримок і додаткового навантаження на користувачів.

Усі діаграми активностей, наведені в цьому підрозділі, виконують важливу роль у візуалізації внутрішньої логіки роботи вебдодатку. Вони наочно демонструють, як саме відбувається взаємодія між користувачем і системою, які кроки проходить той чи інший процес — від початкового запиту до збереження або обробки даних. Завдяки такій формалізації легко простежити послідовність виконання дій, зрозуміти, яка сторона відповідає за конкретні операції — клієнтська частина (користувач) чи серверна (система), — а також визначити, в яких точках можливі помилки або конфлікти.

Таке унаочнення дозволяє не лише краще спланувати архітектуру майбутнього коду, а й значно полегшує процес розробки, оскільки кожен логічний крок заздалегідь передбачений і описаний. Діаграми активностей слугують своєрідним містком між функціональними вимогами, сформульованими в технічному завданні, і їхньою програмною реалізацією. Вони допомагають розробникам уникати непотрібних припущень і неузгодженостей, що, у свою чергу, знижує ризик виникнення помилок на етапі впровадження та тестування. Окрім цього, наявність чітко описаних сценаріїв взаємодії дозволяє легко вносити зміни або масштабувати функціональність у майбутньому без порушення загальної логіки роботи системи.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		48

2.4 Реалізація функціоналу вебдодатку для складання кошторису

Вебдодаток реалізовано з використанням сучасного технологічного стеку, який включає React на фронтенді, Node.js з фреймворком Express.js на бекенді та PostgreSQL як реляційну базу даних. Основним функціоналом додатку є створення, редагування та експорт кошторисів, а також використання користувацьких формул і калькулятора для розрахунків.

Реєстрація та авторизація є початковим етапом взаємодії користувача із системою, який реалізується через функціонал створення облікового запису або входу до нього. Це забезпечує безпечний доступ до персоналізованих функцій вебдодатку. Для цього використано механізм токенів авторизації, зокрема JSON Web Token (JWT), який дозволяє зберігати стан сесії без використання серверної пам'яті. Після реєстрації дані нового користувача зберігаються у таблиці users бази даних, де фіксуються ім'я, електронна пошта, роль (user/admin) та захешований пароль. Авторизація здійснюється через перевірку валідності облікових даних і генерацію токена з відповідним часом дії. Нижче наведено фрагмент коду (Рис. 2.7), що відповідає за створення нового користувача (реєстрацію):

```
const hash = await bcrypt.hash(password, 10);
const result = await pool.query(
  'INSERT INTO users (username, email, password) VALUES ($1, $2, $3) RETURNING id, username, email',
  [username, email, hash]
);

res.status(201).json(result.rows[0]);
```

Рис. 2.7 Реєстрація нового користувача

Під час логіну система виконує перевірку відповідності введених облікових даних — електронної пошти та пароля — тим, що зберігаються у базі даних (Рис. 2.8). Спочатку здійснюється пошук користувача за вказаною електронною адресою. Якщо такий запис не знайдено, система одразу повертає повідомлення про помилку автентифікації. У разі, якщо користувача знайдено, наступним етапом є перевірка пароля: введений пароль порівнюється із захешованим значенням, що зберігається у базі, з використанням алгоритму bcrypt.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				49
Змн.	Арк.	№ докум.	Підпис	Дата		

```

const result = await pool.query('SELECT * FROM users WHERE email = $1', [email]);

if (result.rows.length === 0) return res.status(401).json({ error: 'Користувача не знайдено' });

const user = result.rows[0];
const match = await bcrypt.compare(password, user.password);
if (!match) return res.status(401).json({ error: 'Невірний пароль' });

const token = jwt.sign(
  { id: user.id, username: user.username, role: user.role },
  process.env.JWT_SECRET,
  { expiresIn: '1d' }
);

res.json({ token, user: { id: user.id, username: user.username, email: user.email } });

```

Рис. 2.8 Авторизація користувача та генерація JWT

Таким чином, забезпечується базовий рівень безпеки та контроль доступу до функціональності вебдодатку.

Створення кошторису відбувається після успішної авторизації, коли користувач отримує доступ до інтерфейсу на маршруті /create-estimate, що забезпечує зручне введення даних і формування документа. Цей інтерфейс підтримує динамічне додавання позицій та автоматичний підрахунок вартості. Цей функціонал дозволяє сформувати новий кошторис, обравши позиції з доступного каталогу та вказавши для них кількість і, за потреби, індивідуальні ціни. Окрім основних витрат, також реалізована можливість додати накладні витрати у відсотковому співвідношенні до загальної суми. Сума обчислюється автоматично, і користувач у реальному часі бачить кінцевий результат. Фрагмент коду (Рис. 2.9) нижче ілюструє створення нового кошторису на бекенді. Маршрут /api/estimates обробляє POST-запит і зберігає загальні параметри:

```

// Create estimate
router.post('/', async (req, res) => {
  const { title, description, total, overhead_percentage, overhead_amount } = req.body;
  const estimate = await Estimate.create({
    user_id: req.user.id,
    title,
    description,
    total,
    overhead_percentage,
    overhead_amount
  });
  res.status(201).json(estimate);
});

```

Рис. 2.9 Створення кошторису (бекенд)

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

Після заповнення усіх полів, користувач натискає кнопку збереження, що ініціює відповідний HTTP-запит на сервер. На сервері у відповідь створюється новий запис у таблиці `estimates`, а також автоматично формуються пов'язані записи в таблиці `estimate_items`, що відображають окремі пункти кошторису.

На клієнтській стороні відповідну логіку реалізовано у компоненті `CreateEstimate` (Рис. 2.10), де збираються всі введені користувачем дані та обчислюється фінальна сума. Після натискання кнопки «Зберегти кошторис» відбувається послідовне надсилання запитів — спочатку для створення самого кошторису, потім — для кожної обраної позиції:

```
const res = await API.post('/api/estimates', {
  title,
  description,
  total: grandTotal,
  overhead_percentage: overheadPercentage,
  overhead_amount: overheadAmount
});

const estimateId = res.data.id;
console.log('Створено кошторис з ID:', estimateId);

for (let item of items) {
  console.log('Відправляємо пункт:', item);
  await API.post(`/api/estimates/${estimateId}/items`, {
    item_id: item.id,
    quantity: item.quantity,
    custom_price: item.custom_price !== undefined && item.custom_price !== null ? item.custom_price : null
  });
}
```

Рис. 2.10 Створення кошторису (фронтенд)

Таким чином, реалізація створення кошторису охоплює як зручний інтерфейс введення та підрахунку на фронтенді, так і логіку збереження в базі даних на бекенді, що забезпечує цілісність і надійність процесу.

Редагування, видалення та експорт кошторису здійснюється через інтерфейс вкладки «Мої кошториси», що дозволяє користувачу змінювати вміст раніше створених кошторисів, видаляти непотрібні записи або експортувати документи у зручних форматах PDF та Excel. Цей функціонал забезпечує гнучкість у роботі та зручність подальшого використання кошторисної інформації. Вся відповідна логіка реалізована як на клієнтському, так і на серверному боці. Редагування кошторису передбачає можливість змінити його загальні характеристики (назву, опис, накладні витрати), а також оновити або видалити окремі позиції. Зміни надсилаються

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		51

через PUT-запит до API `/api/estimates/:id`. У разі успішного редагування система оновлює відповідні записи у базі даних. Для видалення кошторису використовується DELETE-запит за аналогічною адресою, який повністю вилучає кошторис та пов'язані з ним пункти з бази. Окремо реалізовано функцію експорту кошторису у два формати: PDF і Excel. Для цього створено відповідні маршрути, які обробляють GET-запити за ID кошторису та генерують документ динамічно на основі отриманих із бази даних даних.

Експорт у PDF реалізовано за допомогою бібліотеки `pdfkit`, яка дозволяє динамічно формувати структуровані документи з кошторисною інформацією у форматі PDF. Це забезпечує високу якість виводу, підтримку стилів та зручність друку або подальшої архівації. Після запиту до маршруту `/export-pdf/:estimateId` система витягує всі дані про кошторис та формує структурований документ: заголовок, дата, основна інформація, таблиця з позиціями, накладні витрати, фінальна сума та блок для підписів. Також додається логотип і декоративна рамка. Фрагмент коду (Рис. 2.11) нижче демонструє ключову логіку формування PDF-документа:

```
router.get('/export-pdf/:estimateId', async (req, res) => {
  try {
    const estimate = await getEstimateWithItems(req.params.estimateId);
    if (!estimate) return res.status(404).send('Кошторис не знайдено');

    const doc = new PDFDocument({ margin: 50 });
    doc.registerFont('Roboto', path.join(__dirname, '../fonts/Roboto-VariableFont_wdth,wght.ttf'));
    doc.font('Roboto');

    res.setHeader('Content-Disposition', `attachment; filename=estimate-${estimate.id}.pdf`);
    res.setHeader('Content-Type', 'application/pdf');
    res.setHeader('Cache-Control', 'no-store');

    doc.pipe(res);

    > // // Логотип ...

    doc.end();

    doc.on('finish', () => {
      console.log(`PDF для кошторису ${estimate.id} успішно згенеровано.`);
    });
  } catch (error) {
    console.error('PDF помилка:', error.stack || error);
    res.status(500).send('Помилка генерації PDF');
  }
});
```

Рис. 2.11 Експорт у PDF

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

Для експорту у форматі Excel застосовується бібліотека `exceljs`. У відповіді на запит до `/export-excel/:estimateId` система створює аркуш Excel, у якому зберігаються ті самі дані, що й у PDF: заголовок, таблиця з позиціями, накладні витрати, фінальна сума. Колонки отримують відповідні заголовки та стилізацію, а ширина колонок підлаштовується автоматично. Приклад коду (Рис. 2.12) з модуля експорту до Excel:

```
router.get('/export-excel/:estimateId', async (req, res) => {
  try {
    const estimate = await getEstimateWithItems(req.params.estimateId);
    if (!estimate) return res.status(404).send('Кошторис не знайдено');

    const workbook = new ExcelJS.Workbook();
    const sheet = workbook.addWorksheet(`Estimate ${estimate.id}`);

    // // Заголовок...
    res.setHeader('Content-Type', 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
    res.setHeader('Content-Disposition', `attachment; filename=estimate-${estimate.id}.xlsx`);

    await workbook.xlsx.write(res);
    res.end();
  } catch (error) {
    console.error('Excel помилка:', error.stack || error);
    res.status(500).send('Помилка генерації Excel');
  }
});
```

Рис. 2.12 Експорт у Excel

Така архітектура дозволяє зручно працювати з кошторисами на різних етапах життєвого циклу: від створення до остаточного збереження або передачі у вигляді документа. Завдяки використанню форматів PDF та Excel дані можуть бути інтегровані у звітність, роздруковані або переслані замовнику без потреби додаткової обробки.

Робота з формулами є одним із ключових функціональних блоків вебдодатку, що передбачає можливість створення, збереження та застосування математичних формул для виконання складних розрахунків. Це дозволяє користувачам автоматизувати обчислення та підвищує точність кошторисних оцінок. Цей модуль дозволяє користувачу створювати власні вирази для розрахунків, зберігати їх у базі даних та використовувати повторно у майбутньому. Після переходу до розділу «Мої формули» користувач може створити нову формулу, вказавши її назву та математичний вираз з використанням змінних. Всі збережені формули зберігаються в таблиці `user_formulas` й асоціюються з автором через `user_id`.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				53
Змн.	Арк.	№ докум.	Підпис	Дата		

Збереження формули здійснюється через форму, яка надсилає POST-запит до маршруту `/api/formulas` для збереження користувацького математичного виразу в базі даних. Це забезпечує можливість подальшого використання формул у розрахунках кошторису та калькуляторі. Сервер перевіряє отримані дані, а потім додає новий запис до бази даних. Фрагмент backend-коду (Рис. 2.13):

```
// POST: додати нову формулу
router.post('/formulas', authenticate, async (req, res) => {
  const userId = req.user.id;
  const { name, expression } = req.body;
  try {
    await db.query(
      'INSERT INTO user_formulas (user_id, name, expression) VALUES ($1, $2, $3)',
      [userId, name, expression]
    );
    res.status(201).send('Формулу додано');
    console.log('Отримано:', req.body);
  } catch (error) {
    console.error('Помилка формули:', error);
    res.status(500).send('Помилка додавання формули');
  }
});
```

Рис. 2.13 Збереження формули (бекенд)

Інтерфейс користувача включає компонент формули, який відображає назву формули, сам математичний вираз та поля для вводу змінних, що дозволяє користувачу швидко вводити дані та отримувати результати обчислень. При натисканні кнопки "Обчислити" користувач отримує результат прямо на сторінці. Це забезпечує гнучкість та інтерактивність без потреби звернення до зовнішніх інструментів. Таке рішення дозволяє автоматизувати обчислення у контексті кошторисів, де багато операцій мають повторювану математичну природу (наприклад, обрахунок об'єму, площі, довжини тощо).

Обчислення формули здійснюється шляхом вибору користувачем збереженої формули та введення необхідних значень змінних для отримання результату. Вираз парситься за допомогою регулярного виразу, і для кожної змінної, знайденої в рядку формули, відображається поле вводу. Обчислення проводиться на клієнтській стороні через `eval()`, попередньо підставивши всі значення. Візуалізація, логіка вводу та обчислення реалізовані в компоненті `FormulaCalculatorPage.jsx` (Рис. 2.14). Нижче наведено спрощений фрагмент обробки:

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		54

```
const calculate = () => {
  if (!formula) return;
  try {
    const expression = formula.expression;
    const filledExpression = expression.replace(/\b[a-zA-Z_]\w*\b/g, match => {
      if (variables[match] === undefined) throw new Error(`Змінна ${match} не задана`);
      return variables[match];
    });
    const evalResult = eval(filledExpression);
    setResult(evalResult);
  } catch (error) {
    console.error('Помилка обчислення:', error);
    message.error('Помилка обчислення формули');
  }
};
```

Рис. 2.14 Збереження формули (фронтенд)

Робота адміністратора

У системі реалізовано адміністративну панель, яка надає можливість керування користувачами, каталогами та пунктами, що використовуються у кошторисах. Ця панель доступна лише авторизованим користувачам із роллю адміністратора та дозволяє централізовано підтримувати актуальність і коректність даних, що використовуються іншими користувачами системи.

Адміністратор має можливість додавати нових користувачів, вказуючи їх ім'я, email, пароль і роль. Після збереження нового запису користувач автоматично з'являється у таблиці. Також доступна функція редагування — при натисканні кнопки відкривається модальне вікно, у якому адміністратор може змінити дані користувача. Крім того, адміністратор може видалити обліковий запис разом з усіма кошторисами, що належать відповідному користувачу. Для зручності реалізовано пошук по імені користувача, що дозволяє швидко знайти потрібний запис.

Окрім керування користувачами, адміністратор може створювати нові каталоги, які об'єднують споріднені пункти кошторисів за типом робіт чи матеріалів. Каталоги додаються у модальному вікні, яке відкривається при натисканні кнопки «Додати каталог». У вже створених каталогів є можливість змінювати назву або видалити їх повністю.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

Також реалізовано керування пунктами кошторису. Кожен пункт містить назву, одиницю виміру, ціну та належить до певного каталогу. У адміністративній панелі можна додати новий пункт, редагувати його властивості або видалити. Всі ці дії також виконуються через зручні форми, які з'являються у модальних вікнах. Для полегшення вибору каталогу використовується випадаючий список, що автоматично заповнюється доступними варіантами.

```

<Form
  form={addingForm}
  layout="inline"
  onFinish={onAddUserFinish}
  style={{ marginBottom: 24 }}
>
  <Form.Item
    name="username"
    rules={[{ required: true, message: 'Введіть ім'я' }]}
  >
    <Input placeholder="Ім'я" />
  </Form.Item>
  <Form.Item
    name="email"
    rules={[{ required: true, message: 'Введіть email' }]}
  >
    <Input placeholder="Email" />
  </Form.Item>
  <Form.Item
    name="password"
    rules={[{ required: true, message: 'Введіть пароль' }]}
  >
    <Input.Password placeholder="Пароль" />
  </Form.Item>
  <Form.Item
    name="role"
    rules={[{ required: true, message: 'Оберіть роль' }]}
  >
    <Select
      placeholder="Оберіть роль"
      style={{ width: 150 }}
      options={roleOptions}
    />
  </Form.Item>
  <Form.Item>
    <Button type="primary" htmlType="submit">
      Додати
    </Button>
  </Form.Item>
</Form>

```

Рис. 2.15 Форма для додавання користувача

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				56
Змн.	Арк.	№ докум.	Підпис	Дата		

Уся логіка побудована так, щоб після виконання будь-якої дії (додавання, редагування, видалення) таблиці автоматично оновлювались. Це дозволяє адміністратору оперативно бачити зміни без перезавантаження сторінки. Таким чином, реалізована панель дозволяє підтримувати чистоту даних, контроль над користувачами та забезпечує гнучке налаштування структури системи.

Діаграма компонентів системи відображає архітектурну структуру вебдодатку, розділену на логічно незалежні частини, які виконують конкретні функціональні завдання. Такий поділ дозволяє досягти високої модульності коду, спростує його підтримку, масштабування, а також забезпечує розділення відповідальності між частинами системи.

Ключовим компонентом є Auth Module, який відповідає за процеси реєстрації, авторизації та автентифікації користувачів. Саме цей модуль забезпечує перевірку облікових даних, генерує токени доступу (JWT) та контролює захищений доступ до внутрішніх ресурсів додатку. Він працює у зв'язці з базою даних, у якій зберігаються облікові записи користувачів.

Інший важливий компонент — Estimate Module. Він реалізує повний цикл роботи з кошторисами: створення, редагування, збереження та перегляд. Модуль взаємодіє з таблицями бази даних estimates та estimate_items, дозволяючи формувати деталізовану структуру кожного кошторису з обраними позиціями.

Export Module реалізує функціональність експорту даних у зовнішні формати, зокрема PDF та Excel. Завдяки йому користувач може отримати кошторис у вигляді зручного файлу, придатного для друку або подальшого використання в офлайн-середовищі. У цьому модулі використовуються бібліотеки для генерації документів (наприклад, PDFKit, ExcelJS або Puppeteer).

Функціонал для роботи з формулами представлений у Formula Module. Він дозволяє користувачам створювати власні математичні вирази, зберігати їх у базі даних, а також виконувати обчислення зі змінними, які вводяться вручну. Цей компонент є прикладом гнучкого розширення системи під специфічні задачі користувача, зокрема розрахунки у сфері будівництва чи виробництва.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		57

Admin Module призначений для адміністраторів системи. Через нього відбувається управління каталогами (групами робіт або матеріалів), окремими пунктами (items), а також користувачами. Модуль дозволяє виконувати CRUD-операції (створення, читання, оновлення, видалення) через інтерфейс, доступний лише авторизованим адміністраторам. Таким чином забезпечується централізований контроль за наповненням бази та структурою системи.

Кожен з компонентів взаємодіє з іншими через чітко визначені API-ендпоінти, що дозволяє підтримувати принципи розділення відповідальності та забезпечує масштабованість рішення.

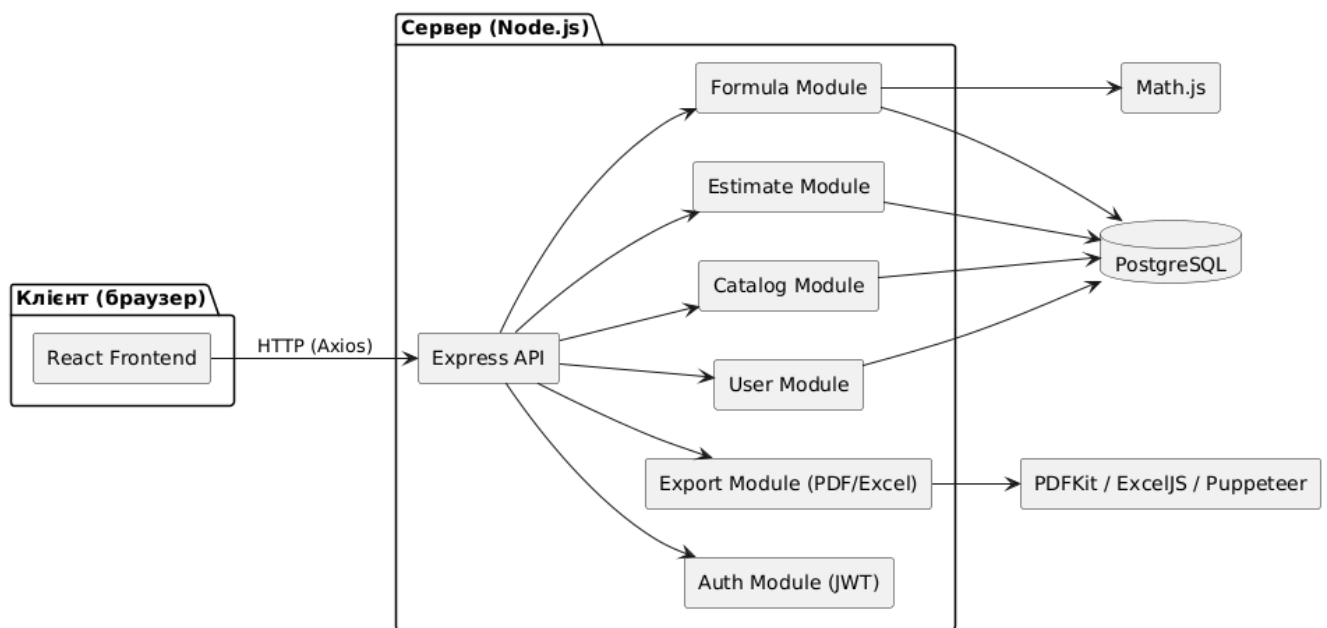


Рис. 2.16 Діаграма компонентів

Таким чином, реалізований вебдодаток покриває всі ключові функції: створення кошторисів, облік ресурсів, математичні розрахунки, експорт документації, підтримку ролей користувачів. Архітектура додатку є модульною монолітною, що дозволяє розширювати функціональність без порушення основної структури, забезпечуючи масштабованість, підтримку та простоту тестування.

Висновки до другого розділу

У другому розділі дипломної роботи здійснено комплексне проєктування вебдодатку для створення та управління кошторисами. В межах підрозділу 2.1 було визначено основні варіанти використання системи з урахуванням ролей користувачів — кошторисника, адміністратора та інших потенційних учасників. Побудовано

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		58

діаграму варіантів використання, яка дозволяє формалізувати основні функції додатку на рівні взаємодії з користувачем. Також розроблено діаграму класів, що відображає об'єктно-орієнтовану структуру системи: класи User, Estimate, EstimateItem, Item, Catalog, Formula, їх атрибути, методи та зв'язки між ними.

Також було спроектовано базу даних, яка складається з шести основних таблиць (users, estimates, estimate_items, items, catalogs, user_formulas). Детально описано кожну таблицю, її поля, типи даних та функціональне призначення. Така структура дозволяє зберігати повноцінну інформацію про кошториси, їх складові елементи, користувацькі формули та взаємозв'язки між сутностями системи.

Було проведено проектування алгоритмів роботи додатку на основі типових сценаріїв взаємодії. Для цього створено набір UML-діаграм активностей, які демонструють логіку виконання основних процесів у системі: створення кошторису, редагування та експорт у PDF/Excel, створення формул, робота адміністратора з таблицями користувачів, позицій та каталогів. Візуальне моделювання дозволило структурувати внутрішню логіку платформи, зробити її більш зрозумілою та придатною до подальшої реалізації.

А також було безпосередньо описано реалізацію функціоналу вебдодатку. Розглянуто процеси реєстрації, створення та редагування кошторису, інтеграцію математичних формул для обчислень, а також реалізацію експорту даних у зручні формати. Представлено приклади клієнтського та серверного коду, а також пояснено принцип роботи з базою даних, API та бібліотеками, що використовуються (зокрема pdfkit, exceljs, math.js). Продемонстровано діаграму компонентів системи, яка відображає логічну структуру модулів вебдодатку.

У результаті, спроектована система відповідає вимогам до сучасного інструменту для складання кошторисної документації. Вона забезпечує зручність, функціональність і масштабованість, що дозволяє ефективно використовувати її як у будівельній сфері, так і в інших галузях, де потрібен точний облік вартості матеріалів, робіт і розрахунків.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				59
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ВЕБДОДАТКОМ ДЛЯ СКЛАДАННЯ КОШТОРИСІВ

3.1 Порядок встановлення та налаштування параметрів системи

Вебдодаток для складання кошторисів складається з кількох ключових компонентів, кожен з яких потребує окремого налаштування й розгортання, але при цьому всі елементи інтегруються в єдину цілісну систему. Її архітектура передбачає серверну частину, клієнтський інтерфейс та систему управління базами даних, які взаємодіють між собою через HTTP-запити та API.

Перший етап підготовки платформи полягає в налаштуванні серверної частини. Серверна логіка реалізована на основі технології Node.js у поєднанні з фреймворком Express.js. Для підключення до бази даних використовується ORM-бібліотека Sequelize, яка дозволяє описати структуру таблиць на рівні коду, спростивши процес створення та оновлення схеми бази даних. Перед запуском сервера необхідно створити файл середовища .env, де зазначаються всі конфігураційні параметри — адреса бази даних PostgreSQL, секретний ключ для JWT-токенів, порт запуску та інші важливі налаштування. Після цього виконується ініціалізація схеми бази даних шляхом запуску міграцій, що створюють таблиці для зберігання користувачів, кошторисів, формул, позицій і довідників.

Клієнтська частина системи реалізована на основі бібліотеки React. Вона відповідає за візуальну частину вебінтерфейсу та забезпечує взаємодію користувача з основним функціоналом додатку. Інтерфейс включає компоненти для створення й редагування кошторисів, навігації по проєктах, перегляду формул і керування обліковими записами. Для обміну даними з сервером використовується бібліотека Axios, яка забезпечує виконання HTTP-запитів до API бекенду. Після завершення розробки клієнтська частина також розгортається на платформі Render або будь-якому іншому хмарному хостингу, де вона стає доступною для користувачів через веббраузер. База даних є критично важливим компонентом системи, що зберігає всю ключову інформацію: від облікових записів користувачів до структурованих кошторисів та математичних формул. В якості СУБД використовується

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				60
Змн.	Арк.	№ докум.	Підпис	Дата		

PostgreSQL. У структурі бази передбачено шість основних таблиць, які мають зв'язки між собою через зовнішні ключі. Наприклад, таблиця з кошторисами зберігає зв'язок із користувачем, що його створив, а таблиця з пунктами кошторису — зв'язок із довідником позицій. Всі зміни в структурі таблиць фіксуються через Sequelize, що спрощує подальші оновлення структури при розгортанні нових версій.

У процесі розгортання системи важливо забезпечити правильне узгодження між клієнтською та серверною частинами. Параметри запитів, шляхи до API, конфігурація безпеки та авторизації мають відповідати одна одній, щоб система функціонувала стабільно й надійно. Крім цього, уся система підтримує модульну архітектуру, що дозволяє легко масштабувати її при збільшенні навантаження або додаванні нового функціоналу.

На завершення етапу розгортання формується цілісна, взаємопов'язана система, готова до роботи з користувачами. Для візуалізації взаємозв'язків між її компонентами, була створена діаграма розгортання, яка демонструє, як саме розміщені й взаємодіють між собою клієнт, сервер і база даних у рамках проєкту. Ця діаграма наведена на (Рис. 3.1).

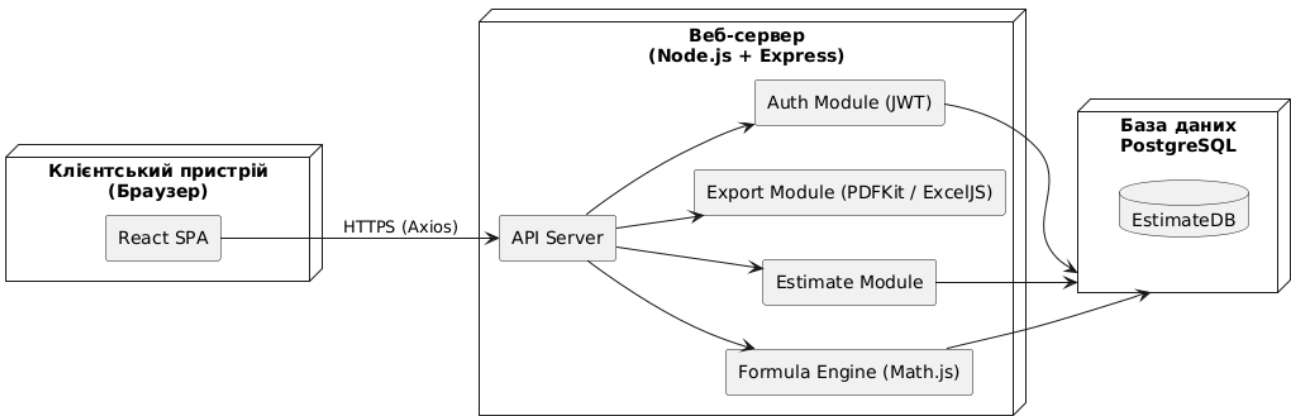


Рис. 3.1. Діаграма розгортання вебдодатку для складання кошторисів

Завдяки чітко визначеному порядку налаштування і розгортання платформи, система забезпечує стабільну роботу, високий рівень безпеки та доступність усіх функцій для кінцевих користувачів, включаючи адміністраторів, кошторисників та інших спеціалістів, що працюють з обліком витрат у проєктах.

3.2 Структура інтерфейсу вебдодатку для складання кошторису

Інтерфейс користувача у вебдодатку для створення та управління кошторисами реалізований за допомогою бібліотеки React, що дозволило забезпечити гнучку й адаптивну систему взаємодії користувача з функціоналом платформи. Головна мета інтерфейсу — зробити процес формування кошторисної документації максимально інтуїтивним, зрозумілим і доступним як для новачків, так і для досвідчених фахівців.

При відкритті додатку користувач потрапляє на сторінку авторизації (Рис. 3.2), як користувач ще не створював акаунт, то може зареєструватись, перейшовши на сторінку реєстрації (Рис. 3.3).

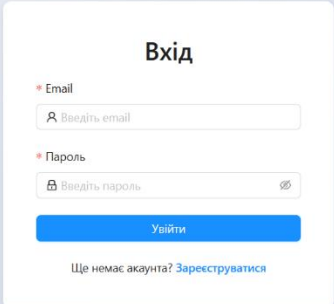
The image shows a login form titled "Вхід" (Login) centered on a light blue background. The form has two input fields: "Email" with a magnifying glass icon and "Пароль" (Password) with a lock icon and a toggle for visibility. Below the fields is a blue button labeled "Увійти" (Login). At the bottom, there is a link that says "Ще немає акаунта? Зареєструватися" (Don't have an account? Register).

Рис. 3.2 Сторінка авторизації

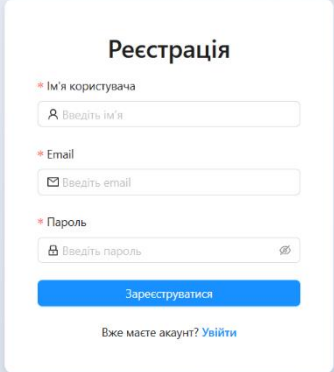
The image shows a registration form titled "Реєстрація" (Registration) centered on a light blue background. The form has three input fields: "Ім'я користувача" (Username) with a magnifying glass icon, "Email" with an envelope icon, and "Пароль" (Password) with a lock icon and a toggle for visibility. Below the fields is a blue button labeled "Зареєструватися" (Register). At the bottom, there is a link that says "Вже маєте акаунт? Увійти" (Already have an account? Login).

Рис. 3.3 Сторінка реєстрації

Після входу система автоматично визначає його роль — адміністратор або звичайний користувач — і відповідно адаптує доступний функціонал (Рис. 3.4). Основна навігація реалізована через бокове меню, звідки можна швидко перейти до створення нового кошторису, перегляду вже створених документів, редагування позицій, відкриття калькулятора фарбування труб або формул.

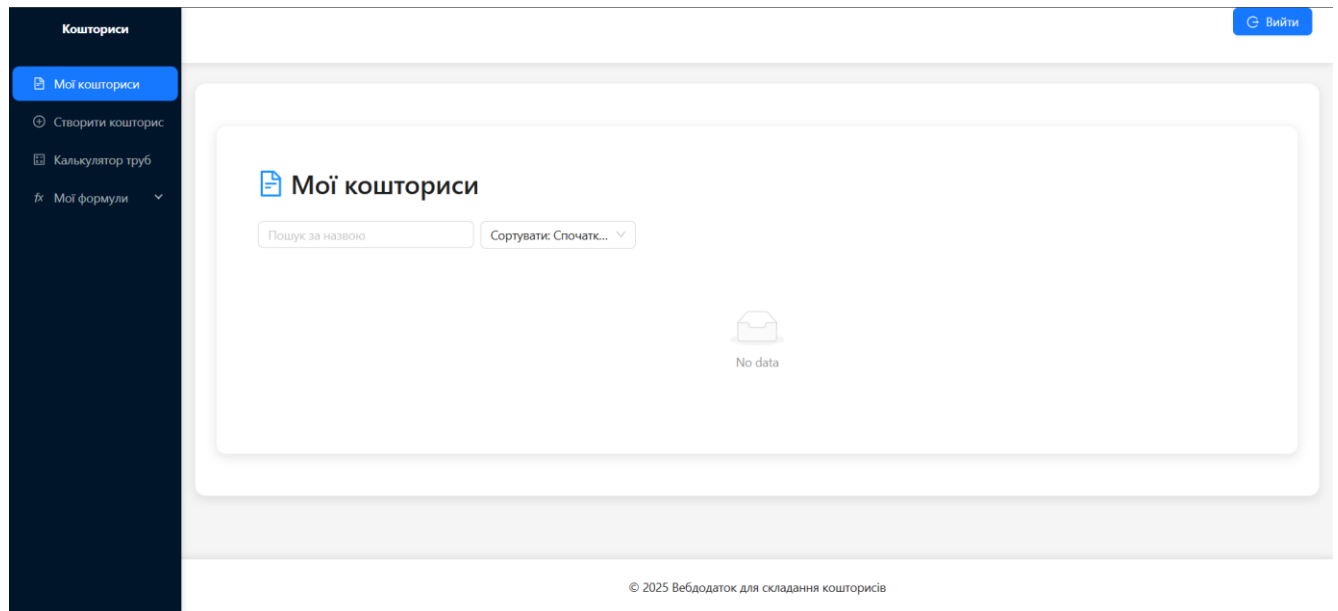


Рис. 3.4 «Головна сторінка» з боковим меню

Окремо варто звернути увагу на сторінку створення кошторису (Рис. 3.5). Вона містить зручну форму, де користувач поетапно вводить основні дані: назву, опис, вибирає необхідні пункти з каталогу та вказує їх кількість.

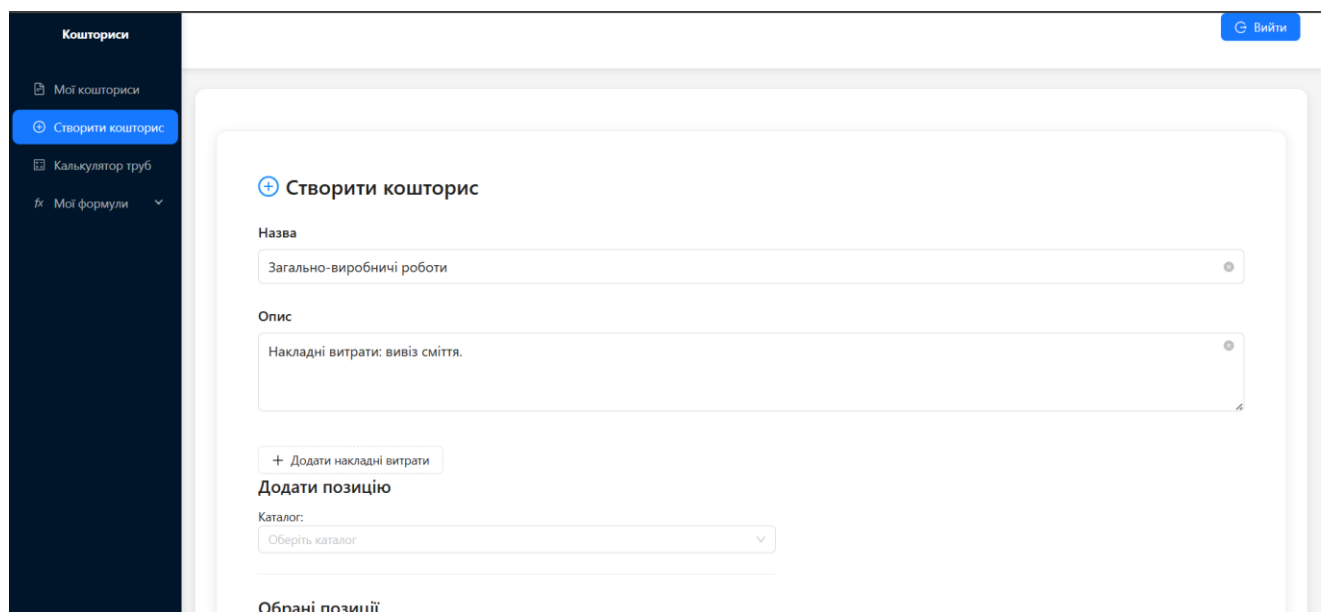


Рис. 3.5 Сторінка «Створення кошторису» без додавання накладних витрат

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		63

Також варто зазначити, що користувач може додавати накладні витрати, а може не додавати, на (Рис. 3.5) є приклад того, що накладні витрати відсутні, тому на (Рис. 3.6) можна побачити як вони відображатимуться, якщо користувач захоче їх додати.

The screenshot shows the 'Створити кошторис' (Create Estimate) page. On the left is a dark sidebar with navigation links: 'Мої кошториси', 'Створити кошторис' (highlighted), 'Калькулятор труб', and 'Мої формули'. The main content area has a header 'Створити кошторис' with a plus icon. Below it are fields for 'Назва' (Name) with the value 'Загально-виробничі роботи' and 'Опис' (Description) with the value 'Накладні витрати: вивіз сміття'. There is a field for 'Накладні витрати (% від загальної суми):' with the value '10' and a '%' symbol. Below this is the 'Додати позицію' (Add position) section, which includes a 'Каталог:' (Catalog) dropdown menu currently showing 'Оберіть каталог'.

Рис. 3.6 Сторінка «Створення кошторису» з додаванням накладних витрат
Далі користувач обирає каталог , з якого потім буде обирати позицію або пункт який записується в сам кошторис (Рис. 3.7).

This screenshot shows the 'Додати позицію' (Add position) section of the 'Create Estimate' page. The 'Каталог:' dropdown menu is open, displaying a list of items: 'Земляні роботи', 'Бетонні роботи', 'Монтаж труб', 'Оздоблювальні роботи', 'Електромонтажні роботи', 'Опалення та вентиляція' (highlighted), 'Сантехнічні роботи', and 'Покрівельні роботи'. To the right of the list, there is a table with two columns: 'Сума' (Sum) and 'Видалити' (Delete). The first row shows '0.00 грн' and a red 'Видалити' button. Below the list, it says 'Накладні витрати: 0.00 грн'. At the bottom, there is a 'Фінальна сума: 0.00 грн' and a blue 'Зберегти кошторис' (Save estimate) button.

Рис. 3.7 Сторінка «Створення кошторису» вибір каталогу

Коли користувач обрав каталог, тоді з'являється поле з пунктом, де можна обрати запропоновану позицію для кошторису (Рис. 3.8).

Додати позицію

Каталог: Опалення та вентиляція

Пункт: Оберіть пункт

Кількість: 1

Додати

Обрані позиції

Назва	Ціна	Кількість	Сума	
Накладні витрати	—	—	0.00 грн	Видалити

Сума без накладних: 0.00 грн

Накладні витрати: 0.00 грн

Фінальна сума: 0.00 грн

Рис. 3.8 Сторінка «Створення кошторису» з полем Пункт

В полі Пункт користувач може обрати запропоновані позиції кошторису, поруч з назвою одразу є ціна з одиницею виміру (Рис 3.9).

Додати позицію

Каталог: Опалення та вентиляція

Пункт: Оберіть пункт

- Установка котла — 2338.87 грн/м
- Монтаж радіаторів — 1762.10 грн/м²
- Монтаж радіаторів — 286.92 грн/м
- Установка котла — 1227.82 грн/м³
- Установка котла — 443.56 грн/шт
- Установка котла — 2013.84 грн/м
- Монтаж радіаторів — 1658.09 грн/м³
- Монтаж вентиляційних каналів — 1611.37 грн/шт

Кількість:

Додати

Обрані позиції

Назва	Ціна	Кількість	Сума	
Накладні витрати	—	—	0.00 грн	Видалити

Сума без накладних: 0.00 грн

Накладні витрати: 0.00 грн

Фінальна сума: 0.00 грн

Рис. 3.9 Сторінка «Створення кошторису» вибір пункту

Після того, коли каталог та пункт обраний користувачем з'являється поле з ціною, а також є поле кількість. Поле з ціною є для того, щоб користувач мав можливість змінювати ціну за ту чи іншу позицію в своєму кошторисі, це зручно та

універсально, а також дуже важливо при створенні кошторисів. Також поле з кількістю логічно для того аби користувач міг ввести потрібну кількість матеріалу, пункту або позиції (Рис. 3.10).

Додати позицію

Каталог: Опалення та вентиляція

Пункт: Монтаж радіаторів — 1762.10 грн/м²

Ціна (можна змінити): 1762.1

Кількість: 19

Додати

Обрані позиції

Назва	Ціна	Кількість	Сума	
Накладні витрати	—	—	0.00 грн	Видалити

Сума без накладних: 0.00 грн

Накладні витрати: 0.00 грн

Рис. 3.10 Сторінка «Створення кошторису» додавання позиції

Після внесення всіх даних доступна функція автоматичного підрахунку загальної вартості кошторису, яка оновлюється в реальному часі (Рис. 3.11).

Додати позицію

Каталог: Підлогові покриття

Пункт: Оберіть пункт

Кількість: 1

Додати

Обрані позиції

Назва	Ціна	Кількість	Сума	
Монтаж радіаторів	1762.10 грн	19	33479.90 грн	Видалити
Накладні витрати	—	—	3347.99 грн	Видалити

Сума без накладних: 33479.90 грн

Накладні витрати: 3347.99 грн

Фінальна сума: 36827.89 грн

Зберегти кошторис

Рис. 3.11 Сторінка «Створення кошторису» загальна таблиця з сумами

Користувач також може додавати наступні позиції, обираючи з потрібний каталог та пункти з нього, налаштовуючи ціну та кількість (Рис. 3.12).

Обрані позиції				
Назва	Ціна	Кількість	Сума	
Монтаж радіаторів	1762.10 грн	19	33479.90 грн	Видалити
Заливка стяжки	1405.61 грн	17	23895.37 грн	Видалити
Установка металокаркасу	847.79 грн	12	10173.48 грн	Видалити
Монтаж колон	2384.99 грн	166	395908.34 грн	Видалити
Накладні витрати	—	—	46345.71 грн	Видалити
Сума без накладних: 463457.09 грн				
Накладні витрати: 46345.71 грн				
Фінальна сума: 509802.80 грн				
Зберегти кошторис				

© 2025 Вебдодаток для складання кошторисів

Рис. 3.12 Сторінка «Створення кошторису» загальна таблиця з кількома позиціями

Після чого користувач зберігає свій кошторис та може перейти на сторінку «Мої кошториси», де зможе побачити вже свій створений кошторис (Рис. 3.13).

Кошториси					Вийти
Мої кошториси Створити кошторис Калькулятор труб Мої формули					
Пошук за назвою Сортувати: Спочатк... Загально-виробничі роботи — 509802.80 грн Накладні витрати: вивіс сміття. Редагувати Експорт PDF Експорт Excel Видалити					
© 2025 Вебдодаток для складання кошторисів					

Рис. 3.13 Сторінка «Мої кошториси» список всіх кошторисів користувача

На сторінці «Мої кошториси» користувач має можливість редагувати вже створені кошториси (Рис. 3.14). Для прикладу можна змінити відсоток накладних витрат, з 10 % збільшити до 17%, тоді одразу видно як змінюються суми кошторису, а особливо фінальна або загальна сума кошторису.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		67

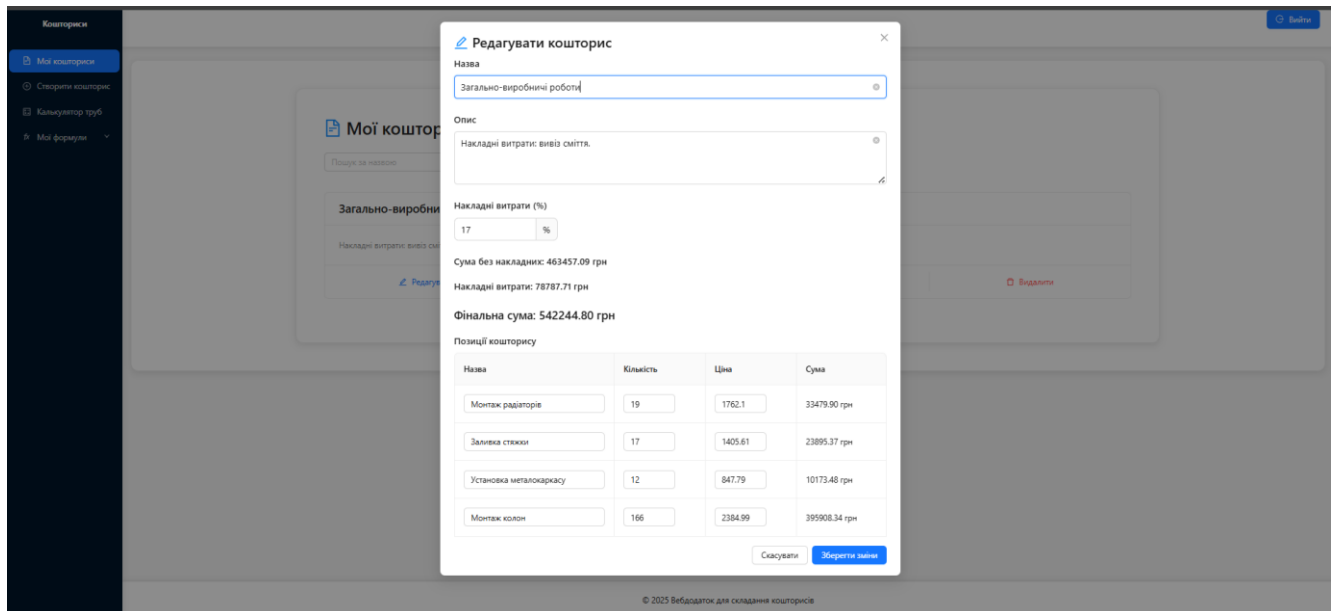


Рис. 3.14 Сторінка «Мої кошториси» редагування кошторису

Наступною не менш важливою можливістю є експортування кошторису в Pdf формат. Структура документу зберігається у вигляді табличних блоків з відповідними підписами, датою створення, логотипом та підсумковою інформацією (Рис. 3.15).

Кошторис №5

Дата створення: 25 червня 2025 р.

Назва: Загально-виробничі роботи

Опис: Накладні витрати: вивіз сміття.

№	Назва	Кількість	Ціна	Сума
1	Монтаж радіаторів	19	1762.10	33479.90
2	Заливка стяжки	17	1405.61	23895.37
3	Установка металокаркасу	12	847.79	10173.48
4	Монтаж колон	166	2384.99	395908.34

Накладні витрати (17%): 92181.62 грн

Загальна сума: 542244.80 грн

Склав:

Затвердив:

Рис. 3.15 Експортований Pdf-файл з кошторисом користувача

		Горбачевська Д.А.			ДУ «Житомирська політехніка». 25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		68

Також реалізовано експортування в Excel для того, аби можна було постійно редагувати або додавати вже потрібну інформацію для користувача. Всі колонки в Excel є адаптивного розміру, що є зручним при сприйнятті інформації (Рис. 3.16). В двох документах, можемо прослідкувати, що наша зміна 17% накладних витрат успішно зберіглась.

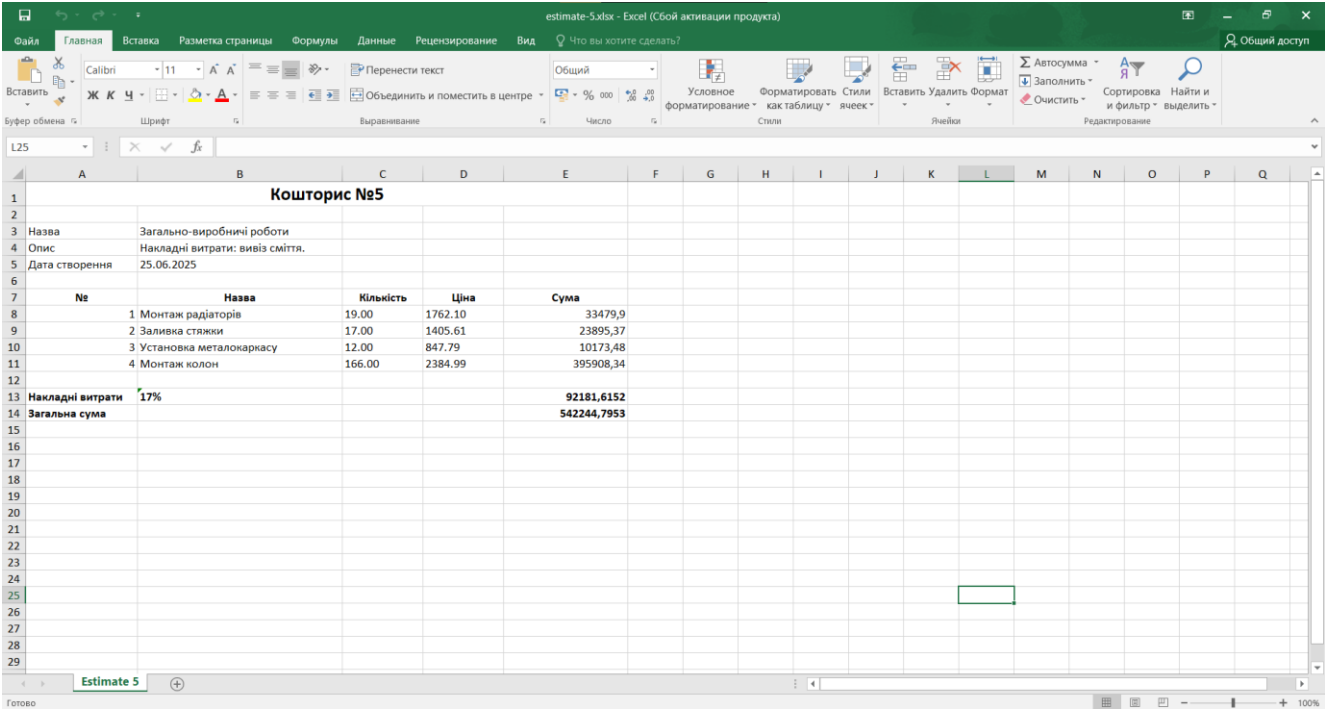


Рис. 3.16 Експортований Excel-файл з кошторисом користувача

А також є можливість видалити створений кошторис користувача (Рис. 3.17).

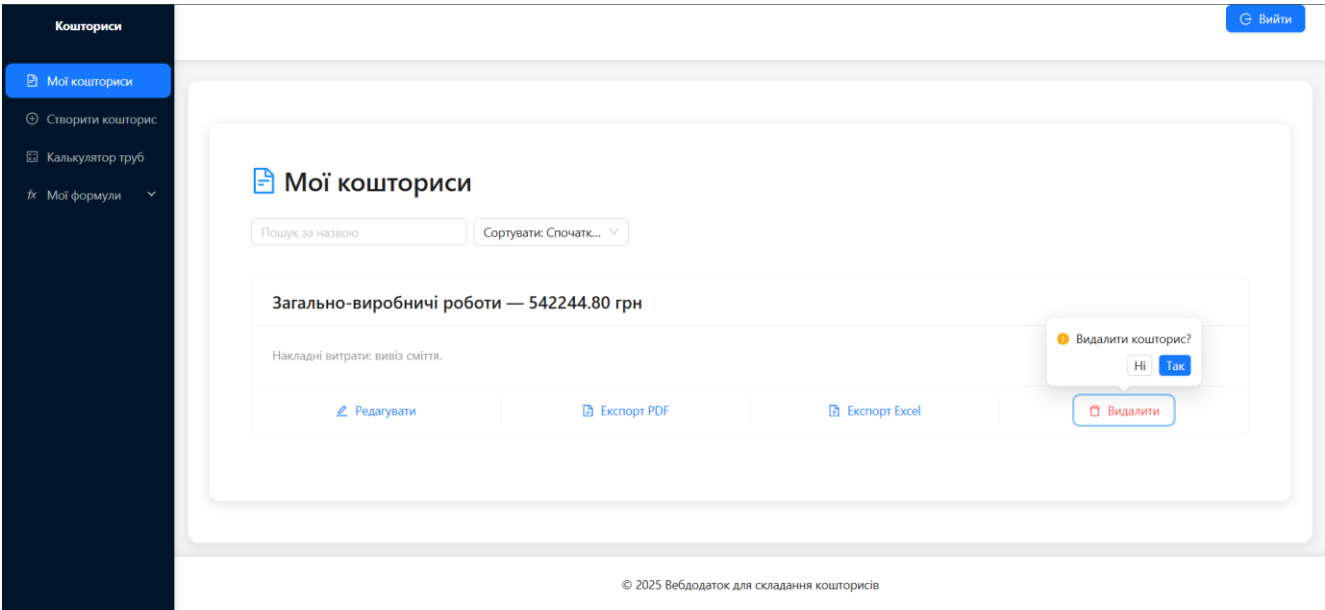


Рис. 3.17 Видалення кошторису користувача

Вебдодаток також містить вбудований калькулятор, який визначає площу труб, зазвичай ця площа обраховується, коли потрібно пофарбувати труби, саме тому ця сторінка називається «Калькулятор фарбування труб» (Рис. 3.18).

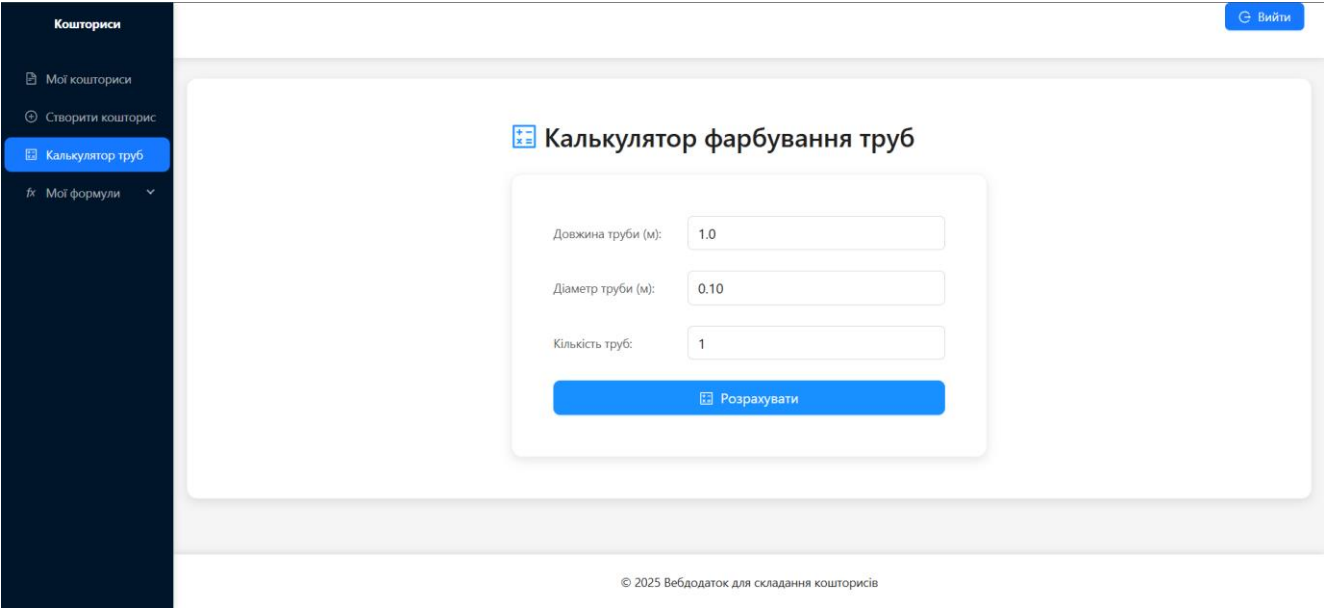


Рис. 3.18 Калькулятор фарбування труб

Після введення даних користувач отримає готовий результат обрахунку(Рис. 3.19).

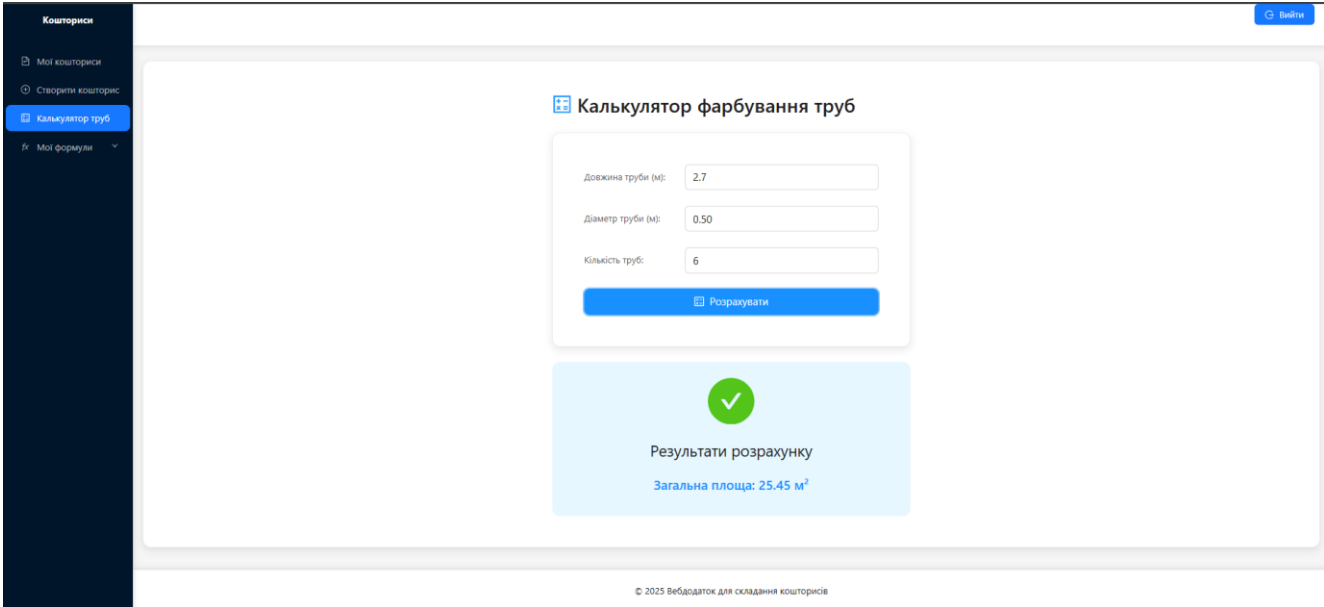


Рис. 3.19 «Калькулятор фарбування труб» готові обрахунки

Особливої уваги заслуговує інтерфейс для роботи з математичними формулами (Рис. 3.20). Користувач може створити власну формулу, описавши її назву та математичний вираз (Рис. 3.21), зберегти її у своєму кабінеті (Рис. 3.22), а також перейти на сторінку обчислення (Рис. 3.23). Там за допомогою динамічної форми

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

користувач вводить значення змінних, які автоматично підтягуються з формули, після чого система обчислює результат (Рис. 3.24). Це дозволяє розширити можливості додатку — зокрема для інженерних розрахунків або побудови складних витратних моделей.

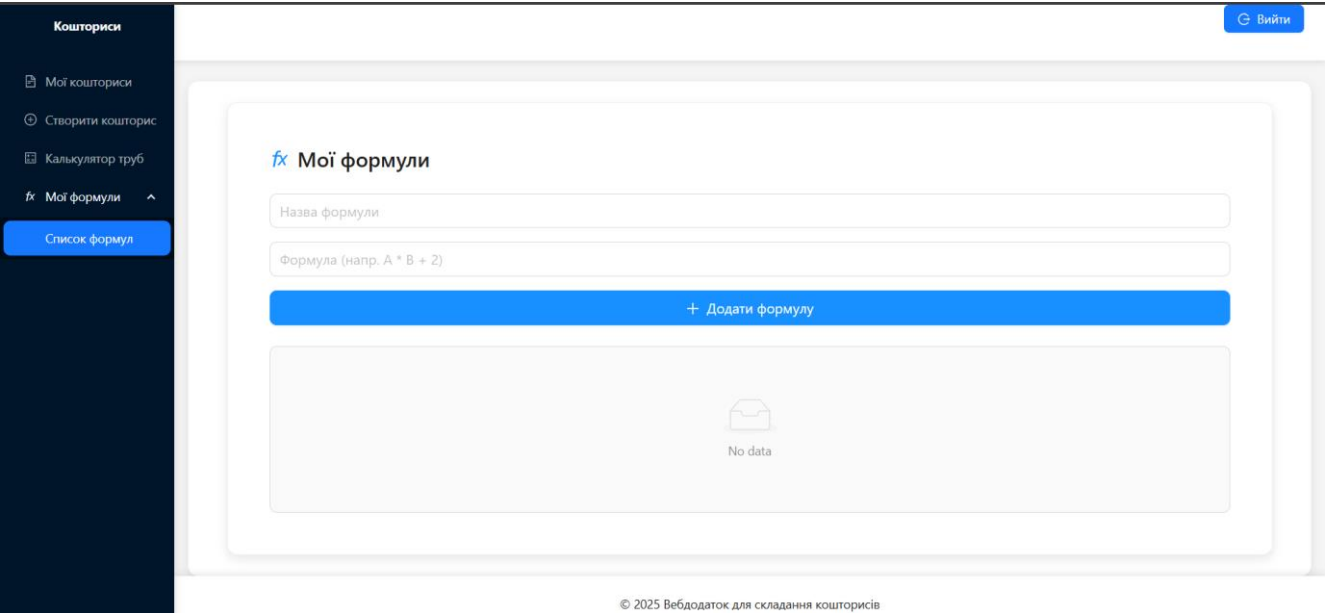


Рис. 3.20 Сторінка «Мої формули»

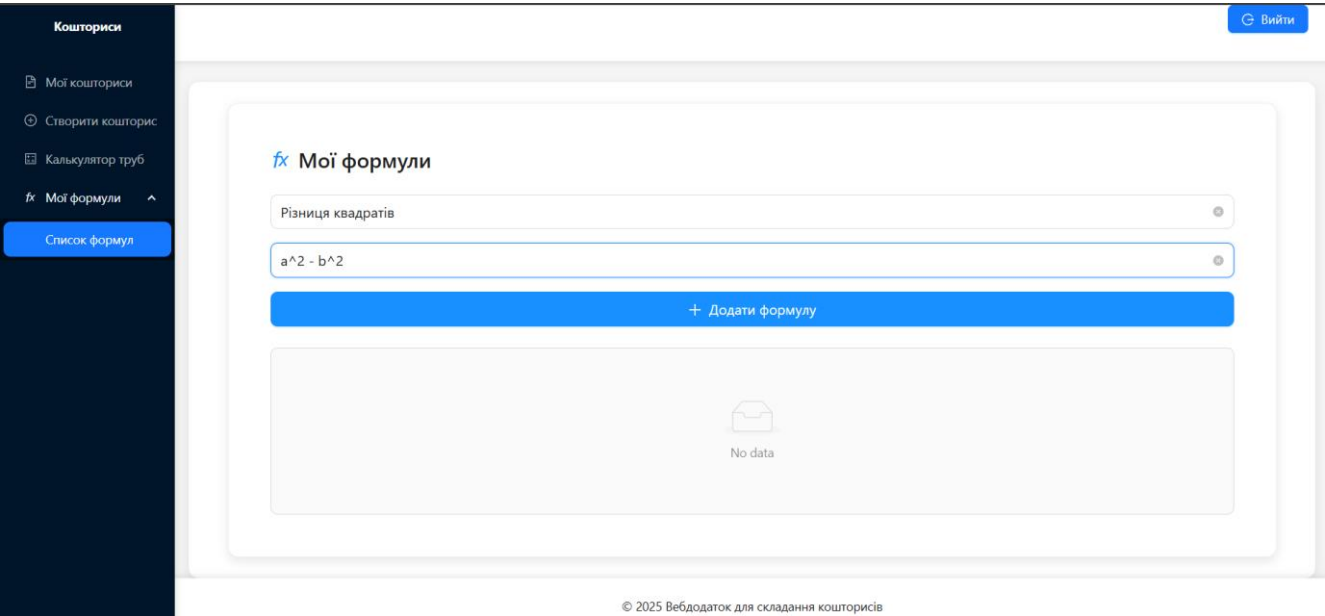


Рис. 3.21 Сторінка «Мої формули» створення власної формули «Різниця квадратів»

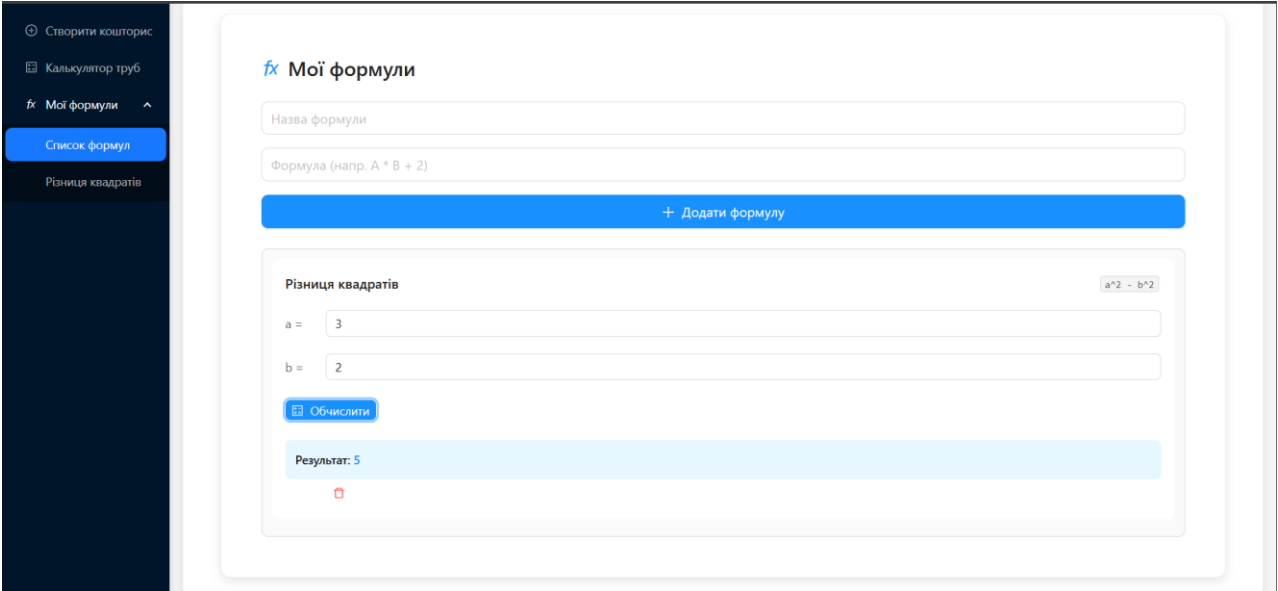


Рис. 3.22 Сторінка «Мої формули» додана формула

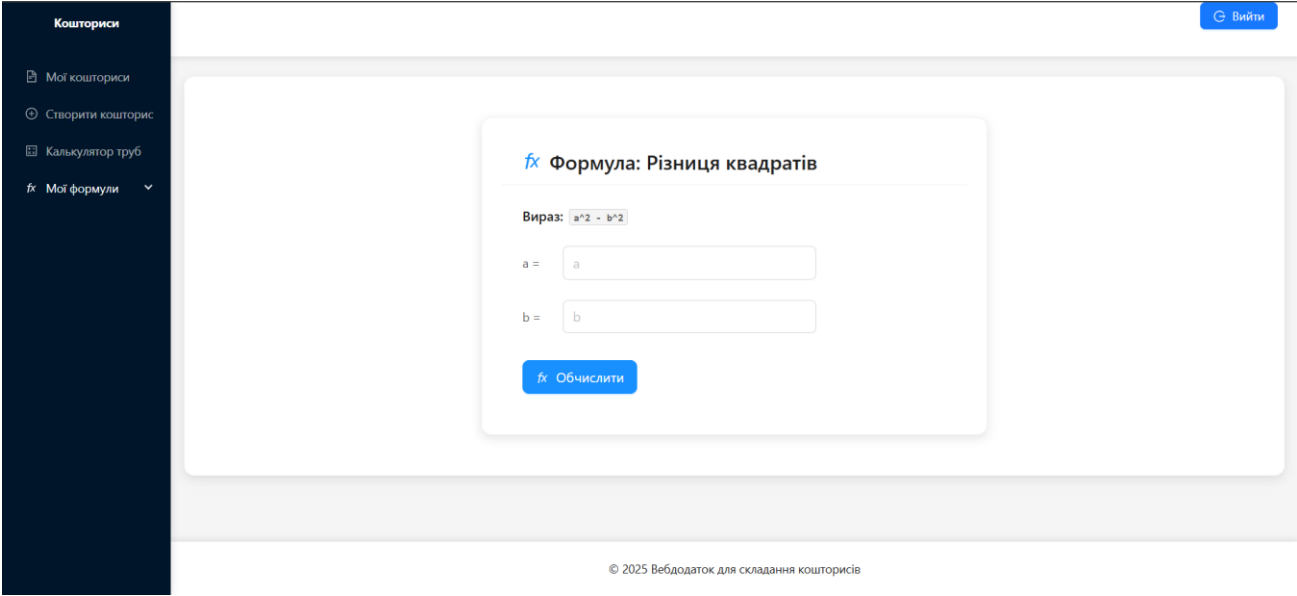


Рис. 3.23 Сторінка «Різниця квадратів»

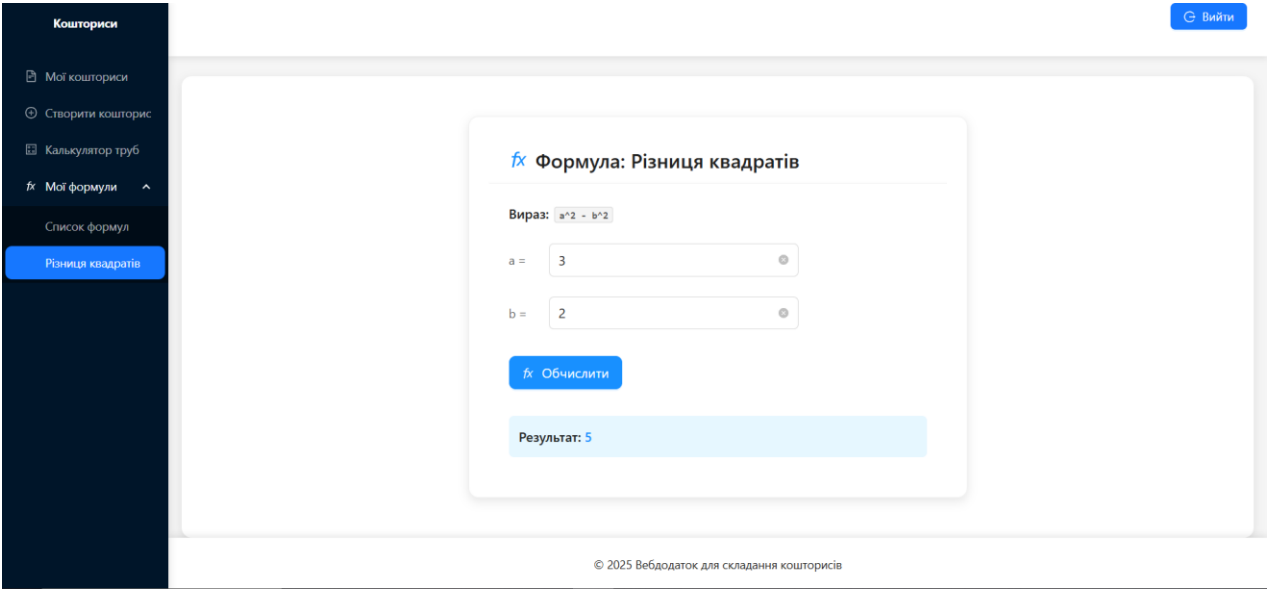


Рис. 3.24 Сторінка «Різниця квадратів» з обчисленнями

У свою чергу, адміністратор після авторизації потрапляє в адміністративну панель, яка відкриває доступ до таблиць з усіма зареєстрованими користувачами, створеними кошторисами, каталогами й позиціями (Рис. 3.25).

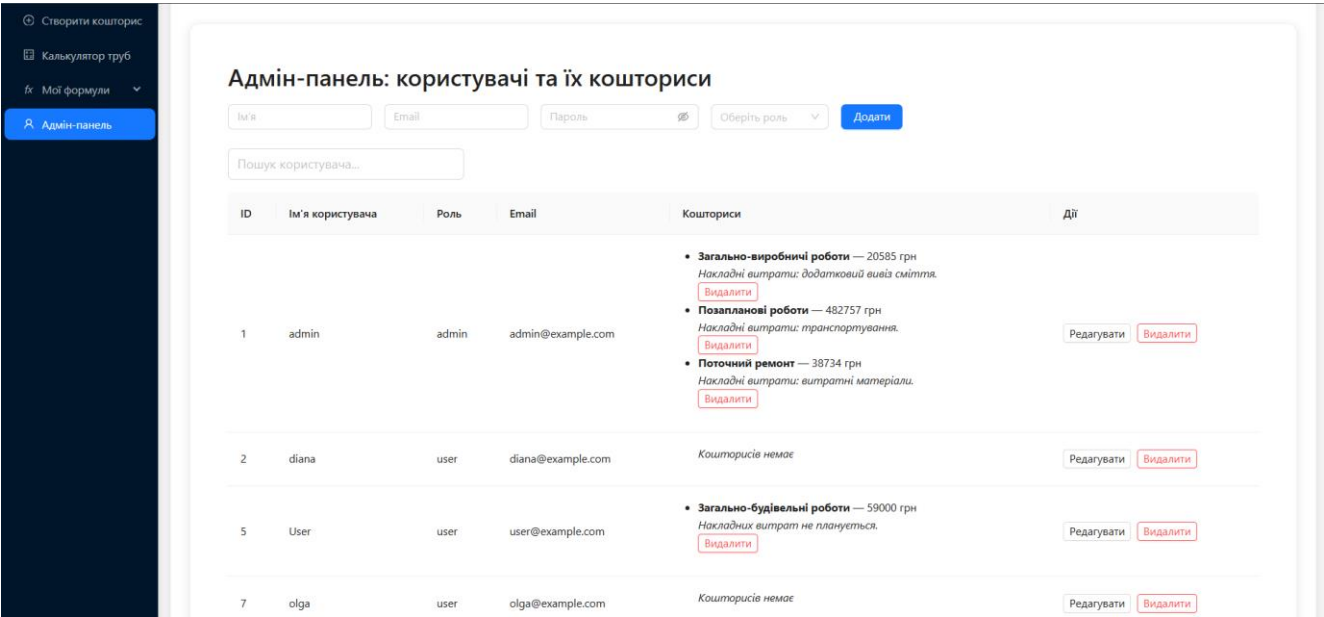


Рис. 3.25 Адмін-панель таблиця з користувачами

Адміністратор може переглядати й редагувати користувачів (Рис.3.26) — змінювати їхні ролі, видаляти неактивні акаунти (Рис. 3.27) або надавати права адміністратора, а також може видаляти кошториси користувачів (Рис. 3.28).

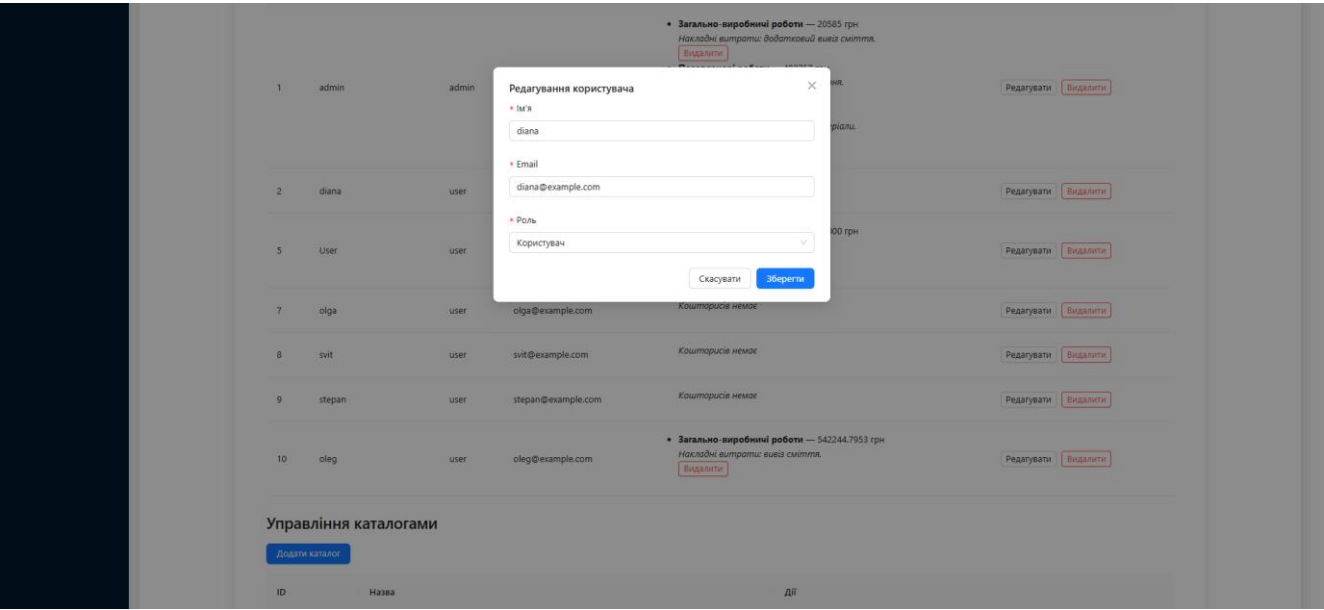


Рис. 3.26 Адмін-панель редагування користувача

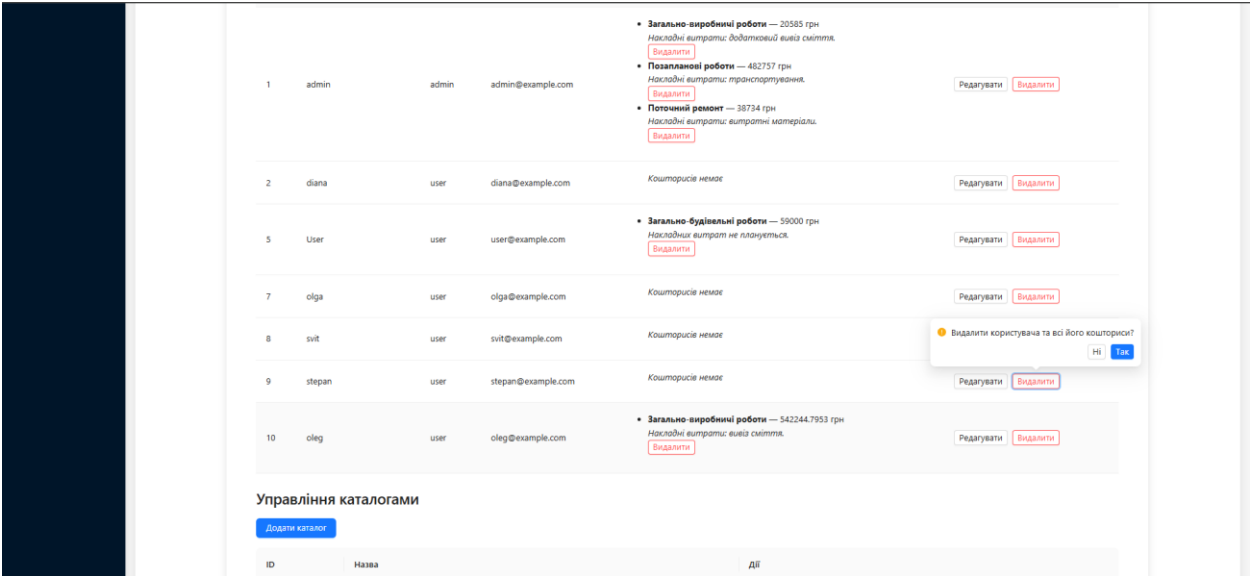


Рис. 3.26 Адмін-панель видалення користувача

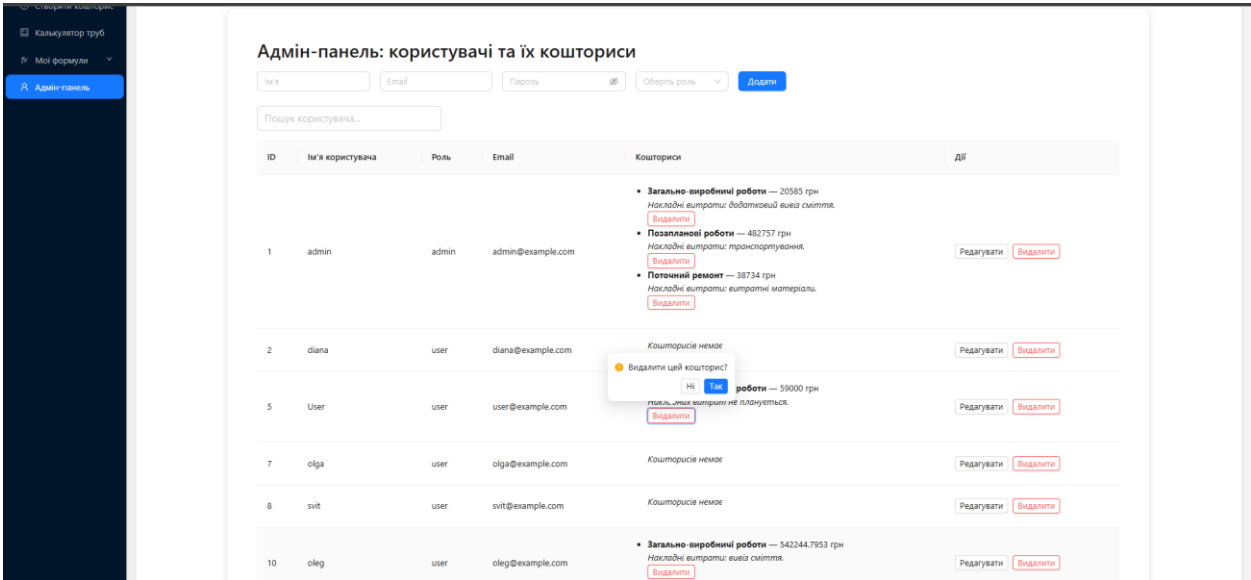


Рис. 3.27 Адмін-панель видалення кошторису в користувача

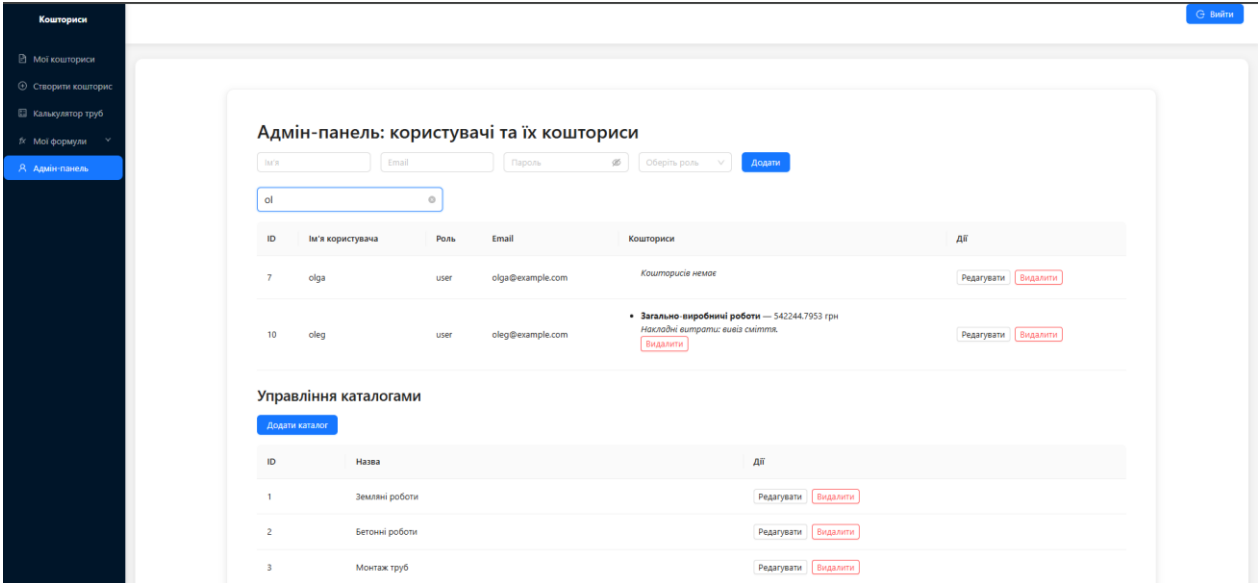


Рис. 3.28 Адмін-панель пошук користувача

Також реалізований пошук для користувачів (Рис. 3.28) та створення нового користувача адміністратором (Рис. 3.29).

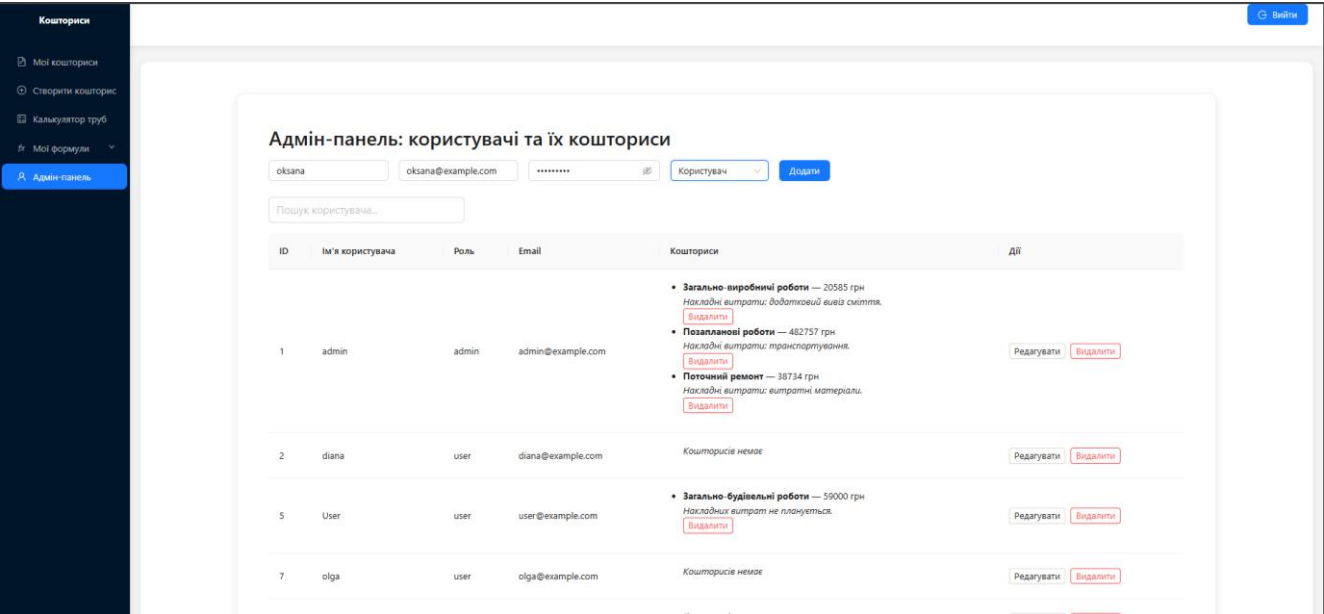


Рис. 3.29 Адмін-панель створення нового користувача

Також доступна повноцінна робота з каталогами: створення нових категорій (Рис. 3.30), перейменування, видалення або додавання нових позицій (Рис. 3.31) з вказаною одиницею виміру та ціною. Усі зміни вносяться через зручний візуальний інтерфейс, що значно полегшує адміністрування платформи.

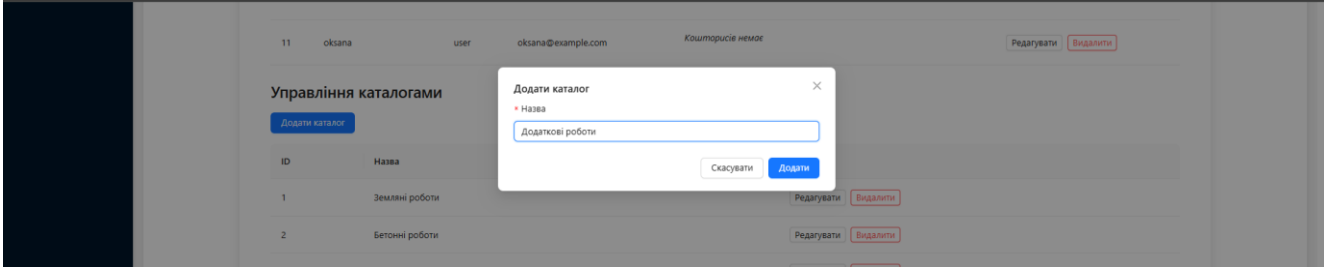


Рис. 3.30 Адмін-панель створення нового каталогу

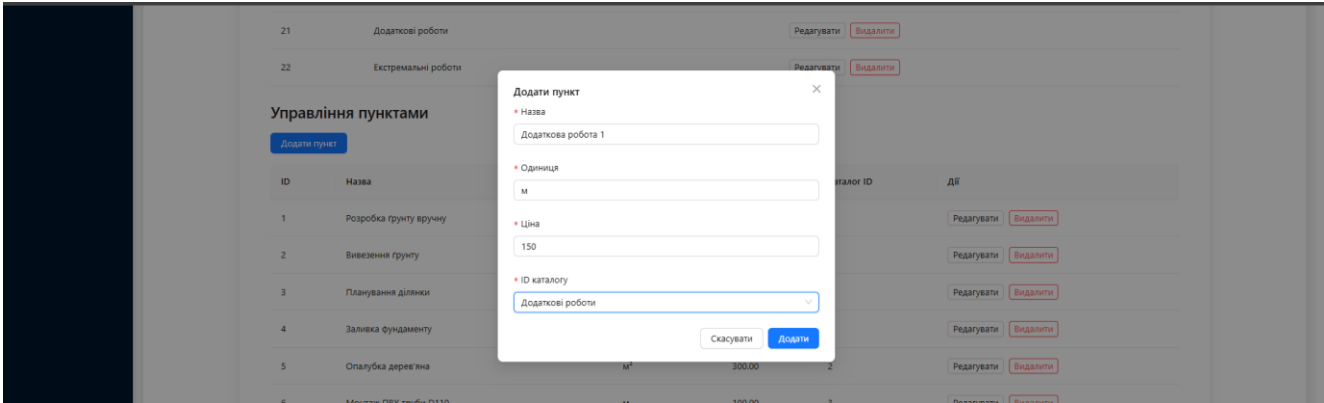


Рис. 3.30 Адмін-панель створення нового пункту в каталозі

Загальна візуальна структура додатку є мінімалістичною, але функціонально насиченою. Таким чином, інтерфейс вебдодатку повністю відповідає вимогам сучасних програмних систем: він інтуїтивний, багатофункціональний та дозволяє швидко й ефективно керувати усіма етапами створення й редагування кошторисної документації.

3.3 Тестування роботи програмного комплексу

Для перевірки коректності роботи REST API вебдодатку для складання кошторисів було використано середовище Postman. За його допомогою були протестовані основні маршрути системи: автентифікація користувачів, створення кошторисів, додавання позицій, збереження формул, обробка помилок доступу та валідації.

Тестування авторизації на першому етапі полягало у перевірці можливості входу зареєстрованих користувачів (Рис. 3.31). Запит до маршруту `/api/auth/login` надсилається з поштовою адресою та паролем. У разі успіху система повертає токен доступу, який надалі використовується для автентифікації при інших запитах.

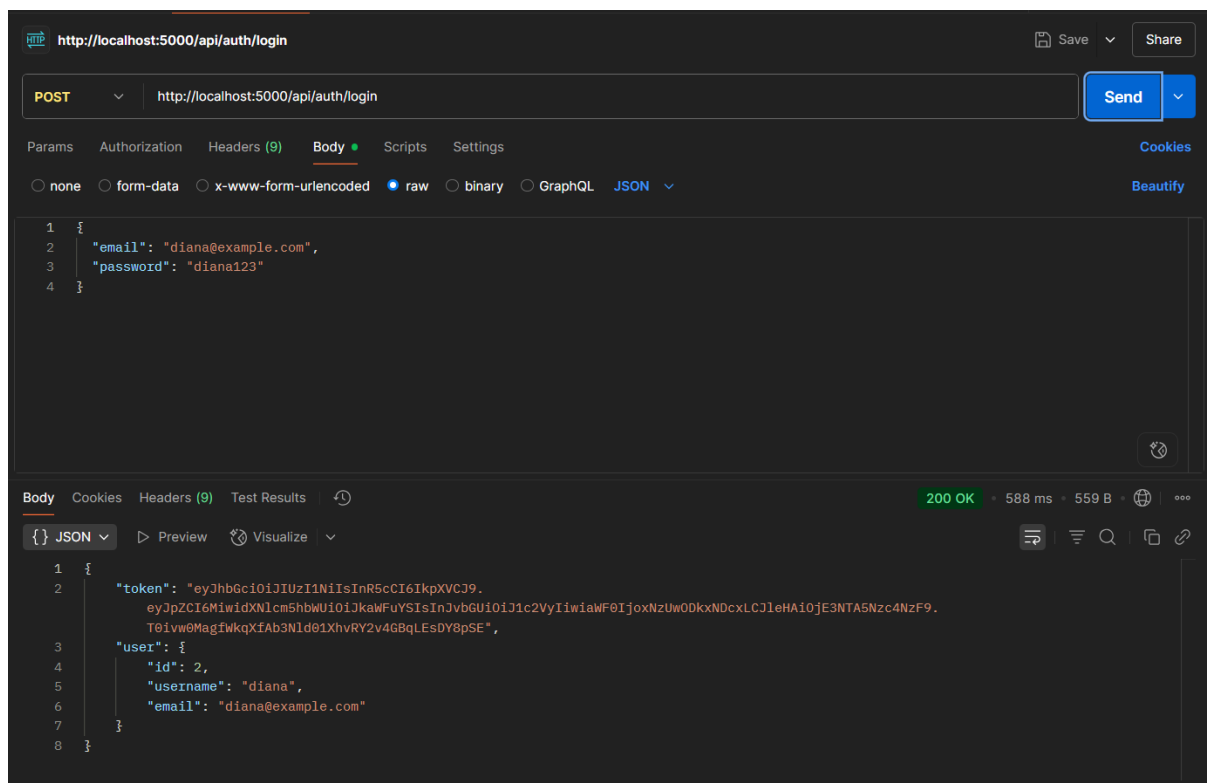


Рис. 3.31 Тестування авторизації користувача через Postman

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		76

Створення кошторису після авторизації включало перевірку можливості створення нового кошторису (Рис. 3.32). Для цього надсилався POST-запит до маршруту /api/estimates з JWT-токеном у заголовках. У тілі запиту містились такі дані, як назва, опис та відсоток накладних витрат. У відповіді система повертала унікальний ідентифікатор створеного кошторису.

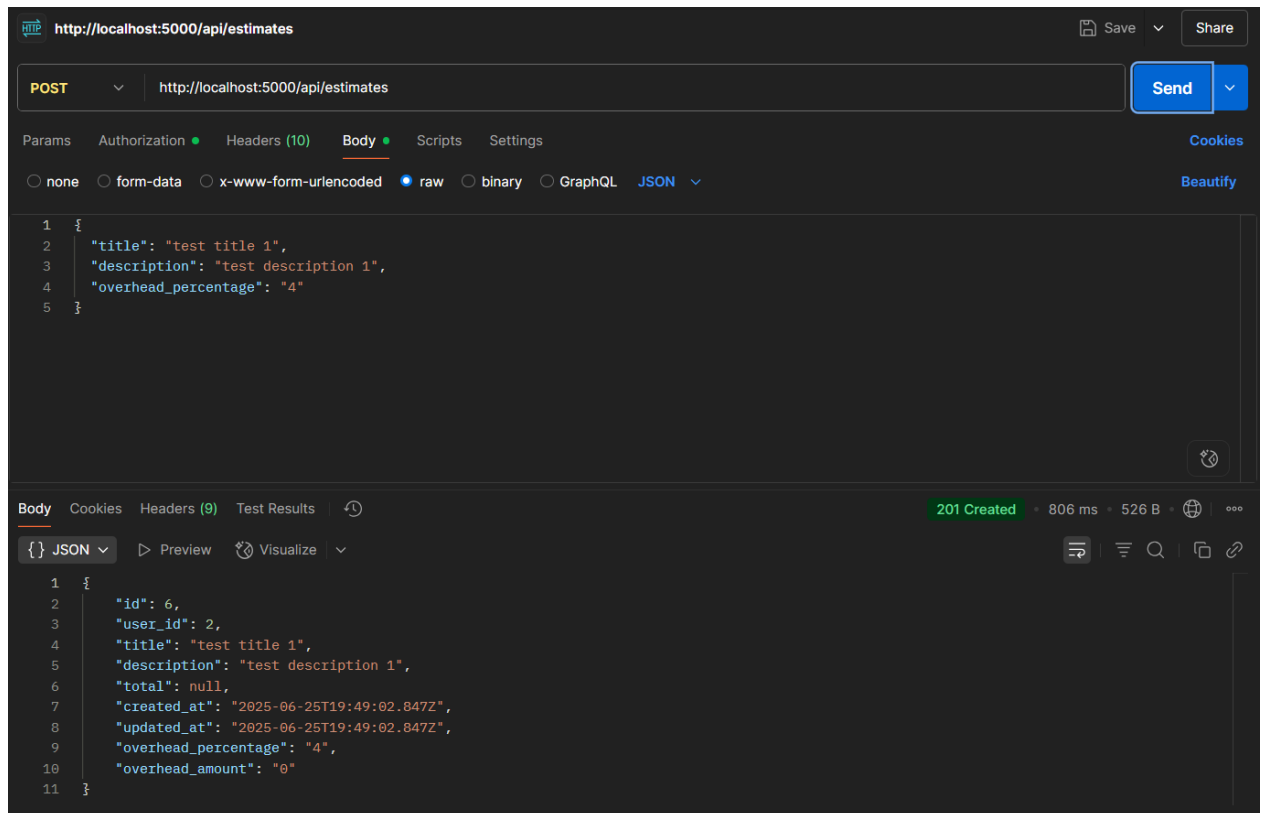


Рис. 3.32 Створення нового кошторису через Postman

Подальше тестування охоплювало функціональність додавання позицій до вже створеного кошторису. У запиті до маршруту /api/estimate-items передавались параметри estimateId, itemId, quantity та custom_price. У відповіді сервер повертав розраховану суму (total), що дозволяє динамічно формувати загальну вартість.

Також перевірялась безпека маршрутизаторів — система не дозволяє неавторизованому користувачу отримувати дані про чужі кошториси або змінювати формули інших користувачів. Для цього в Postman застосовувались окремі запити з фальшивими або простроченими JWT-токенами.

Крім тестування API, ручне тестування також проводилось у вебінтерфейсі. Було перевірено коректність роботи основних сторінок: створення кошторису, ре-

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		77

дагування, пошуку, експорту, калькулятора формул і панелі адміністратора. Усі виявлені на етапі тестування помилки було усунуто, що забезпечило стабільну роботу додатку під час розгортання.

Таким чином, було проведено комплексне ручне тестування основних модулів API вебдодатку за допомогою Postman. Отримані результати підтверджують коректність реалізації механізмів автентифікації, обробки даних, бізнес-логіки кошторисів, а також стабільну поведінку при виняткових ситуаціях.

Висновки до третього розділу

У третьому розділі було детально розглянуто етапи встановлення та налаштування системи, структуру її інтерфейсу та процес тестування ключових компонентів. Описано повний цикл розгортання вебдодатку на платформі Render, з урахуванням клієнтської частини (React), серверної частини (Node.js з Express) та підключення до бази даних PostgreSQL. Окрему увагу приділено взаємодії між компонентами системи, що було проілюстровано на діаграмі розгортання.

Проаналізовано архітектуру інтерфейсу вебдодатку, яка забезпечує інтуїтивну роботу користувача з такими функціями, як створення кошторису, додавання позицій із каталогу, розрахунок вартості та експорт у формати PDF і Excel. Адміністративна панель дає змогу ефективно керувати користувачами, каталогами та позиціями, забезпечуючи підтримку й контроль даних системи.

Завдяки використанню середовища Postman було проведено тестування ключових API-запитів, зокрема маршрутів авторизації, створення та редагування кошторисів, додавання позицій, збереження формул та обробки помилок. Це дозволило перевірити стійкість системи до помилкових даних, стабільність взаємодії з базою даних і правильність реалізації бізнес-логіки.

У результаті реалізований вебдодаток демонструє повноцінну інтеграцію клієнтської, серверної частини та бази даних, що дозволяє користувачам ефективно створювати, редагувати й аналізувати кошториси у зручному цифровому середовищі.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		78

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи бакалавра було спроектовано та реалізовано вебдодаток для створення, редагування, збереження та експорту кошторисної документації. Проведено глибокий аналіз предметної області, визначено актуальні проблеми у сфері формування кошторисів та обґрунтовано потребу в сучасному цифровому рішенні для автоматизації відповідних процесів.

На основі аналізу існуючих інструментів та практик було сформовано основні вимоги до функціональності системи, включаючи підтримку індивідуальних формул, калькулятор розрахунку, каталогізацію матеріалів, а також повноцінне управління кошторисами з можливістю експорту в PDF та Excel.

Розроблено архітектуру вебдодатку на основі модульного моноліту з використанням сучасного стеку технологій: React на фронтенді, Node.js з Express на бекенді, PostgreSQL як СУБД, Sequelize як ORM, та додаткових бібліотек для експорту (PDFKit, ExcelJS) і математичних обчислень (Math.js). Здійснено моделювання взаємодії системних компонентів за допомогою UML-діаграм: варіантів використання, класів, компонентів, активностей та розгортання, що дозволило чітко структурувати логіку роботи системи.

Розглянуто логіку роботи окремих функціональних модулів, таких як створення кошторису, редагування позицій, додавання формул користувача, а також функціонал адміністратора з управління контентом та правами доступу. Реалізовано зручний інтерфейс користувача з урахуванням ролей, що забезпечує доступ до функцій системи відповідно до прав користувача.

Окрему увагу приділено етапам тестування системи, перевірці стабільності роботи API за допомогою Postman, зокрема тестування основних CRUD-операцій, обробки помилок і перевірки авторизації.

У результаті виконаної роботи розроблено функціональний, масштабований і зручний у використанні вебдодаток, який відповідає вимогам до сучасного інструменту для формування кошторисів, що може використовуватись як у будівельній галузі, так і в інших сферах, де потрібне точне фінансове планування.

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		79

ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Онлайн-сервіс «Скласти кошторис» [Електронний ресурс] – Режим доступу: https://msmeta.com.ua/sklasty_koshtorys.php
2. Платформа «OneB Invoice» [Електронний ресурс] – Режим доступу: <https://invoice.oneb.app/mWk2EY/create>
3. JavaScript Tutorial [Електронний ресурс] – Режим доступу: <https://www.w3schools.com/js/>
4. Node.js [Електронний ресурс] – Режим доступу: <https://nodejs.org/en>
5. PostgreSQL [Електронний ресурс] – Режим доступу: <https://www.postgresql.org/>
6. ORM [Електронний ресурс] – Режим доступу: <https://it-rating.ua/optimizatsiya-roboti-z-bazami-danih-za-dopomogoyu-orm-object-relational-mapping>
7. Сучасний підручник з JavaScript [Електронний ресурс] – Режим доступу: <https://uk.javascript.info/>
8. JWT [Електронний ресурс] – Режим доступу: <https://jwt.io/>
9. ExcelJS [Електронний ресурс] – Режим доступу: <https://www.npmjs.com/package/exceljs>
10. PDFKit [Електронний ресурс] – Режим доступу: <https://pdfkit.org/>
11. Puppeteer [Електронний ресурс] – Режим доступу: <https://pptr.dev/>
12. Math.js [Електронний ресурс] – Режим доступу: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math
13. React [Електронний ресурс] – Режим доступу: <https://react.dev/>
14. React Router [Електронний ресурс] – Режим доступу: <https://reactrouter.com/>
15. Ant Design [Електронний ресурс] – Режим доступу: <https://ant.design/>
16. Axios [Електронний ресурс] – Режим доступу: <https://www.npmjs.com/package/axios>

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				80
Змн.	Арк.	№ докум.	Підпис	Дата		

17. Що таке API? [Електронний ресурс] – Режим доступу: <https://hire1.com.ua/glossary/api>
18. Postman [Електронний ресурс] – Режим доступу: <https://www.postman.com/>
19. Express.js [Електронний ресурс] – Режим доступу: <https://expressjs.com/>
20. Sequelize [Електронний ресурс] – Режим доступу: <https://sequelize.org/>
21. Монолітна архітектура ПЗ [Електронний ресурс] – Режим доступу: <https://qalight.ua/baza-znaniy/shho-take-monolitna-arhitektura/>
22. Що таке SQL і де його використовують? [Електронний ресурс] – Режим доступу: <https://goit.global.ua/articles/shcho-take-sql-i-de-yoho-vykorystovuiut/>
23. SQL підручник [Електронний ресурс] – Режим доступу: <https://w3schoolsua.github.io/sql/index.html#gsc.tab=0>
24. Send a request with the Postman API client [Електронний ресурс] – Режим доступу: <https://learning.postman.com/docs/sending-requests/create-requests/request-basics/>
25. Google Fonts API [Електронний ресурс] – Режим доступу: <https://fonts.google.com/>
26. StackOverflow [Електронний ресурс] – Режим доступу: <https://stackoverflow.com/questions>
27. Git [Електронний ресурс] – Режим доступу: <https://git-scm.com/>
28. Діаграми UML [Електронний ресурс] – Режим доступу: <https://evergreens.com.ua/ua/articles/uml-diagrams.html>
29. React icons [Електронний ресурс] – Режим доступу: <https://react-icons.github.io/react-icons/>
30. CSS підручник [Електронний ресурс] – Режим доступу: <https://w3schoolsua.github.io/css/index.html#gsc.tab=0>

ДОДАТКИ

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		82

Фрагмент коду файлу “AdminPanel.jsx”

```

import React, { useEffect, useState } from 'react';
import {
  Table,
  Input,
  Button,
  Typography,
  Spin,
  message,
  Popconfirm,
  Card,
  Form,
  Modal,
  Select,
  Space,
} from 'antd';
import API from '../services/api';

const { Title } = Typography;

const roleOptions = [
  { label: 'Користувач', value: 'user' },
  { label: 'Адмін', value: 'admin' },
];

// Мапа для перетворення цифр у нарядкові символи
const superscriptMap = {
  '0': '⁰',
  '1': '¹',
  '2': '²',
  '3': '³',
  '4': '⁴',
  '5': '⁵',
  '6': '⁶',
  '7': '⁷',
  '8': '⁸',
  '9': '⁹',
  '.': '·',
};

// Функція для конвертації числової частини в нарядкові символи
const toSuperscript = (numberStr) => {
  return numberStr
    .split('.')
    .map(char => superscriptMap[char] || char)
    .join('.');
};

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		83

```

// Універсальна функція для форматування одиниць виміру
const formatUnit = (unit) => {
  if (!unit) return unit;
  // Регулярний вираз для розпізнавання основи та числової частини
  const regex = /^(^[\d]+)(\d*\.\d*)$/i;
  const match = unit.match(regex);
  if (!match) return unit;
  const [, base, numberPart] = match;
  if (!numberPart) return base;
  return `${base}${toSuperscript(numberPart)}`;
};

const AdminPanel = () => {
  const [users, setUsers] = useState([]);
  const [loading, setLoading] = useState(true);
  const [searchTerm, setSearchTerm] = useState('');
  const [addingForm] = Form.useForm();
  const [editingUser, setEditingUser] = useState(null);
  const [editForm] = Form.useForm();

  // Стан для catalogs
  const [catalogs, setCatalogs] = useState([]);
  const [addingCatalog, setAddingCatalog] = useState(false);
  const [editingCatalog, setEditingCatalog] = useState(null);
  const [catalogForm] = Form.useForm();

  // Стан для items
  const [items, setItems] = useState([]);
  const [addingItem, setAddingItem] = useState(false);
  const [editingItem, setEditingItem] = useState(null);
  const [itemForm] = Form.useForm();

  useEffect(() => {
    fetchUsers();
    fetchCatalogs();
    fetchItems();
  }, []);

  const fetchUsers = async () => {
    setLoading(true);
    try {
      const response = await API.get('/api/users-estimates');
      setUsers(response.data);
    } catch (err) {
      message.error('Не вдалося завантажити користувачів');
    } finally {
      setLoading(false);
    }
  };

  const fetchCatalogs = async () => {

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		84

```

    try {
      const response = await API.get('/api/catalogs');
      setCatalogs(response.data);
    } catch (err) {
      message.error('Не вдалося завантажити каталоги');
    }
  };

const fetchItems = async () => {
  try {
    const response = await API.get('/api/items');
    const formattedItems = response.data.map(item => ({
      ...item,
      unit: formatUnit(item.unit),
    }));
    setItems(formattedItems);
  } catch (err) {
    message.error('Не вдалося завантажити пункти');
  }
};

const deleteEstimate = async (estimateId) => {
  try {
    await API.delete(`/api/estimates/${estimateId}`);
    message.success('Кошторис видалено');
    fetchUsers();
  } catch {
    message.error('Не вдалося видалити кошторис');
  }
};

const deleteUser = async (userId) => {
  try {
    await API.delete(`/api/users/${userId}`);
    message.success('Користувача видалено');
    fetchUsers();
  } catch {
    message.error('Не вдалося видалити користувача');
  }
};

const onAddUserFinish = async (values) => {
  try {
    await API.post('/api/users', values);
    message.success('Користувача додано');
    addingForm.resetFields();
    fetchUsers();
  } catch (err) {
    message.error(err.response?.data?.message || 'Помилка при додаванні користувача');
  }
}

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				85
Змн.	Арк.	№ докум.	Підпис	Дата		

```

};

const mapRoleToValue = (roleFromApi) => {
  if (roleFromApi === 'admin' || roleFromApi === 'user') return roleFromApi;
  if (roleFromApi === 1) return 'admin';
  if (roleFromApi === 0) return 'user';
  return 'user';
};

const handleEdit = (user) => {
  setEditingUser(user);
  editForm.setFieldsValue({
    username: user.username,
    email: user.email,
    role: mapRoleToValue(user.role),
  });
};

const handleEditSubmit = async () => {
  try {
    const values = await editForm.validateFields();
    await API.put(`/api/users/${editingUser.user_id}`, values);
    message.success('Користувача оновлено');
    setEditingUser(null);
    fetchUsers();
  } catch (err) {
    message.error('Помилка при редагуванні користувача');
  }
};

const handleAddCatalog = () => {
  setAddingCatalog(true);
};

const handleCatalogSubmit = async (values) => {
  try {
    await API.post('/api/catalogs', { name: values.name });
    message.success('Каталог додано');
    catalogForm.resetFields();
    setAddingCatalog(false);
    fetchCatalogs();
  } catch (err) {
    message.error('Помилка при додаванні каталогу');
  }
};

const handleEditCatalog = (catalog) => {
  setEditingCatalog(catalog);
  catalogForm.setFieldsValue({ name: catalog.name });
};

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		86

```

const handleUpdateCatalog = async () => {
  try {
    const values = await catalogForm.validateFields();
    await API.put(`/api/catalogs/${editingCatalog.id}`, values);
    message.success('Каталог оновлено');
    setEditingCatalog(null);
    fetchCatalogs();
  } catch (err) {
    message.error('Помилка при оновленні каталогу');
  }
};

const handleDeleteCatalog = async (id) => {
  try {
    await API.delete(`/api/catalogs/${id}`);
    message.success('Каталог видалено');
    fetchCatalogs();
    fetchItems();
  } catch (err) {
    message.error('Помилка при видаленні каталогу');
  }
};

const handleAddItem = () => {
  setAddingItem(true);
};

const handleItemSubmit = async (values) => {
  try {
    const formattedValues = {
      ...values,
      unit: formatUnit(values.unit),
    };
    await API.post('/api/items', formattedValues);
    message.success('Пункт додано');
    itemForm.resetFields();
    setAddingItem(false);
    fetchItems();
  } catch (err) {
    message.error('Помилка при додаванні пункту');
  }
};

const handleEditItem = (item) => {
  setEditingItem(item);
  itemForm.setFieldsValue({
    name: item.name,
    unit: item.unit,
    price: item.price,
    catalog_id: item.catalog_id,
  });
};

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				87
Змн.	Арк.	№ докум.	Підпис	Дата		

```

};

const handleUpdateItem = async () => {
  try {
    const values = await itemForm.validateFields();
    const formattedValues = {
      ...values,
      unit: formatUnit(values.unit),
    };
    await API.put(`/api/items/${editingItem.id}`, formattedValues);
    message.success('Пункт оновлено');
    setEditingItem(null);
    fetchItems();
  } catch (err) {
    message.error('Помилка при оновленні пункту');
  }
};

const handleDeleteItem = async (id) => {
  try {
    await API.delete(`/api/items/${id}`);
    message.success('Пункт видалено');
    fetchItems();
  } catch (err) {
    message.error('Помилка при видаленні пункту');
  }
};

const filteredUsers = users.filter((user) =>
  user.username.toLowerCase().includes(searchTerm.toLowerCase())
);

const userColumns = [
  { title: 'ID', dataIndex: 'user_id', key: 'user_id' },
  { title: 'Ім'я користувача', dataIndex: 'username', key: 'username' },
  { title: 'Роль', dataIndex: 'role', key: 'role' },
  { title: 'Email', dataIndex: 'email', key: 'email' },
  {
    title: 'Кошториси',
    key: 'estimates',
    render: (_, record) => (
      <ul style={{ paddingLeft: 20 }}>
        {record.estimates && record.estimates.length > 0 ? (
          record.estimates.map((est) => (
            <li key={est.id}>
              <strong>{est.title}</strong> – {est.total} грн <br />
              <em>{est.description}</em>
            <br />
            <Popconfirm
              title="Видалити цей кошторис?"
              onConfirm={() => deleteEstimate(est.id)}
            />
          )
        ) : null}
      </ul>
    )
  }
];

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		88

```

        okText="Так"
        cancelText="Hi"
      >
      <Button danger size="small">
        Видалити
      </Button>
    </Popconfirm>
  </li>
))
) : (
  <em>Кошторисів немає</em>
)}
</ul>
),
},
{
  title: 'Дії',
  key: 'actions',
  render: (_, record) => (
    <Space>
      <Button size="small" onClick={() => handleEdit(record)}>
        Редагувати
      </Button>
      <Popconfirm
        title="Видалити користувача та всі його кошториси?"
        onConfirm={() => deleteUser(record.user_id)}
        okText="Так"
        cancelText="Hi"
      >
        <Button danger size="small">
          Видалити
        </Button>
      </Popconfirm>
    </Space>
  ),
},
];

const catalogColumns = [
  { title: 'ID', dataIndex: 'id', key: 'id' },
  { title: 'Назва', dataIndex: 'name', key: 'name' },
  {
    title: 'Дії',
    key: 'actions',
    render: (_, record) => (
      <Space>
        <Button size="small" onClick={() => handleEditCatalog(record)}>
          Редагувати
        </Button>
        <Popconfirm
          title="Видалити каталог?"

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				89
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        onConfirm={() => handleDeleteCatalog(record.id)}
        okText="Так"
        cancelText="Ні"
      >
      <Button danger size="small">
        Видалити
      </Button>
    </Popconfirm>
  </Space>
),
},
];

const itemColumns = [
  { title: 'ID', dataIndex: 'id', key: 'id' },
  { title: 'Назва', dataIndex: 'name', key: 'name' },
  { title: 'Одиниця', dataIndex: 'unit', key: 'unit' },
  { title: 'Ціна', dataIndex: 'price', key: 'price' },
  { title: 'Каталог ID', dataIndex: 'catalog_id', key: 'catalog_id' },
  {
    title: 'Дії',
    key: 'actions',
    render: (_, record) => (
      <Space>
        <Button size="small" onClick={() => handleEditItem(record)}>
          Редагувати
        </Button>
        <Popconfirm
          title="Видалити пункт?"
          onConfirm={() => handleDeleteItem(record.id)}
          okText="Так"
          cancelText="Ні"
        >
          <Button danger size="small">
            Видалити
          </Button>
        </Popconfirm>
      </Space>
    ),
  },
];

return (
  <Card
    style={{
      padding: 24,
      borderRadius: 12,
      boxShadow: '0 4px 12px rgba(0, 0, 0, 0.1)',
      maxWidth: 1500,
      margin: '24px auto',
      backgroundColor: '#fff',

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				90
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }}
  >
  <Title
    level={2}
    style={{
      marginBottom: 16,
      whiteSpace: 'normal',
      wordBreak: 'break-word',
      maxWidth: '100%',
    }}
  >
    Адмін-панель: користувачі та їх кошториси
  </Title>

  <Form
    form={addingForm}
    layout="inline"
    onFinish={onAddUserFinish}
    style={{ marginBottom: 24 }}
  >
    <Form.Item
      name="username"
      rules={[{ required: true, message: 'Введіть ім'я' }]}
    >
      <Input placeholder="Ім'я" />
    </Form.Item>
    <Form.Item
      name="email"
      rules={[{ required: true, message: 'Введіть email' }]}
    >
      <Input placeholder="Email" />
    </Form.Item>
    <Form.Item
      name="password"
      rules={[{ required: true, message: 'Введіть пароль' }]}
    >
      <Input.Password placeholder="Пароль" />
    </Form.Item>
    <Form.Item
      name="role"
      rules={[{ required: true, message: 'Оберіть роль' }]}
    >
      <Select
        placeholder="Оберіть роль"
        style={{ width: 150 }}
        options={roleOptions}
      />
    </Form.Item>
    <Form.Item>
      <Button type="primary" htmlType="submit">
        Додати
      </Button>
    </Form.Item>
  </Form>

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				91
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        </Button>
      </Form.Item>
    </Form>

    <Input
      placeholder="Пошук користувача..."
      value={searchTerm}
      onChange={(e) => setSearchTerm(e.target.value)}
      size="large"
      style={{ width: 300, marginBottom: 16, borderRadius: 6 }}
      allowClear
    />

    {loading ? (
      <Spin style={{ display: 'block', margin: '0 auto' }} />
    ) : (
      <Table
        columns={userColumns}
        dataSource={filteredUsers}
        rowKey="user_id"
        pagination={false}
        style={{ marginBottom: 24 }}
      />
    )}

    <Title level={3} style={{ marginTop: 24, marginBottom: 16 }}>
      Управління каталогами
    </Title>
    <Button
      type="primary"
      onClick={handleAddCatalog}
      style={{ marginBottom: 16 }}
    >
      Додати каталог
    </Button>
    <Table
      columns={catalogColumns}
      dataSource={catalogs}
      rowKey="id"
      pagination={false}
      style={{ marginBottom: 24 }}
    />
    <Modal
      visible={addingCatalog}
      title="Додати каталог"
      onCancel={() => setAddingCatalog(false)}
      onOk={() => catalogForm.submit()}
      okText="Додати"
      cancelText="Скасувати"
    >

```

```

        <Form form={catalogForm} onFinish={handleCatalogSubmit}
layout="vertical">
            <Form.Item
                name="name"
                label="Назва"
                rules={[{ required: true, message: 'Введіть назву' }]}
            >
                <Input />
            </Form.Item>
        </Form>
    </Modal>
    <Modal
        visible={!editingCatalog}
        title="Редагувати каталог"
        onCancel={() => setEditingCatalog(null)}
        onOk={handleUpdateCatalog}
        okText="Зберегти"
        cancelText="Скасувати"
    >
        <Form form={catalogForm} layout="vertical">
            <Form.Item
                name="name"
                label="Назва"
                rules={[{ required: true, message: 'Введіть назву' }]}
            >
                <Input />
            </Form.Item>
        </Form>
    </Modal>

    <Title level={3} style={{ marginTop: 24, marginBottom: 16 }}>
        Управління пунктами
    </Title>
    <Button
        type="primary"
        onClick={handleAddItem}
        style={{ marginBottom: 16 }}
    >
        Додати пункт
    </Button>
    <Table
        columns={itemColumns}
        dataSource={items}
        rowKey="id"
        pagination={false}
    />
    <Modal
        visible={addingItem}
        title="Додати пункт"
        onCancel={() => setAddingItem(false)}
        onOk={() => itemForm.submit()}
    >

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				93
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        okText="Додати"
        cancelText="Скасувати"
    >
    <Form form={itemForm} onFinish={handleItemSubmit} layout="vertical">
        <Form.Item
            name="name"
            label="Назва"
            rules={[{ required: true, message: 'Введіть назву' }]}
        >
            <Input />
        </Form.Item>
        <Form.Item
            name="unit"
            label="Одиниця"
            rules={[
                { required: true, message: 'Введіть одиницю' },
                {
                    pattern: /^[^\d\s]+(\d*\.\d*)?$/i,
                    message: 'Введіть коректну одиницю (наприклад, м2,
л3.5)',
                },
            ]}
        >
            <Input placeholder="Наприклад, м2, л3.5, см²" />
        </Form.Item>
        <Form.Item
            name="price"
            label="Ціна"
            rules={[{ required: true, message: 'Введіть ціну' }]}
        >
            <Input type="number" step="0.01" />
        </Form.Item>
        <Form.Item
            name="catalog_id"
            label="ID каталогу"
            rules={[{ required: true, message: 'Виберіть каталог' }]}
        >
            <Select>
                {catalogs.map((cat) => (
                    <Select.Option key={cat.id} value={cat.id}>
                        {cat.name}
                    </Select.Option>
                ))}
            </Select>
        </Form.Item>
    </Form>
</Modal>
<Modal
    visible={!editingItem}
    title="Редагувати пункт"
    onCancel={() => setEditingItem(null)}

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				94
Змн.	Арк.	№ докум.	Підпис	Дата		

```

onOk={handleUpdateItem}
okText="Зберегти"
cancelText="Скасувати"
>
<Form form={itemForm} layout="vertical">
  <Form.Item
    name="name"
    label="Назва"
    rules={[{ required: true, message: 'Введіть назву' }]}
  >
    <Input />
  </Form.Item>
  <Form.Item
    name="unit"
    label="Одиниця"
    rules={[
      { required: true, message: 'Введіть одиницю' },
      {
        pattern: /^[^\d\s]+(\d*\.\d*)?$/i,
        message: 'Введіть коректну одиницю (наприклад, м2,
л3.5)',
      },
    ]}
  >
    <Input placeholder="Наприклад, м2, л3.5, см²" />
  </Form.Item>
  <Form.Item
    name="price"
    label="Ціна"
    rules={[{ required: true, message: 'Введіть ціну' }]}
  >
    <Input type="number" step="0.01" />
  </Form.Item>
  <Form.Item
    name="catalog_id"
    label="ID каталогу"
    rules={[{ required: true, message: 'Виберіть каталог' }]}
  >
    <Select>
      {catalogs.map((cat) => (
        <Select.Option key={cat.id} value={cat.id}>
          {cat.name}
        </Select.Option>
      ))}
    </Select>
  </Form.Item>
</Form>
</Modal>

<Modal
  visible={!editingUser}

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				95
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        title="Редагування користувача"
        onCancel={() => setEditingUser(null)}
        onOk={handleEditSubmit}
        okText="Зберегти"
        cancelText="Скасувати"
        destroyOnClose
    >
    <Form form={editForm} layout="vertical">
        <Form.Item
            name="username"
            label="Ім'я"
            rules={[{ required: true, message: 'Введіть ім'я' }]}
        >
            <Input />
        </Form.Item>
        <Form.Item
            name="email"
            label="Email"
            rules={[{ required: true, message: 'Введіть email' }]}
        >
            <Input />
        </Form.Item>
        <Form.Item
            name="role"
            label="Роль"
            rules={[{ required: true, message: 'Оберіть роль' }]}
        >
            <Select options={roleOptions} />
        </Form.Item>
    </Form>
</Modal>
</Card>
    );
};

export default AdminPanel;

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		96

Фрагмент коду файлу “server.js”

```

const express = require('express');
const cors = require('cors');
const dotenv = require('dotenv');

const app = express();
dotenv.config();

// app.use(cors());
app.use(cors({
  origin: 'http://localhost:3000',
  allowedHeaders: ['Content-Type', 'Authorization'],
}));
app.use(express.json());

const authRoutes = require('./routes/authRoutes');
app.use('/api/auth', authRoutes);

const estimatesRoutes = require('./routes/estimatesRoutes');
app.use('/api/estimates', estimatesRoutes);

// Тестовий маршрут
app.get('/', (req, res) => {
  res.send('Backend працює!');
});

const adminRoutes = require('./routes/adminRoutes');
app.use('/api', adminRoutes);

const catalogRoutes = require('./routes/catalogRoutes');
app.use('/api/catalogs', catalogRoutes);

const estimateItemRoutes = require('./routes/estimateItemRoutes');
app.use('/api/estimates', estimateItemRoutes);

const exportPdfRoutes = require('./routes/exportPdfRoutes');
app.use('/api', exportPdfRoutes);

const exportExcelRoutes = require('./routes/exportExcelRoutes');
app.use('/api', exportExcelRoutes);

const formulaRoutes = require('./routes/formulaRoutes');
app.use('/api', formulaRoutes);

const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				97
Змн.	Арк.	№ докум.	Підпис	Дата		

Фрагмент коду файлу “exportPdfRoutes.js”

```

const PDFDocument = require('pdfkit');
const express = require('express');
const path = require('path');
const fs = require('fs');
const router = express.Router();
const { getEstimateWithItems } = require('../services/estimateService');

router.get('/export-pdf/:estimateId', async (req, res) => {
  try {
    const estimate = await getEstimateWithItems(req.params.estimateId);
    if (!estimate) return res.status(404).send('Кошторис не знайдено');

    const doc = new PDFDocument({ margin: 50 });
    doc.registerFont('Roboto', path.join(__dirname, '../fonts/Roboto-VariableFont_wdth,wght.ttf'));
    doc.font('Roboto');

    res.setHeader('Content-Disposition', `attachment; filename=estimate-${estimate.id}.pdf`);
    res.setHeader('Content-Type', 'application/pdf');
    res.setHeader('Cache-Control', 'no-store');

    doc.pipe(res);

    // Логотип
    const logoPath = path.join(__dirname, '../public/logo.jpg');
    if (fs.existsSync(logoPath)) {
      doc.image(logoPath, 50, 45, { width: 50, height: 50 });
    }

    // Заголовок
    doc.fontSize(20).text(`Кошторис №${estimate.id}`, 0, 50, { align: 'center'
  });

  // Дата (форматування)
  const createdAt = new Date(estimate.created_at);
  const formattedDate = createdAt.toLocaleDateString('uk-UA', {
    year: 'numeric', month: 'long', day: 'numeric'
  });
  doc.moveDown();
  doc.fontSize(12).text(`Дата створення: ${formattedDate}`, { align: 'center'
});

doc.moveDown(2);

// Назва і опис
doc.fontSize(14).text(`Назва: ${estimate.title}`, 50);

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				98
Змн.	Арк.	№ докум.	Підпис	Дата		

```

doc.moveDown(0.5);
doc.fontSize(12).text(`Опис: ${estimate.description} || '-'`, 50);

doc.moveDown();

// Таблиця позицій – колонки: № | Назва | Кількість | Ціна | Сума
const tableTop = doc.y + 10;
const itemNumberX = 50;
const itemNameX = 80;
const itemQtyX = 320;
const itemPriceX = 390;
const itemTotalX = 460;
const rowHeight = 20;

// Заголовок таблиці
doc.fontSize(12).text('№', itemNumberX, tableTop, { bold: true });
doc.text('Назва', itemNameX, tableTop);
doc.text('Кількість', itemQtyX, tableTop);
doc.text('Ціна', itemPriceX, tableTop);
doc.text('Сума', itemTotalX, tableTop);

// Лінія під заголовком
doc.moveTo(itemNumberX, tableTop + 15)
  .lineTo(itemTotalX + 70, tableTop + 15)
  .stroke();

// Позиції
let y = tableTop + 20;

if (!estimate.items || estimate.items.length === 0) {
  doc.text('Немає позицій', itemNumberX, y);
  y += rowHeight;
} else {
  estimate.items.forEach((item, i) => {
    const rawPrice = item.custom_price ?? item.price ?? 0;
    const price = Number(rawPrice);
    const quantity = Number(item.quantity) || 0;
    const total = quantity * price;
    const name = item.name || 'Без назви';

    doc.text((i + 1).toString(), itemNumberX, y);
    doc.text(name, itemNameX, y, { width: 220 });
    doc.text(quantity.toString(), itemQtyX, y, { width: 50, align:
'right' });
    doc.text(price.toFixed(2), itemPriceX, y, { width: 60, align: 'right'
});
    doc.text(total.toFixed(2), itemTotalX, y, { width: 60, align: 'right'
});

    y += rowHeight;
  });
}

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				99
Змн.	Арк.	№ докум.	Підпис	Дата		

```

}

// Лінія внизу таблиці
doc.moveTo(itemNumberX, y - 5)
  .lineTo(itemTotalX + 70, y - 5)
  .stroke();

// Накладні витрати та загальна сума
const total = Number(estimate.total) || 0;
const overheadPercentage = Number(estimate.overhead_percentage) || 0;
const overheadAmount = (total * overheadPercentage) / 100;

y += 10;
doc.fontSize(12).text(`Накладні витрати (${overheadPercentage}%):
${overheadAmount.toFixed(2)} грн`, 50, y, { align: 'right' }, 30);
y += 30;
doc.fontSize(14).text(`Загальна сума: ${total.toFixed(2)} грн`, 50, y, {
align: 'right'}, 30);

// Підписи внизу сторінки
y += 70;
const signatureX1 = 50;
const signatureX2 = 350;
const signatureY = y;

doc.moveTo(signatureX1, signatureY)
  .lineTo(signatureX1 + 200, signatureY)
  .stroke();
doc.text('Склав:', signatureX1, signatureY + 5);

doc.moveTo(signatureX2, signatureY)
  .lineTo(signatureX2 + 200, signatureY)
  .stroke();
doc.text('Затвердив:', signatureX2, signatureY + 5);

// Рамка по краях сторінки
const pageWidth = doc.page.width;
const pageHeight = doc.page.height;
const margin = 50;

doc.lineWidth(1);
doc.rect(margin / 2, margin / 2, pageWidth - margin, pageHeight -
margin).stroke();

doc.end();

doc.on('finish', () => {
  console.log(`PDF для кошторису ${estimate.id} успішно згенеровано.`);
});

} catch (error) {

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		100

```

        console.error('PDF помилка:', error.stack || error);
        res.status(500).send('Помилка генерації PDF');
    }
});

module.exports = router;

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		101

Фрагмент коду файлу “exportExcelRoutes.js”

```

const express = require('express');
const router = express.Router();
const ExcelJS = require('exceljs');
const { getEstimateWithItems } = require('../services/estimateService');

router.get('/export-excel/:estimateId', async (req, res) => {
  try {
    const estimate = await getEstimateWithItems(req.params.estimateId);
    if (!estimate) return res.status(404).send('Кошторис не знайдено');

    const workbook = new ExcelJS.Workbook();
    const sheet = workbook.addWorksheet(`Estimate ${estimate.id}`);

    // Заголовок
    sheet.mergeCells('A1', 'E1');
    sheet.getCell('A1').value = `Кошторис №${estimate.id}`;
    sheet.getCell('A1').font = { bold: true, size: 16 };
    sheet.getCell('A1').alignment = { vertical: 'middle', horizontal: 'center' };

    sheet.addRow([]);

    // Основна інформація
    sheet.addRow(['Назва', estimate.title || '']);
    sheet.addRow(['Опис', estimate.description || '']);
    sheet.addRow(['Дата створення', new
Date(estimate.created_at).toLocaleDateString('uk-UA')]);
    sheet.addRow([]);

    // Заголовки таблиці
    const headerRow = sheet.addRow(['№', 'Назва', 'Кількість', 'Ціна', 'Сума']);
    headerRow.eachCell((cell) => {
      cell.font = { bold: true };
      cell.alignment = { vertical: 'middle', horizontal: 'center' };
    });

    // Таблиця з позиціями
    estimate.items.forEach((item, index) => {
      const price = item.custom_price ?? item.price ?? 0;
      const quantity = item.quantity || 0;
      const total = quantity * price;

      sheet.addRow([
        index + 1,
        item.name || 'Без назви',
        quantity,
        price,
        total

```

		Горбачевська Д.А.			ДУ «Житомирська політехніка».25.121.8.000 – ПЗ	Арк.
		Годлевський Ю.О.				102
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    });
  });

  sheet.addRow([]);

  // Накладні витрати та загальна сума
  const total = Number(estimate.total) || 0;
  const overheadPercentage = Number(estimate.overhead_percentage) || 0;
  const overheadAmount = (total * overheadPercentage) / 100;

  sheet.addRow(['Накладні витрати', `${overheadPercentage}%`, '', '', overheadAmount]);
  sheet.addRow(['Загальна сума', '', '', '', total]);

  // Стили останніх рядків
  const lastRow1 = sheet.lastRow;
  lastRow1.eachCell(cell => cell.font = { bold: true });

  const lastRow2 = sheet.getRow(sheet.lastRow.number - 1);
  lastRow2.eachCell(cell => cell.font = { bold: true });

  // 📏 Автоматична ширина колонок
  sheet.columns.forEach((column) => {
    let maxLength = 10;
    column.eachCell({ includeEmpty: true }, (cell) => {
      const value = cell.value ? cell.value.toString() : '';
      maxLength = Math.max(maxLength, value.length + 2);
    });
    column.width = maxLength;
  });

  // Відправлення файлу
  res.setHeader('Content-Type', 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
  res.setHeader('Content-Disposition', `attachment; filename=estimate-${estimate.id}.xlsx`);

  await workbook.xlsx.write(res);
  res.end();
} catch (error) {
  console.error('Excel помилка:', error.stack || error);
  res.status(500).send('Помилка генерації Excel');
}
});

module.exports = router;

```