НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

“До захисту допущено”
Завідувач кафедри
_____ Наталія АУШЕВА
“___” _____2026 р.

Дипломна робота
на здобуття ступеня бакалавра

За освітньою програмою “Цифрові технології в енергетиці”

Спеціальності 122 “Комп’ютерні науки”

на тему: “Вебсистема аналізу та управління ризиками на основі кількісних методів”

Виконала: студентка 4 курсу, групи ТР-22

Канавець Вікторія Олександрівна _____
(прізвище, ім’я, по батькові) (підпис)

Керівник: асистент каф. цифрових технологій в енергетиці,

Рудик Володимир Іванович _____
(посада, науковий ступінь, вчене звання, прізвище, ім’я, по батькові) (підпис)

Рецензент: к.т.н., доц. каф. цифрових технологій в енергетиці,

Пешко Віталій Анатолійович _____
(посада, науковий ступінь, вчене звання, прізвище, ім’я, по батькові) (підпис)

Н.контроль доцент каф. цифрових технологій в енергетиці,
д.ф., Гурін Артем Леонідович _____
(посада, науковий ступінь, вчене звання, прізвище, ім’я, по батькові) (підпис)

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ — 2026

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ
Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ
Рівень вищої освіти — перший (бакалаврський)
Спеціальність 122 “Комп’ютерні науки”
Освітньо-професійна програма “Цифрові технології в енергетиці”

ЗАТВЕРДЖУЮ
Завідувач кафедри ЦТЕ
_____ Наталія АУШЕВА
“__”_____2026 р.

З А В Д А Н Н Я
на дипломну роботу студенту

Канавець Вікторії Олександрівні
(прізвище, ім’я, по батькові)

1. Тема роботи “Вебсистема аналізу та управління ризиками на основі кількісних методів ”

керівник роботи Рудик Володимир Іванович
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від “28” травня 2026 р. №1992-с

2. Термін подання роботи студентом 04.06.26 р.

3. Вихідні дані до роботи мова програмування TypeScript, фреймворки NestJS та Next.js (React), реляційна СУБД PostgreSQL з ORM Prisma, додаткові засоби: Tailwind CSS, Recharts (візуалізація), PDFKit (звітування), середовище Docker і система контролю версій Git.

4. Перелік питань, які потрібно розробити 1) аналіз предметної області та програмних аналогів: дослідження методів управління проєктними ризиками та порівняння існуючих рішень; 2) визначення вимог і методів оцінювання ризиків: формування вимог до системи, обґрунтування методів кількісного аналізу та вибору технологічного стеку розробки; 3) проєктування архітектури та структури бази даних: розробка загальної клієнт-серверної моделі та логічної структури БД для збереження інформації; 4) програмна реалізація бекенду та вебінтерфейсу: розробка серверної логіки для виконання розрахунків і клієнтської частини для зручної взаємодії з користувачем.

5. Орієнтовний перелік ілюстративного матеріалу діаграми (Use-case, класів), схема архітектури; скріншоти інтерфейсу: автентифікація, реєстр ризиків, матриця ризиків, FMEA, Монте-Карло, Торнадо, Парето, Спірмен, PDF-звіт.

6. Дата видачі завдання 13.09.25 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів дипломної роботи	Примітка
1	Затвердження теми роботи	13.09.2025	Виконано
2	Аналіз предметної області, дослідження існуючих рішень для оцінки ризиків та визначення вимог	14-20.04.26	Виконано
3	Дослідження та обґрунтування кількісних і якісних методів оцінки ризиків (FMEA, Монте-Карло тощо)	21-23.04.26	Виконано
4	Розробка архітектури системи, проєктування БД та вибір технологій реалізації	24-25.04.26	Виконано
5	Розробка серверної частини системи та реалізація обчислювального ядра	26-27.04.26	Виконано
6	Розробка клієнтської частини, інтеграція модулів візуалізації та генерації звітів	28.04-04.05.26	Виконано
7	Оформлення пояснювальної записки	05-13.05.26	Виконано
8	Захист програмного продукту	14.05.26	Виконано
9	Передзахист	04.06.26	Виконано
10	Захист	15-20.06.26	Виконано

Студент _____ Вікторія КАНАВЕЦЬ
(підпис) (ім'я, ПРІЗВИЩЕ)

Керівник роботи _____ Володимир РУДИК
(підпис) (ім'я, ПРІЗВИЩЕ)

АНОТАЦІЯ

Дипломну роботу викладено на 75 сторінках тексту, містить 20 ілюстрацій, 8 таблиць, 2 додатки та список використаних джерел із 20 найменувань.

Метою роботи є розробка інтегрованої вебсистеми для аналізу та управління ризиками, що передбачає впровадження щонайменше п’яти кількісних методів оцінки, забезпечення їх пріоритизації та автоматизацію звітності.

Основний зміст роботи охоплює аналіз предметної області та існуючих рішень у сфері оцінки ризиків, обґрунтування застосованих математичних моделей, а також проєктування масштабованої клієнт-серверної архітектури. Практична частина присвячена формуванню реєстру ризиків, програмній реалізації комплексу методів оцінки, розробці механізму валідації результатів із використанням рангової кореляції Спірмена, впровадженню системи аудит-логу та автоматизації створення аналітичних звітів у форматі PDF.

Методи та інструментарій: об’єктно-орієнтоване програмування, мова TypeScript, фреймворки NestJS та Next.js, реляційна СУБД PostgreSQL, Prisma ORM, контейнеризація за допомогою Docker. У процесі моделювання застосовано методи математичного та статистичного аналізу: матрицю ризиків, FMEA, метод Монте-Карло, аналіз Парето, діаграму Торнадо та Cost-Benefit аналіз.

Результатом роботи є програмний застосунок для автоматизації процесів виявлення, кількісного вимірювання та стратегічного впорядкування ризиків у межах інвестиційної, виробничої та проєктної діяльності.

Розроблений програмний продукт може бути адаптований для управління проєктами в галузях ІТ, будівництва, фінансів, виробництва та енергетики з метою прийняття обґрунтованих управлінських рішень та оптимізації часових витрат на проведення комплексного аналізу.

Ключові слова: ВЕБСИСТЕМА, УПРАВЛІННЯ РИЗИКАМИ, КІЛЬКІСНИЙ АНАЛІЗ, МЕТОД МОНТЕ-КАРЛО, COST-BENEFIT АНАЛІЗ, NESTJS, NEXT.JS.

ABSTRACT

This diploma project comprises 75 pages of text, featuring 20 illustrations, 8 tables, 2 appendices, and a reference list of 20 items.

Objective: The primary goal of this research is the development of an integrated web-based system for risk analysis and management. The project incorporates at least five quantitative assessment methodologies, provides prioritized risk evaluation, and automates the reporting process.

Content: The study encompasses an analysis of the subject domain and existing risk assessment solutions, the justification of selected mathematical models, and the design of a scalable client-server architecture. The practical implementation focuses on establishing a risk register, developing software modules for assessment techniques, validating results using Spearman's rank correlation, implementing an audit log system, and automating the generation of analytical reports in PDF format.

Methods and Tools: The development utilized object-oriented programming, TypeScript, NestJS, Next.js, PostgreSQL, Prisma ORM, and Docker for containerization. The mathematical and statistical modeling employed includes the risk matrix, FMEA, the Monte Carlo method, Pareto analysis, Tornado diagrams, and Cost-Benefit analysis.

Results: The project concludes with the delivery of a software application designed to automate the identification, quantitative measurement, and strategic prioritization of risks across investment, production, and project management activities.

Practical Application: The developed software is adaptable for project management across the IT, construction, finance, manufacturing, and energy sectors, facilitating informed decision-making and optimizing the time required for comprehensive risk assessment.

Keywords: WEB SYSTEM, RISK MANAGEMENT, QUANTITATIVE ANALYSIS, MONTE CARLO METHOD, COST-BENEFIT ANALYSIS, NESTJS.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

АСУ ТП (Автоматизована система управління технологічним процесом) — комплекс програмних та технічних засобів, призначених для автоматизації управління технологічними об’єктами.

БД (База даних) — структурований набір даних, організований для ефективного зберігання, пошуку та обробки інформації.

СУБД (Система управління базами даних) — програмне забезпечення для створення, адміністрування та експлуатації баз даних.

API (Application Programming Interface) — набір протоколів та інструментів, що забезпечують взаємодію між різними програмними системами.

CBA (Cost-Benefit Analysis) — метод економічного аналізу, що застосовується для оцінки доцільності прийняття управлінських чи інвестиційних рішень шляхом порівняння витрат і вигод.

CRUD (Create, Read, Update, Delete) — сукупність чотирьох базових операцій, що складають основу маніпулювання даними в інформаційних системах.

DTO (Data Transfer Object) — об’єкт, спеціалізований для безпечного транспортування даних між підсистемами програмного забезпечення.

FMEA (Failure Mode and Effects Analysis) — методологія ідентифікації, аналізу та запобігання потенційним відмовам у системі або процесі.

JWT (JSON Web Token) — стандарт безпечного передавання даних (токенів) між сторонами, що використовується для автентифікації та авторизації користувачів.

ORM (Object-Relational Mapping) — технологія, що забезпечує відображення об’єктно-орієнтованих структур коду на реляційні моделі баз даних.

REST (Representational State Transfer) — архітектурний стиль, що визначає

принципи взаємодії компонентів у розподілених системах на основі протоколу HTTP.

ROI (Return on Investment) — показник рентабельності інвестицій, що демонструє ефективність використання капіталу.

RPN (Risk Priority Number) — кількісний показник, що використовується в методології FMEA для пріоритезації ризиків на основі їх критичності.

SSR (Server-Side Rendering) — метод формування контенту вебсторінок на сервері перед відправкою клієнту, що забезпечує швидке відображення інформації.

VaR (Value at Risk) — метрика фінансового ризику, що оцінює максимальні очікувані збитки портфеля за певний період із заданим рівнем довірчої ймовірності.

Аудит-лог (Audit Log) — системний журнал, що забезпечує реєстрацію всіх дій користувачів та змін конфігурацій для забезпечення простежуваності операцій.

Ендпоїнт (Endpoint) — кінцева точка доступу в межах API, через яку здійснюється запит до конкретного ресурсу або сервісу.

Метод Монте-Карло — група стохастичних методів, заснованих на імітаційному моделюванні для аналізу ризиків та прогнозування складних систем.

Коефіцієнт кореляції Спірмена — непараметричний статистичний показник, що оцінює силу та напрямок монотонного зв'язку між двома змінними.

ВСТУП

Сучасні бізнес-умови потребують впровадження інноваційних інструментів управління ризиками. Традиційні підходи, зокрема матриця ризиків та FMEA, забезпечують якісний аналіз, проте не дозволяють досягти належної точності при оцінюванні фінансових збитків та часових затримок. Водночас застосування методів симуляції Монте-Карло, аналізу чутливості та аналізу «витрати-вигоди» (Cost-Benefit Analysis) у межах стандартних табличних процесорів є надмірно трудомістким. Своєю чергою, спеціалізоване програмне забезпечення (наприклад, @RISK, Primavera) вирізняється високою вартістю та обмеженою доступністю через орієнтацію на десктопні платформи. Це призводить до розривів у даних, труднощів із візуалізацією та зниження ефективності аналітичних процесів. З огляду на це, розробка вебзастосунку, що інтегрує якісні та кількісні методи в єдине середовище для оперативного розрахунку впливу ризиків, пріоритизації загроз та забезпечення цілісності даних, є стратегічно актуальною.

Метою роботи є проєктування та розробка вебсистеми аналізу й управління ризиками на основі кількісних методів із застосуванням сучасних вебтехнологій та імовірнісних алгоритмів для автоматизації процесів прогнозування фінансових і часових втрат проєкту.

Для досягнення поставленої мети визначено чотири завдання:

1. Проаналізувати ринкові рішення, підходи, методи та алгоритми кількісної і якісної оцінки ризиків.

2. Обґрунтувати вибір технологічного стеку для розробки вебсистеми «VIKTORIA».

3. Спроєктувати та розробити програмне забезпечення для платформи управління ризиками.

4. Здійснити програмну реалізацію клієнт-серверної архітектури, обчислювального ядра.

Апробація результатів. Основні наукові та практичні результати дипломного проєкту пройшли апробацію на 3-й Міжнародній науково-практичній конференції «Science and Technology: New Horizons of Development» (м. Прага, Чехія, 3–5 червня 2026 р.). За результатами дослідження підготовлено та опубліковано матеріали у відповідному міжнародному збірнику наукових праць.

Робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел (20 найменувань) та двох додатків. Загальний обсяг дипломної роботи становить 75 сторінок, включаючи 20 ілюстрацій та 8 таблиць.

У першому розділі здійснено аналіз предметної області, огляд аналогічних програмних продуктів, постановку задачі та вибір технологічного стеку (NestJS та Next.js).

У другому розділі обґрунтовано вибір алгоритмів та описано математичні моделі: матриці ризиків, FMEA, метод Монте-Карло, аналіз Парето, діаграми Торнадо, аналіз «витрати-вигоди» (Cost-Benefit), а також методологію розрахунку рангової кореляції Спірмена.

У третьому розділі викладено технічні аспекти реалізації системи: архітектуру REST API, структуру реляційної бази даних та програмну реалізацію обчислювального ядра на серверній частині.

У четвертому розділі розглянуто умови розгортання системи, сценарії взаємодії з користувачем, інструменти візуалізації аналітичних даних та функціонування системи аудит-логу для моніторингу змін на прикладі практичного енергетичного проєкту.

1 АНАЛІЗ ЗАДАЧІ РОЗРОБКИ ВЕБСИСТЕМИ УПРАВЛІННЯ РИЗИКАМИ

У цьому розділі наведено комплексну постановку завдання щодо розробки вебзастосунку, який інтегрує якісні та кількісні методи в єдине автоматизоване середовище для оперативного оцінювання впливу ризиків на проєкт. Актуальність проєкту підтверджена результатами огляду та порівняльного аналізу наявних аналогів, які засвідчили відсутність гнучких рішень із підтримкою комплексного моделювання. Визначено цільову аудиторію продукту та формалізовано багаторівневу структуру вхідних і вихідних даних, необхідних для проведення аналітичних розрахунків. Обґрунтовано вибір сучасного технологічного стека, зокрема NestJS, Next.js та PostgreSQL, що дозволяє реалізувати високопродуктивну, безпечну та масштабовану клієнт-серверну архітектуру.

1.1 Постановка прикладної задачі

Програмний продукт розроблено для менеджерів проєктів, ризик-менеджерів, керівників малого та середнього бізнесу, а також аналітиків у сферах IT, FinTech, будівництва, охорони здоров’я, промисловості та енергетики. Метою системи є мінімізація суб’єктивізму під час оцінювання ризиків шляхом надання фахівцям уніфікованого інструментарію для об’єктивного розрахунку ймовірностей та прогнозування потенційних втрат, у роботі [7].

Функціонал системи охоплює повний цикл роботи з даними: від формування реєстру ризиків із класифікацією загроз та фіксацією тригерів їх настання до кількісного аналізу. На відміну від систем, що обмежуються лише якісними оцінками, даний продукт базується на модульній архітектурі обчислювального

ядра. Воно інтегрує комплекс із п'яти кількісних методів, зокрема імітаційне моделювання Монте-Карло, аналіз чутливості та імовірнісні підходи [10]. Аналітичний модуль забезпечує пріоритизацію загроз через порівняння результатів різних методів, формуючи узгоджений рейтинг із застосуванням коефіцієнтів рангової кореляції.

Вебсистема дозволяє обирати стратегії реагування на ризики (уникнення, мінімізація, передача, прийняття) на основі аналізу «витрати–вигоди» (Cost-Benefit) [11]. Прозорість та безперервний моніторинг забезпечуються завдяки інтегрованому аудиту змін параметрів. Завершальний етап циклу передбачає автоматичну генерацію звітів у форматі PDF та візуалізацію даних за допомогою інтерактивних матриць ризиків, діаграм Парето та Торнадо. Це надає керівництву можливість приймати обґрунтовані управлінські рішення на основі точних математичних моделей.

1.2 Огляд та порівняння програмних аналогів

Сучасний етап розвитку характеризується переходом до динамічного моделювання, проте аналіз ринку показує, що більшість рішень мають вузьку галузеву спеціалізацію [7]. Наукові огляди кількісних методів свідчать про те, що існуючі інструменти часто фокусуються лише на одній сфері, наприклад зовнішньоекономічній діяльності, залишаючи поза увагою комплексний аналіз бюджету [8].

Одним із найбільш відомих рішень для фінансового моделювання є програмний комплекс «@RISK» від компанії «Lumivero», який реалізує симуляцію Монте-Карло, проте функціонує як надбудова до Excel і має обмежені можливості візуалізації першопричин загроз. Система «RiskyProject» («Intaver Institute») інтегрує кількісні методи з календарним плануванням, але її орієнтація на лінійні графіки Ганта є малоефективною для сучасних гнучких методологій. Платформа

«BowTieXP» від «CGE Risk» спеціалізується на візуалізації бар'єрів безпеки, проте залишається ізольованою від імовірнісного фінансового моделювання загального бюджету проєкту. Спеціалізовані рішення на кшталт «Relyence» характеризуються надмірною академічністю, а корпоративні платформи «Sphera» або «DNV Synergi Life» є високовартісними та орієнтованими переважно на розслідування вже наявних інцидентів.

Для обґрунтування доцільності розробки системи «VIKTORIA» проведено порівняльну характеристику основних аналогів, результати якої наведено в таблиці 1.1.

Таблиця 1.1 — Порівняльна характеристика засобів аналізу ризиків

Інструмент	**Якісний**	**Монте-Карло**	**ROI**	**Web**	**Ціна**
MS Excel + макроси	+	Ручне	Ручне	−	Платно
@RISK / Palisade	−	+	+	−	Платно (~$1500)
Primavera Risk	+	+	−	−	Платно
VIKTORIA	+	+	+	+	Open-source

Дані таблиці 1.1 підтверджують відсутність на ринку гнучкої вебсистеми, яка б гармонійно поєднувала структурні методи ідентифікації з імовірнісними розрахунками.

Наукова новизна системи «VIKTORIA» полягає в інтеграції різнопланових методів аналізу (від Матриці та FMEA до симуляції Монте-Карло та аналізу Парето) над спільною моделлю даних. Впровадження автоматичної кореляційної валідації дозволяє кількісно підтвердити доцільність застосування складних методів для кожного конкретного проєкту [9].

Практична цінність рішення полягає у можливості отримання критичних метрик за секунди через зручний вебінтерфейс, що значно скорочує часові витрати

менеджера.

1.3 Вхідні та вихідні дані системи

Система функціонує на основі трьох рівнів даних: облікових записів, параметрів портфеля проєктів та деталізованого реєстру ризиків. До вхідних даних належать автентифікаційні відомості користувача та ключові показники проєктів, зокрема бюджет, який є базовою величиною для проведення кількісного моделювання [6]. Реєстр ризиків охоплює понад шістнадцять параметрів для кожної загрози, включаючи її опис, класифікацію та відповідальних осіб. Якісна експертиза здійснюється шляхом оцінювання ймовірності, впливу та виявлюваності за п’ятибальною шкалою, що відповідає міжнародним стандартам управління проєктами [1] та принципам ризик-менеджменту [2].

Вихідні дані системи включають результати застосування семи методів аналізу, що визначають зони пріоритетності та рівні критичності загроз. Кількісне моделювання забезпечує розрахунок очікуваних втрат та показника вартості під ризиком (VaR) з впевненістю 95%, згідно з методами математичного прогнозування економічних ризиків [5, 9]. Платформа генерує дані для побудови графіків розподілу втрат, аналізу чутливості та оцінки кумулятивного впливу кожної події. Економічний модуль здійснює розрахунок коефіцієнтів окупності інвестицій у безпеку (ROSI) та визначає ефективність стратегій реагування. Коректність математичних обчислень верифікується шляхом розрахунку коефіцієнта рангової кореляції для забезпечення узгодженості між різними методами оцінювання. Фінальним результатом роботи системи є автоматично сформований аналітичний звіт у форматі PDF.

1.4 Вибір технологічного стеку

Вибір програмно-технологічного забезпечення для реалізації інтелектуальної автоматизованої системи управління ризиками проєктів «VIKTORIA» зумовлений потребою у створенні високопродуктивної, масштабованої та безпечної вебплатформи. Використання сучасних інструментальних засобів гарантує точність математичних розрахунків і стабільність функціонування системи під час виконання значних обсягів ітераційних обчислень, зокрема моделювання методом Монте-Карло, що обґрунтовано в працях [3] – [5].

Серверна частина системи спроєктована на основі модульної архітектури, що забезпечує логічну ізоляцію окремих методів аналізу. Ключові характеристики обраного бекенд-стеку наведено в таблиці 1.2.

Таблиця 1.2 — Технологічне забезпечення серверної частини системи

Технологія	**Версія**	**Призначення**
NestJS	11.x	Модульний фреймворк (Controller-Service-Module)
TypeScript	5.7	Сувора типізація
Prisma ORM	7.8	Type-safe ORM, міграції, генерація клієнта
PostgreSQL	15	Реляційна СУБД для зберігання даних
Passport + JWT	11 / 4	Автентифікація через токени
bcrypt	6	Хешування паролів
class-validator	0.15	Валідація DTO
PDFKit	0.16	Генерація PDF-звітів з кирилицею

Для реалізації клієнтської частини було обрано фреймворк Next.js, що забезпечує високу швидкість рендерингу інтерфейсу та оптимізовану навігацію. Перелік інструментів, використаних для фронтенд-розробки, наведено у таблиці

1.3.

Таблиця 1.3 — Технологічне забезпечення клієнтської частини системи

Технологія	**Версія**	**Призначення**
Next.js	16.2 (App Router)	React-фреймворк з SSR і файловим роутингом
React	19.2	UI-бібліотека
Tailwind CSS	4	Utility-first стилізація
Recharts	3.8	Інтерактивні графіки (гістограми, торнадо, Парето)
Axios	1.16	HTTP-клієнт із інтерсепторами
react-hot-toast	2.6	Сповіщення користувача

Систему розгорнуто за допомогою Docker Compose, що включає контейнери PostgreSQL, сервера та клієнтського додатка. Використання персистентних томів гарантує цілісність даних, а інтегровані шрифти забезпечують коректне відображення PDF-звітів. Архітектура рішення базується на розподілі відповідальності між контролерами, сервісами та PrismaService, а впроваджений механізм перевірки прав доступу надійно захищає реєстри ризиків від несанкціонованого перегляду [14].

2 ОГЛЯД МЕТОДІВ ТА ЗАСОБІВ ОЦІНКИ РИЗИКІВ

Цей розділ присвячений теоретичному обґрунтуванню та детальному опису математичних моделей і алгоритмів, що становлять обчислювальну базу системи. Проаналізовано інструментарій якісного оцінювання, зокрема класичну матрицю ризиків та методологію FMEA, яка забезпечує ідентифікацію прихованих загроз шляхом інтеграції фактора виявлюваності. Значну увагу приділено методам стохастичного кількісного аналізу: імітаційному моделюванню Монте-Карло для розрахунку ризикової вартості (VaR), аналізу Парето, діаграмам Торнадо та оцінці економічної ефективності заходів реагування (Cost-Benefit Analysis). У завершальній частині розглянуто застосування апарату непараметричної статистики — коефіцієнта рангової кореляції Спірмена — для автоматизованої валідації та верифікації результатів застосованих методів оцінювання.

2.1 Обґрунтування обраних алгоримтів

Методологічний комплекс системи «VIKTORIA» розроблено на засадах міжнародних стандартів ISO 31000 та IEC 31010, що забезпечує відповідність процедур управління ризиками світовим практикам. Система застосовує багаторівневий підхід, що трансформує суб’єктивні експертні оцінки у точні математичні прогнози [2, 10].

Якісний аналіз починається з класичної матриці ризиків для первинного ранжування за ймовірністю та рівнем впливу. Для мінімізації обмежень цього методу інтегровано FMEA-аналіз. Додавання критерію виявлюваності дозволяє ідентифікувати приховані ризики, які потребують пріоритетного реагування через складність їх своєчасної діагностики. Математичну обґрунтованість цього підходу

верифіковано за допомогою коефіцієнта рангової кореляції Спірмена. Зокрема, значення коефіцієнта нижче 0,7 свідчить про необхідність застосування FMEA через суттєвий вплив фактора виявлення на структуру пріоритетів [9].

Кількісний аналіз реалізовано шляхом стохастичного моделювання за методом Монте-Карло, що конвертує дискретні експертні оцінки у статистичні розподіли ймовірностей (10–90%) [5]. Модель виключає граничні значення (0% та 100%), оскільки вони не відповідають природі невизначеності. Після проведення 10 000 ітерацій система визначає сукупні фінансові втрати, необхідний резервний капітал (на основі показника VaR 95%) та прогноз відхилень календарних планів, що є критично важливим для IT-менеджменту [6].

Архітектурна логіка системи повністю відповідає стандартам PMBOK (PMI), забезпечуючи чітке розмежування між якісним ранжуванням та кількісним моделюванням впливу на бюджет і графік проєкту [1].

Фінальна пріоритезація ресурсів здійснюється за допомогою аналізу Парето (виокремлення 20% ризиків, що спричиняють 80% негативного впливу) та аналізу чутливості за діаграмою Торнадо, яка візуалізує потенційну амплітуду коливань бюджету.

Стратегічне прийняття рішень базується на принципах COSO ERM та COBIT із застосуванням аналізу витрат і вигод (Cost-Benefit Analysis). Система обчислює рентабельність інвестицій у безпеку (ROI): позитивне значення ROI обґрунтовує доцільність впровадження заходів захисту, тоді як від’ємне вказує на економічну доцільність прийняття ризику без додаткових інвестицій.

2.2 Матриця ризиків та FMEA

Першим етапом аналізу в системі, що розробляється, є застосування кількісної матриці ризиків, яка слугує базовим інструментом для оцінки рівня загроз. Відповідно до міжнародного стандарту ISO 31000 та методології PMBOK,

матриця ймовірності та наслідків є ключовим механізмом для переходу від якісного до кількісного аналізу. У межах цього підходу величина ризику визначається як функція двох незалежних змінних. Розрахунковий бал (Score) визначається за формулою 2.1:

$$Score = Probability \times Impact \quad (2.1)$$

де $Probability$ — експертна оцінка ймовірності настання небажаної події за п’ятибальною шкалою;

$Impact$ — експертна оцінка рівня впливу на проєкт у межах від 1 до 5 [1].

Використання лінійної або нелінійної п'ятибальної шкали дає змогу уникнути надмірного усереднення даних, забезпечуючи точну сегментацію ризикового ландшафту проєкту. Ризики класифікуються за чотирма зонами критичності:

Низька (1–4 бали): ризики не потребують активних стратегій реагування, проте підлягають моніторингу в межах реєстру.

Середня (5–9 балів): передбачають виділення резервів на випадок виникнення непередбачуваних витрат.

Висока (10–16 балів): вимагають обов'язкового проактивного реагування та постійного контролю з боку керівника проєкту.

Екстремальна (17–25 балів): критичні загрози, що створюють ризик зриву проєкту та потребують негайного втручання або ескалації на рівень спонсора.

Зважаючи на те, що стандартна двовимірна матриця не враховує здатність команди своєчасно ідентифікувати загрозу до моменту її трансформації у проблему, систему було доповнено методологією FMEA (Failure Mode and Effects Analysis — аналіз видів та наслідків відмов), що є стандартом в інженерії та управлінні якістю (зокрема, згідно з IATF 16949). Інтеграція показника виявлення (Detection) як третього виміру дозволяє обчислити пріоритетне число ризику (RPN) за формулою 2.2.

$$RPN = Probability \times Impact \times Detection \quad (2.2)$$

де $Detection$ — показник здатності виявити ризик до моменту його настання.

Запропонована шкала використовує інвертований підхід: значення «1»

відповідає ризикам, що легко ідентифікуються завдяки наявним системам контролю, тоді як значення «5» вказує на приховані загрози, виявлення яких до моменту настання критичних наслідків є вкрай складним. Впровадження методології FMEA у вебсистему сприяє виявленню «сліпих зон» проєкту — ризиків із середнім рівнем імовірності та високим ступенем прихованості, які часто залишаються поза увагою під час застосування стандартної матриці оцінювання.

2.3 Імітаційне моделювання Монте-Карло

Попри практичну цінність експертних оцінок, вони мають властивий їм елемент суб'єктивності. Для врахування стохастичного характеру невизначеності та комплексної оцінки впливу різних факторів на фінансові результати проєкту в систему впроваджено метод імітаційного моделювання Монте-Карло. Цей аналітичний інструмент дозволяє перейти від використання точкових оцінок до аналізу розподілу ймовірностей.

У межах розробленої системи дискретні експертні дані трансформуються у статистичні ймовірності в діапазоні 10–90%. Обсяг потенційних збитків моделюється не фіксованим значенням, а випадковою величиною, що відповідає законам теорії ймовірностей. Для забезпечення обчислювальної ефективності за умови збереження репрезентативності результатів на початковому етапі застосовується рівномірний розподіл (Uniform Distribution), згідно з яким випадкове значення збитку визначається за формулою 2.3:

$$X = C_{min} + R \times (C_{max} - C_{min}) \quad (2.3)$$

де C_{min} та C_{max} — задані експертами нижня та верхня межі діапазону фінансових втрат (або відхилень у розкладі);

R — випадкове число, згенероване алгоритмом в інтервалі $[0; 1]$.

Система виконує 10 000 ітерацій моделювання, що, відповідно до закону великих чисел, забезпечує високу точність та стабільність результатів. У межах

кожної ітерації алгоритм здійснює стохастичне моделювання ризикових подій на основі заданих імовірностей, визначаючи їхній фінансовий вплив. Сукупний збиток для окремої ітерації розраховується як сума впливів усіх реалізованих ризиків.

На основі отриманого масиву даних система формує гістограму розподілу сукупних ризиків та розраховує показник VaR (Value at Risk — ризикова вартість). Зокрема, метрика VaR 95% визначає максимальний рівень збитків, який не буде перевищений із імовірністю 95% [2]. Це забезпечує керівництво проєкту науково обґрунтованими даними для формування фінансового резерву. Наприклад, за базового бюджету проєкту в $100 000 та показника VaR 95%, що становить $15 000, менеджмент отримує математичне підґрунтя для формування загального бюджету в розмірі $115 000.

2.4 Аналіз Парето та діаграма Торнадо

Ефективне управління проєктами передбачає оптимізацію обмежених ресурсів: часу, бюджету та людського капіталу. Для фокусування зусиль на пріоритетних завданнях у систему впроваджено метод аналізу Парето, що базується на принципі 80/20: близько 20% критичних ризиків зумовлюють 80% загального негативного впливу на проєкт.

Система автоматично ранжує ризики за показниками RPN або EMV та розраховує їхню кумулятивну частку. Візуалізація даних за допомогою діаграми Парето дозволяє команді оперативно виявляти критичні «вузькі місця» та розробляти цілеспрямовані стратегії мінімізації загроз.

Додатковим інструментом є одновимірний аналіз чутливості, представлений діаграмою Торнадо. Цей метод дозволяє ранжувати фактори за ступенем їхнього впливу на фінансовий результат проєкту, оцінюючи зміну показників при зміні одного фактора за незмінності інших.

На діаграмі Торнадо ризики впорядковані за рівнем невизначеності, що формує характерну графічну структуру. Такий підхід дає керівництву можливість оцінити діапазон коливань — від оптимістичного до песимістичного сценарію — для кожного ризику окремо. Інтегроване застосування методів Парето та Торнадо забезпечує обґрунтовану базу для прийняття стратегічних рішень.

2.5 Аналіз витрат і вигод (Cost-Benefit)

Процеси ідентифікації та оцінки ризиків є недостатньо ефективними без формування економічно обґрунтованих стратегій реагування. Ключовим інструментом забезпечення стратегічної доцільності системи виступає метод аналізу витрат і вигод (Cost-Benefit Analysis, CBA), який визначає фінансову ефективність впровадження антиризикових заходів. Базовим критерієм у цьому контексті є дотримання принципу, згідно з яким витрати на реалізацію стратегії мінімізації ризиків (Mitigation Strategy) не мають перевищувати прогнозований економічний ефект від її впровадження.

Для кількісної оцінки такої доцільності застосовується показник рентабельності інвестицій у заходи реагування на ризики (ROI — Return on Investment), розрахунок якого здійснюється за формулою 2.4.

$$ROI = \frac{Benefit - MitigationCost}{MitigationCost} \times 100\% \quad (2.4)$$

де $Benefit$ — очікувана вигода. Вона розраховується як різниця між початковими (прогнозованими до впровадження заходів) збитками та залишковими збитками (Residual Risk), які залишаться після реалізації стратегії;

$MitigationCost$ — пряма вартість впровадження стратегії реагування (наприклад, купівля ліцензій, наймання додаткових фахівців, страхування).

Цей інструмент дозволяє мінімізувати типову управлінську помилку — надмірні витрати на забезпечення безпеки. Якщо розрахунковий показник ROI є

від'ємним або близьким до нуля, система сигналізує про економічну недоцільність обраної стратегії (наприклад, уникнення або зниження ризиків). У такому разі команді рекомендується перейти до стратегії прийняття ризику (Risk Acceptance) та спрямувати ресурси до фінансового резерву. Застосування методології СВА трансформує процес управління ризиками з абстрактної технічної процедури на прозорий бізнес-процес, орієнтований на ефективне збереження капіталу інвесторів та стейкхолдерів проєкту [15].

2.6 Рангова кореляція Спірмена

Для підтвердження наукової обґрунтованості архітектури системи та верифікації узгодженості інтегрованих методів застосовується апарат непараметричної статистики. Зокрема, ключовим завданням є оцінка впливу параметра виявлення (Detection), визначеного згідно з методологією FMEA, на пріоритетність ризиків, встановлену класичною матрицею. З цією метою впроваджено механізм автоматичного розрахунку коефіцієнта рангової кореляції Спірмена.

На відміну від лінійної кореляції Пірсона, критерій Спірмена базується на аналізі рангів, а не абсолютних значень, що робить його оптимальним інструментом для опрацювання експертних бальних оцінок. Розрахункова формула коефіцієнта p (ро) наведена у формулі 2.5:

$$p = 1 - \frac{6 \sum d_i^2}{n(n^2-1)} \quad (2.5)$$

де d_i — різниця між рангами і-го ризику в двох досліджуваних списках (наприклад, ранг за методом Score та ранг за методом RPN);

n — загальна кількість ідентифікованих ризиків у реєстрі проєкту.

Інтерпретація цього показника є визначальною для валідації системи. Коефіцієнт кореляції p варіюється в діапазоні від -1 до +1. Значення, близькі до 1,

вказують на ідентичність ранжування, отриманого за допомогою класичної матриці та методології FMEA, що ставить під сумнів доцільність застосування фактора виявлення (Detection). Проте в складних IT- та інженерних проєктах цей коефіцієнт зазвичай становить 0,6–0,8, що підтверджує наявність значущої, але не абсолютної кореляції. Саме різниця в рангах d_i дозволяє обґрунтувати ефективність алгоритму: вона демонструє, як приховані загрози з високим показником виявлення (Detection), що мали низький пріоритет у двовимірній матриці, коректно підвищують свій статус у рейтингу RPN. Це запобігає виникненню критичних ризиків [17].

3 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ВЕБСИСТЕМИ

У цьому розділі викладено технічні аспекти проєктування та програмної реалізації вебсистеми «VIKTORIA». Описано архітектуру взаємодії компонентів, що ґрунтується на принципах REST API та модульному підході до побудови захищеного серверного середовища. Докладно представлено концептуальну модель реляційної бази даних, яка забезпечує оптимізовані зв'язки між сутностями користувачів, проєктів, реєстрів ризиків та журналів аудиту. Окрему увагу приділено алгоритмічній реалізації математичного ядра, зокрема логіці обробки масивів даних та механізмам асинхронного виконання ресурсомістких ітераційних обчислень, необхідних для функціонування інтегрованих методів аналізу.

3.1 Архітектура системи та специфікація API

Загальна схема взаємодії компонентів та архітектура розробленого додатку наочно представлена на рисунку 3.1.

Згідно з рисунком 3.1, клієнтська частина застосунку побудована на базі фреймворку Next.js та здійснює взаємодію із сервером за допомогою HTTP-клієнта Axios. Для оптимізації процесів розробки, підтримки та масштабованості архітектури фронтенду застосовано підхід маршрутизації App Router.

Серверна частина (backend), відображена на структурній схемі (рисунок 3.1), реалізована на фреймворку NestJS. Архітектура побудована на модульному принципі, що забезпечує ізоляцію бізнес-логіки системи. Перелік основних модулів та їхні функціональні обов'язки систематизовано у таблиці 3.1.

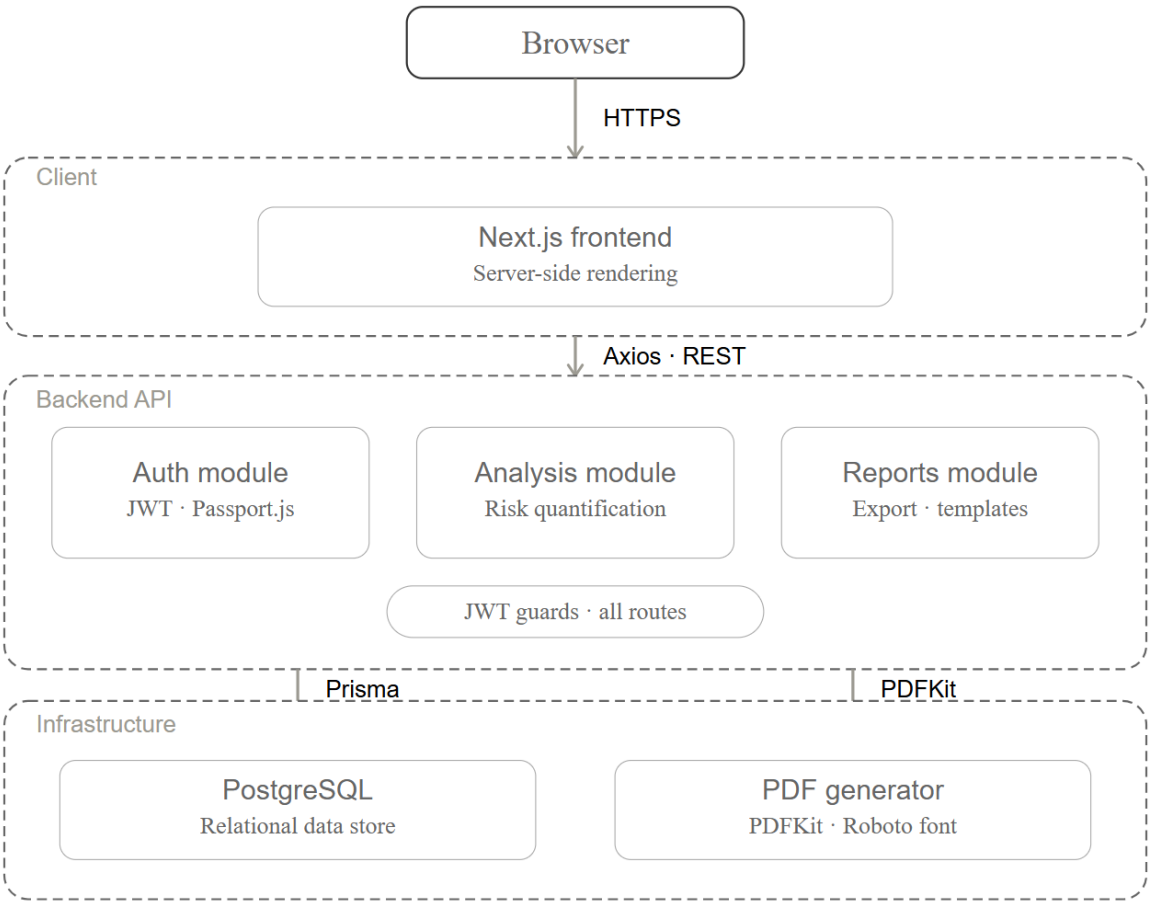


Рисунок 3.1 — Загальна схема архітектури додатку

Таблиця 3.1 — Відповідальність модулів серверної частини

Модуль	**Файли**	**Відповідальність**
AuthModule	auth/*	Реєстрація, логін, JWT-стратегія, guard'и
ProjectsModule	projects/*	CRUD проєктів, перевірка ownership
RisksModule	risks/*	CRUD ризиків + автоматичний AuditLog
AnalysisModule	analysis/*	6 методів аналізу (matrix, fmea, monte-carlo, tornado, pareto, cost-benefit)
ComparisonModule	comparison/*	Кореляція Спірмена між Матрицею і FMEA
ReportsModule	reports/*	Генерація PDF-звіту
PrismaModule	prisma/*	Spільний клієнт БД

Для забезпечення конфіденційності та цілісності даних інтегровано багаторівневу систему безпеки, що базується на таких механізмах:

LocalStrategy (passport-local): автентифікація користувача шляхом перевірки облікових даних (email та пароль) за допомогою алгоритму bcrypt.compare.

JwtStrategy (passport-jwt): валідація токенів доступу, що передаються у заголовку Authorization (Bearer scheme).

JwtAuthGuard: механізм контролю доступу, що застосовується через декоратор `@UseGuards` до всіх контролерів, за винятком загальнодоступних маршрутів (/auth/login та /auth/register).

Взаємодія з системою розпочинається з етапу автентифікації, під час якого користувач надає електронну адресу та пароль. У разі успішної верифікації система генерує об’єкт користувача та унікальний токен доступу, що дозволяє ініціювати захищену сесію та надати права доступу до ресурсів платформи.

Після отримання JWT-токена користувач отримує доступ до захищених функціональних модулів, зокрема до системи управління портфелем проєктів. Обмін даними здійснюється за допомогою об’єктів передачі даних (DTO), що забезпечує коректність створення та оновлення проєктних параметрів, а також отримання деталізованої інформації про поточний стан проєктів.

Ключовим функціоналом системи є управління даними про потенційні загрози. Система підтримує повний цикл операцій (CRUD) для реєстру ризиків, оперуючи такими показниками, як імовірність, ступінь впливу, виявлюваність та фінансові діапазони втрат. Задля забезпечення прозорості та відстеження історії змін, кожна модифікація даних автоматично реєструється в системі аудит-логу.

Аналітичний модуль генерує результати кількісних та якісних оцінок, які передаються на клієнтську частину у вигляді структурованих наборів даних. Ці дані адаптовані для візуалізації, включаючи побудову гістограм, діаграм Торнадо та кривих Парето.

Завдяки стандартизованому обміну даними між сервером та клієнтом забезпечується ефективна візуалізація аналітичних показників у вигляді

інтерактивних матриць пріоритетності та автоматичне формування звітів у форматі PDF. Такий підхід гарантує точність, оперативність та надійність управління ризиками проєкту.

3.2 Структура бази даних, діаграми компонентів

Для забезпечення надійного збереження інформації, цілісності зв'язків та швидкої обробки математичних запитів у системі «VIKTORIA» спроєктовано оптимізовану реляційну базу даних під управлінням СУБД «PostgreSQL». Концептуальну модель даних, що відображає основні сутності системи (користувачі, проєкти, ризики, записи логів) та зв'язки типу «один-до-багатьох» (1:N) між ними, наочно представлено на концептуальній діаграмі класів (рисунок 3.2).

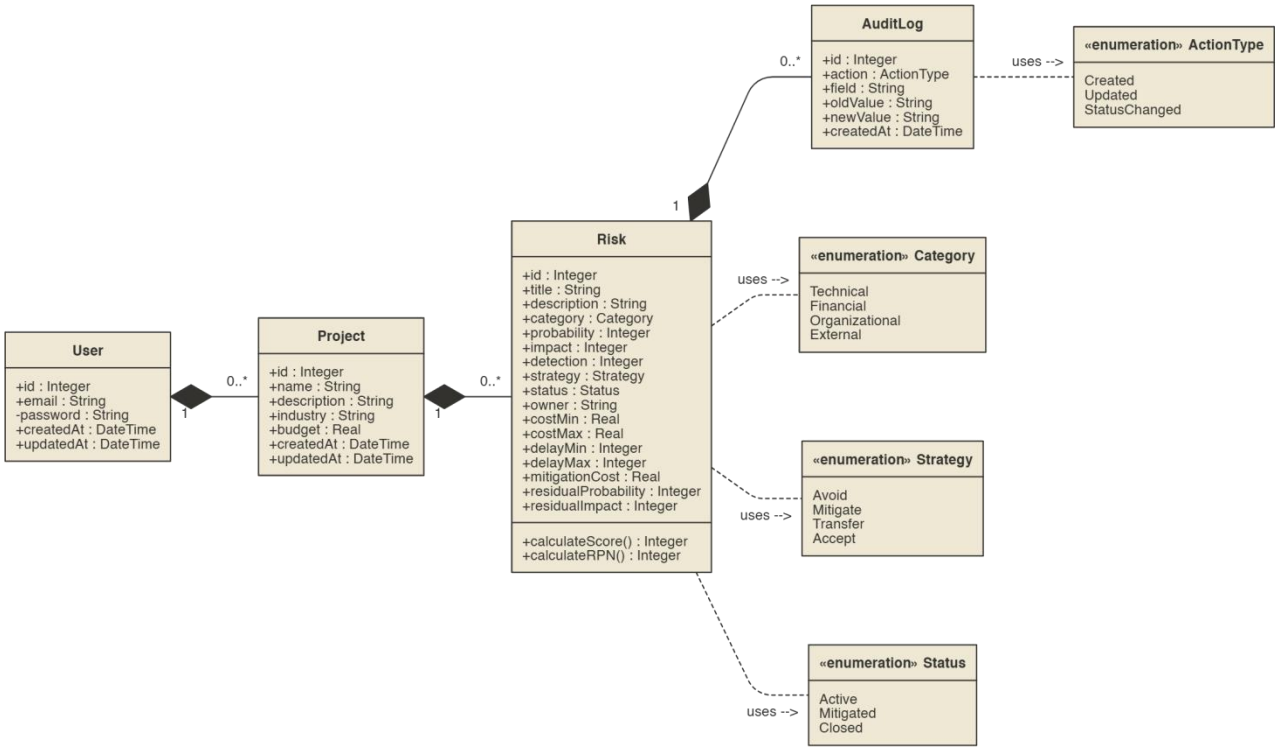


Рисунок 3.2 — Концептуальна діаграма класів (доменна модель) системи

Основою бази даних є сутність «Користувач» (User), призначена для керування процесом автентифікації та зберігання облікових даних. Для гарантування безпеки паролі зберігаються виключно у хешованому вигляді із застосуванням алгоритму Bcrypt. Детальна специфікація полів зазначеної сутності представлена у таблиці 3.2.

Таблиця 3.2 — Опис полів сутності User (Користувач)

Поле	**Тип**	**Опис**
id	Int (PK, auto)	Первинний ключ
email	String (unique)	Логін
password	String	Bcrypt-хеш (10 раундів)
createdAt / updatedAt	DateTime	Аудит часу

Кожен зареєстрований користувач наділений правом створення персональних проєктів. Об’єкт «Проєкт» (Project) включає метадані та фінансові показники, необхідні для проведення кількісного аналізу. Структуру відповідної таблиці наведено в таблиці 3.3.

Таблиця 3.3 — Опис полів сутності Project (Проєкт)

Поле	**Тип**	**Опис**
name, description, industry	String	Назва, опис, галузь
budget	Float?	Бюджет (₴) — обов’язковий для Монте-Карло
userId	Int (FK)	Власник, onDelete: Cascade

Основним елементом системи є реєстр ризиків (Risk). Ця таблиця консолідує повний перелік параметрів, забезпечуючи функціонування системи на основі єдиного джерела даних, що унеможливлює дублювання інформації під час проведення аналітичних процедур. Детальна специфікація полів реєстру

представлена в таблиці 3.4.

Таблиця 3.4 — Опис полів сутності Risk (Ризик)

Поле	**Тип**	**Призначення**
title, description	String	Назва, опис
category	String	Technical / Financial / Organizational / External
probability, impact	Int 1-5	Ймовірність і вплив (Матриця, Монте-Карло)
detection	Int 1-5	Виявлюваність (FMEA — третій вимір)
strategy	String	Avoid / Mitigate / Transfer / Accept
status	String	Active / Mitigated / Closed
owner	String?	Відповідальний
costMin, costMax	Float?	Діапазон фінансових втрат для Монте-Карло
delayMin, delayMax	Int?	Діапазон затримки в днях
mitigationCost	Float?	Вартість стратегії (Cost-Benefit)
residualProbability	Int?	Залишкова ймовірність після пом’якшення
residualImpact	Int?	Залишковий вплив після пом’якшення

Метою забезпечення прозорості та можливості відстеження історії змін оцінок, у системі застосовано патерн Event Sourcing шляхом впровадження сутності аудиторського лога (AuditLog). Цей механізм забезпечує автоматичну реєстрацію всіх модифікацій ключових параметрів ризиків, зокрема зміну їхнього статусу чи ймовірності, безпосередньо у базі даних. Детальний опис структури полів зазначеної сутності наведено у таблиці 3.5.

Таблиця 3.5 — Опис полів сутності AuditLog (Аудит-лог)

Поле	**Тип**	**Опис**
riskId	Int (FK)	Прив’язка до ризику
action	String	Created / Updated / StatusChanged
field	String?	Поле, що змінилось
oldValue, newValue	String?	Значення до і після
userId	Int?	Хто змінив
createdAt	DateTime	Часова мітка

Зв’язки між зазначеними таблицями налаштовані з використанням механізму каскадного видалення (Cascade Delete). Це забезпечує автоматичне видалення пов’язаних даних про ризики та записів журналу аудиту в разі видалення проєкту, що запобігає накопиченню в системі неактуальних (залишкових) записів. Функціональні межі програмного продукту та взаємодія користувача з даними деталізовані на діаграмі прецедентів (Use Case), наведеній на рисунку 3.3. Використання відношення типу «include» вказує на те, що будь-яка операція з керування ризиками автоматично ініціює запис у відповідну сутність AuditLog.

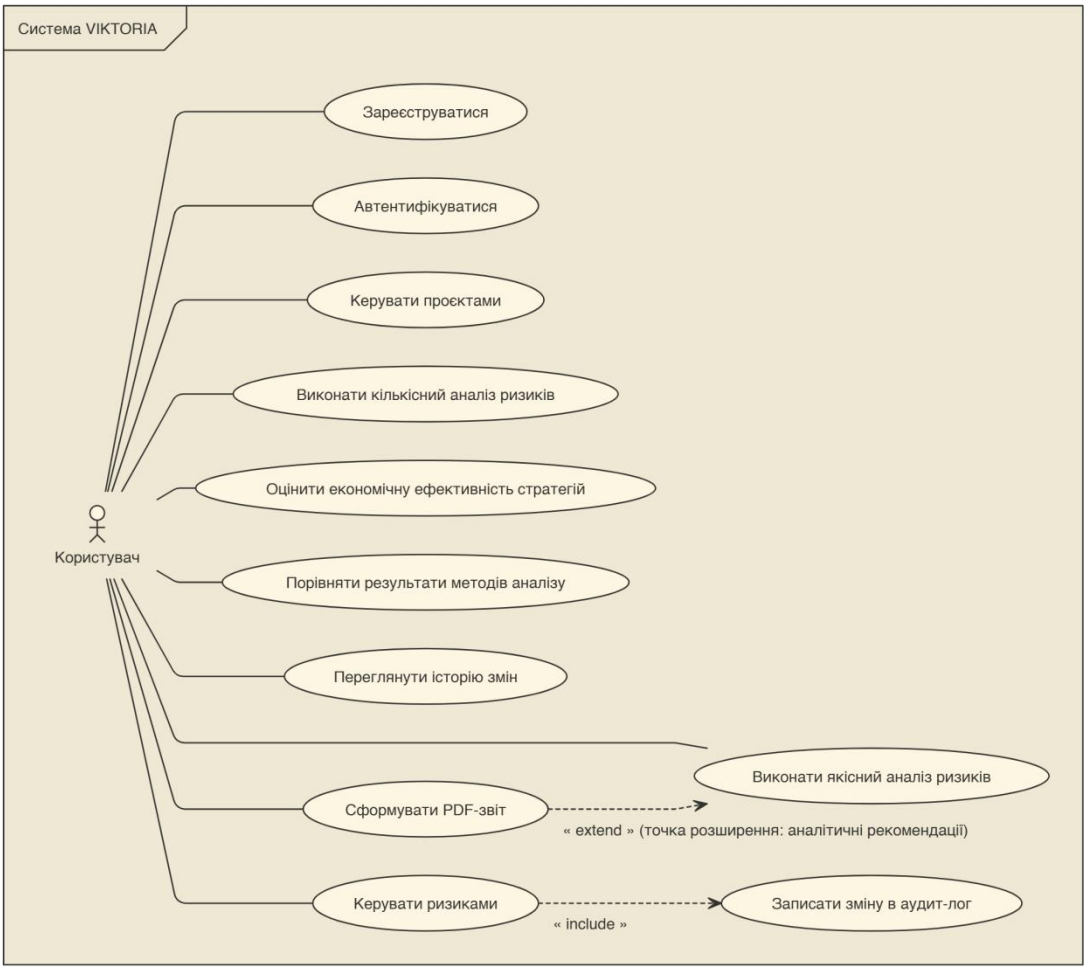


Рисунок 3.3 — Діаграма прецедентів системи (Use Case)

Програмна реалізація серверної частини та механізми взаємодії внутрішніх модулів представлені на діаграмі класів (рисунок 3.4). Архітектура системи базується на принципах інверсії управління та чіткому розмежуванні відповідальності між компонентами. Контролери забезпечують виключно маршрутизацію HTTP-запитів та верифікацію JWT-токенів із використанням відповідних Guard-класів, тоді як сервісний рівень концентрує виконання бізнес-логіки та математичних обчислень.

Взаємодію з базою даних централізовано реалізовано через PrismaService, що забезпечує типобезпечну роботу з даними. Архітектурне рішення гарантує сувору ізоляцію: кожен сервіс інтегрований із модулем управління проєктами для

перевірки прав доступу, що виключає можливість несанкціонованого звернення користувачів до реєстрів ризиків, які їм не належать.

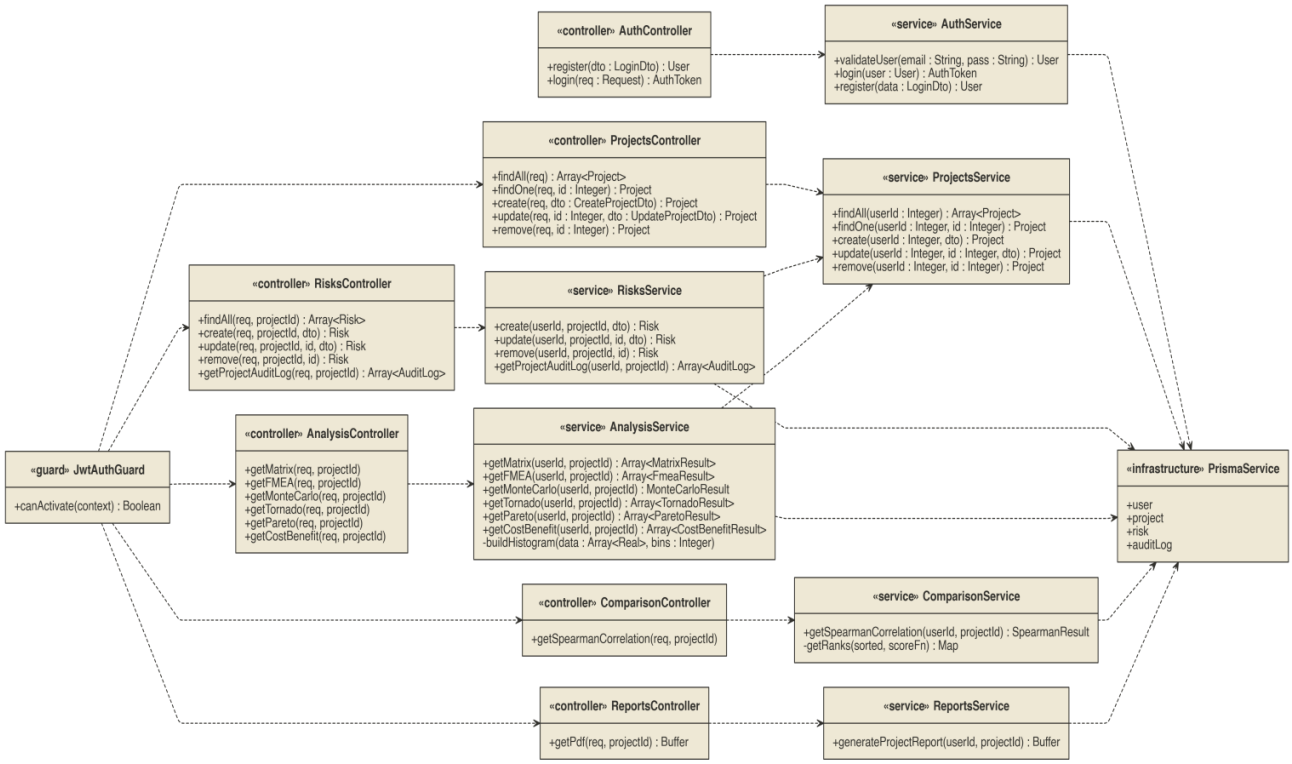


Рисунок 3.4 — Проєктна діаграма класів (архітектура backend)

Запропонована архітектура та логічна структура бази даних утворюють інтегрований комплекс, який гарантує стабільне функціонування системи «VIKTORIA» під час виконання високонавантажених ітераційних обчислень.

3.3 Програмна реалізація методів аналізу

Зазначений опис наведено згідно з кодом, поданим у додатку А.

Програмна реалізація методів аналізу в системі зосереджена в єдиному асинхронному сервісному класі, який забезпечує виконання всієї обчислювальної бізнес-логіки. Взаємодія з базою даних для отримання вхідних параметрів

здійснюється через інфраструктурний рівень. Кожен метод аналізу представлений окремою функцією, що опрацьовує ідентифікатори користувача та проєкту, перевіряє права доступу та виконує цільові алгоритмічні перетворення над масивом зареєстрованих ризиків.

Реалізація методу побудови матриці ризиків ґрунтується на застосуванні функції трансформації масивів. Процес розпочинається з вивантаження повного набору об’єктів ризиків із бази даних. Для кожного елемента масиву здійснюється розрахунок показника критичності шляхом множення цілочисельних значень ймовірності та впливу. На наступному етапі алгоритм ініціалізує змінну для визначення зони ризику, присвоюючи їй базове значення низького рівня. За допомогою послідовних умовних операторів програма проводить класифікацію отриманого бала відповідно до встановлених діапазонів: від 5 до 9 — середній рівень, від 10 до 16 — високий, від 17 і вище — екстремальний. Фінальним етапом роботи функції є формування та повернення оновленого масиву об’єктів, що містить початкові дані ризику, доповнені розрахунковим балом та відповідним ідентифікатором зони.

Алгоритм методу аналізу видів та наслідків відмов (FMEA) здійснює ітераційну обробку масиву загроз, отриманих із бази даних. Для кожного об’єкта формується окрема структура даних, що включає ідентифікатор, найменування та базові параметри. Ключовим етапом процедури є розрахунок пріоритетного числа ризику (RPN), яке визначається як добуток трьох показників: імовірності виникнення, ступеня впливу та здатності до виявлення. Сформований масив об’єктів передається до стандартного алгоритму сортування. Використання вбудованої стрілочної функції зворотного виклику забезпечує порівняння пріоритетних чисел суміжних елементів шляхом обчислення їхньої різниці, що дозволяє впорядкувати результуючий масив за спаданням рівня критичності.

Найбільш ресурсомістким етапом розробки є програмна реалізація методу Монте-Карло. З огляду на високу обчислювальну складність імітаційного моделювання, алгоритм методу доцільно формалізувати за допомогою UML-

діаграми діяльності. Як відображено на рисунку 3.5, процедура передбачає застосування механізму верифікації бюджету з подальшим виконанням багатокрокового циклу, що складається з 10 000 ітерацій. Використання зазначеної архітектури забезпечує послідовне накопичення стохастичних даних із їх подальшим сортуванням для розрахунку показника ризикової вартості (VaR).

Алгоритм реалізовано через вкладені цикли. Зовнішній цикл забезпечує виконання десяти тисяч ітерацій, ініціалізуючи на кожному кроці змінні для накопичення фінансових втрат та часових затримок. Внутрішній цикл послідовно обробляє перелік ризиків: для кожного ризику значення ймовірності порівнюється з псевдовипадковим числом. Якщо згенероване число не перевищує поріг імовірності, система фіксує настання ризикової події.

Розрахунок наслідків здійснюється згідно з алгоритмом умовного розподілу: за наявності встановлених цінових порогів фінансові втрати та часові затримки визначаються як випадкові величини в межах заданого діапазону; за відсутності визначених порогів розрахунок проводиться як частка від загального бюджету відповідно до значень, передбачених довідником рівнів впливу

Після обробки всіх ризиків підсумкові значення ітерації додаються до відповідних масивів результатів.

Після завершення симуляцій масиви результатів сортуються за зростанням. Показник ризикової вартості (VaR) на рівні довірчого інтервалу 95% визначається за індексом, що відповідає 95-му перцентилю відсортованого масиву. Середні значення розраховуються як середнє арифметичне елементів, а максимальна затримка — як останній елемент впорядкованого масиву часових показників.

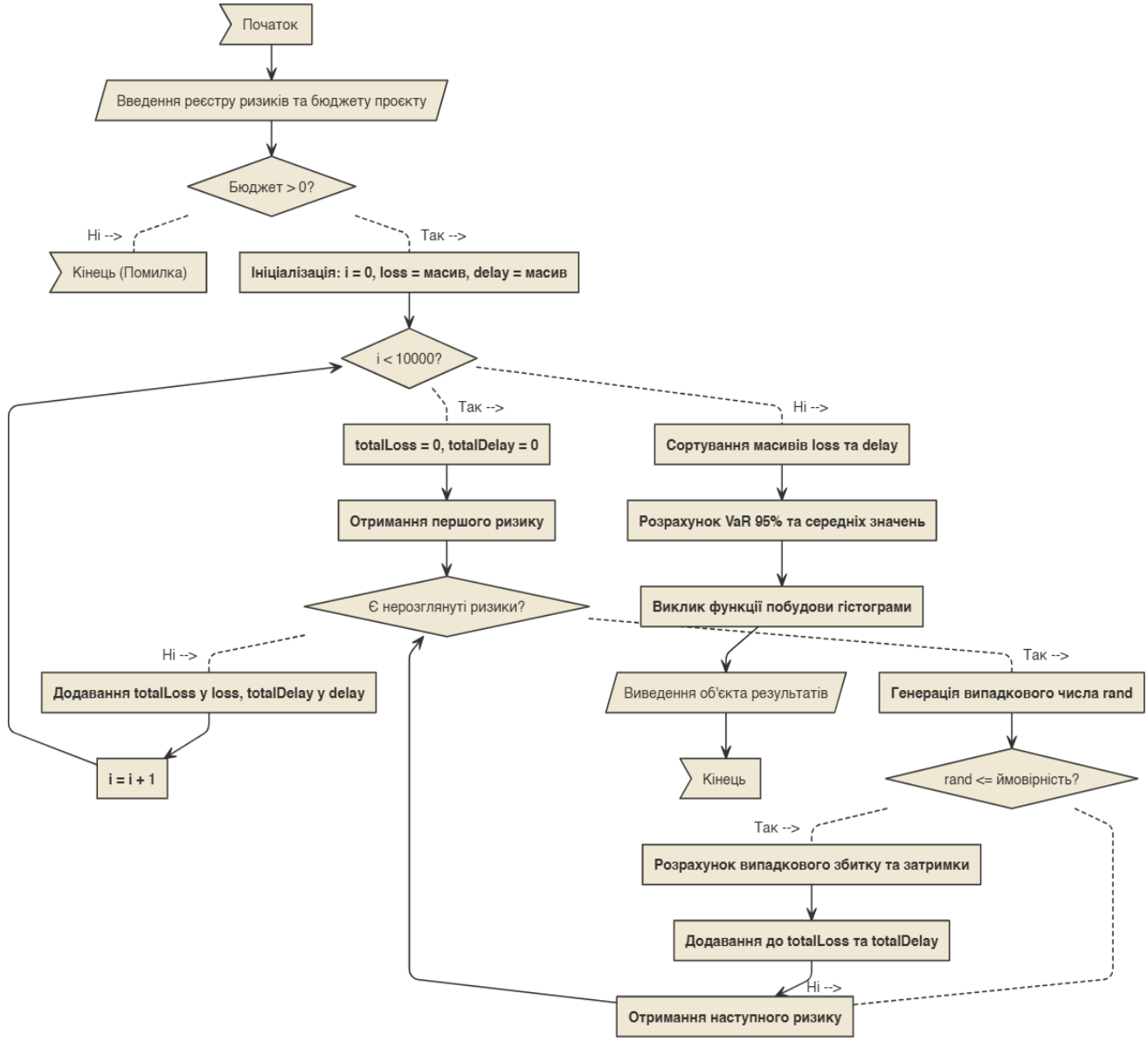


Рисунок 3.5 — Блок-схема алгоритму імітаційного моделювання Монте-Карло

Для візуалізації результатів реалізовано функцію генерації гістограми. Діапазон значень ділиться на сегменти фіксованої ширини, після чого алгоритм розподіляє результати симуляції по відповідних сегментах, інкрементуючи лічильники для формування статистичного розподілу.

Програмна реалізація аналізу чутливості для діаграми Торнадо передбачає ініціалізацію резервного значення бюджету (у разі його відсутності) та словника для конвертації балів впливу. У процесі ітеративного опрацювання масиву

формується набір об'єктів, ключовим показником яких є максимальний фінансовий вплив. Розрахунок цього показника здійснюється шляхом множення бюджету проєкту на відповідний відсотковий коефіцієнт впливу. На завершальному етапі отриманий масив сортується за спаданням розрахованого фінансового показника.

Реалізація методу Парето базується на послідовному виконанні ланцюгових методів масивів. На першому етапі для кожного ризику розраховується базовий показник як добуток ймовірності та впливу, після чого масив впорядковується за спаданням отриманих значень. Далі за допомогою методу згортання (reduce) обчислюється загальна сума всіх балів. На фінальній стадії ініціалізується акумулятор з початковим нульовим значенням, після чого здійснюється ітерація відсортованим масивом: на кожному кроці поточний бал додається до накопиченої суми, а кумулятивний відсоток розраховується як відношення цієї суми до загального показника. Отримане значення зберігається як окрема властивість кожного об’єкта ризику.

Функціонал аналізу витрат і вигод реалізовано шляхом створення інтегрованого розрахункового середовища для кожного ризику в межах циклу трансформації. Алгоритм економічного обґрунтування стратегій реагування базується на безперервному циклічному аналізі реєстру загроз. Внутрішня логіка розгалуження, представлена на рисунку 3.6, передбачає динамічну перевірку наявності заходів пом’якшення для кожного ризику. За умови їхньої наявності система виконує перерахунок залишкових збитків, визначає чисту вигоду та показник рентабельності інвестицій (ROI), результатом чого є встановлення булевого індикатора ефективності (isEffective).

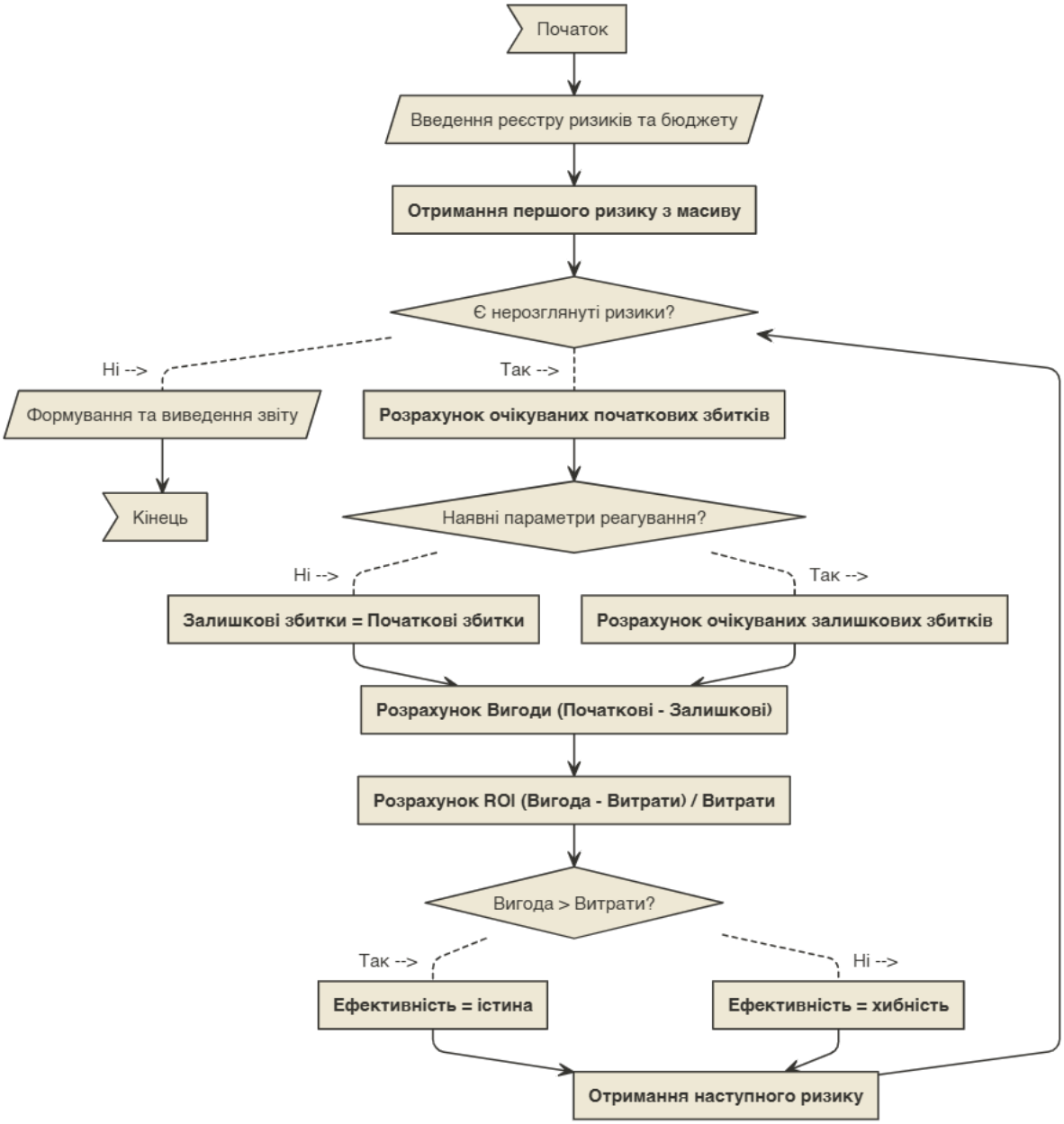


Рисунок 3.6 — Блок-схема алгоритму аналізу витрат і вигод (Cost-Benefit)

Процес розрахунку розпочинається з конвертації балів у ймовірнісні показники з використанням відповідних словників даних. На етапі визначення очікуваної суми первинних втрат алгоритм обчислює математичне сподівання: якщо користувачем встановлено мінімальні та максимальні межі витрат, застосовується середнє арифметичне; за відсутності зазначених параметрів розрахунок базується на пропорційній частці загального бюджету. Отримане

значення первинних втрат коригується на коефіцієнт імовірності їх реалізації.

Наступний етап передбачає валідацію параметрів залишкового ризику. У разі їх наявності процедура обчислення математичного сподівання повторюється для залишкового стану; в іншому випадку залишкові збитки прирівнюються до первинних. Економічний ефект визначається як різниця між очікуваними первинними та залишковими втратами. Показник рентабельності інвестицій (ROI) розраховується за класичною формулою співвідношення чистого прибутку до витрат із попередньою перевіркою коректності операції ділення. Завершальним кроком ітерації є формування булевого індикатора ефективності, який набуває значення «істина» лише за умови, що розрахована вигода перевищує вартість заходів із мінімізації ризику.

Програмна реалізація порівняльного аналізу методів оцінювання ризиків передбачає обчислення коефіцієнта рангової кореляції Спірмена. Алгоритм розпочинається з ініціалізації масиву загроз та створення двох його копій, що впорядковуються за спаданням: перша — відповідно до оцінок матриці ризиків, друга — згідно з пріоритетним числом ризику (RPN) методу аналізу видів та наслідків відмов (FMEA).

Для забезпечення коректності обробки ідентичних оцінок застосовується механізм дробового ранжування. Процедура ідентифікує групи ризиків з однаковими показниками, обчислює середнє арифметичне їхніх позицій та фіксує отримані ранги у відповідних словниках. Наступним етапом алгоритм здійснює ітерацію за вихідним масивом, обчислює квадрат різниці між рангами, присвоєними кожним методом, і накопичує суму цих значень.

На завершальному етапі програма розраховує підсумковий коефіцієнт кореляції, використовуючи накопичену суму квадратів різниць та загальну кількість ризиків. Результатом виконання алгоритму є фінальний показник кореляції, що супроводжується масивом деталізованих проміжних даних.

4 РОБОТА КОРИСТУВАЧА З ВЕБСИСТЕМОЮ

Рекомендованим методом розгортання є використання Docker. У терміналі кореневої директорії проєкту виконайте команду docker-compose up -d --build. Після успішного запуску контейнерів необхідно застосувати міграції бази даних за допомогою команди npx prisma migrate deploy та ініціалізувати тестові дані через npx prisma db seed. Веб-інтерфейс буде доступний за адресою http://localhost:3002.

4.1 Технічні умови та порядок інсталяції

Для стабільної роботи системи «VIKTORIA» необхідно забезпечити функціонування в середовищі ОС Windows, macOS або Linux з мінімальним обсягом оперативної пам'яті 4 ГБ та відкритими мережевими портами 3001, 3002 і 5434. Процес розгортання передбачає наявність інстальованого Docker Desktop або Node.js (версії 20+) та бази даних PostgreSQL 15.

У разі локального розгортання без використання Docker, необхідно самостійно створити базу даних risk_db та налаштувати параметр DATABASE_URL у файлі backend/.env. Подальша конфігурація потребує ручного встановлення програмних залежностей, виконання міграцій бази даних, а також послідовного запуску серверного та клієнтського компонентів у відповідних директоріях.

4.2 Сценарії роботи користувача та аналіз оцінок

Взаємодія з вебсистемою розпочинається з автентифікації користувача на сторінці входу, що надає доступ до персонального портфеля проєктів (рисунок 4.1).

Цей етап забезпечує безпеку та логічну ізоляцію даних, гарантуючи конфіденційність роботи кожного користувача з його власною базою об'єктів. Процес авторизації реалізовано з використанням захищених токенів доступу, які підтверджують правомірність виконання подальших операцій у системі.

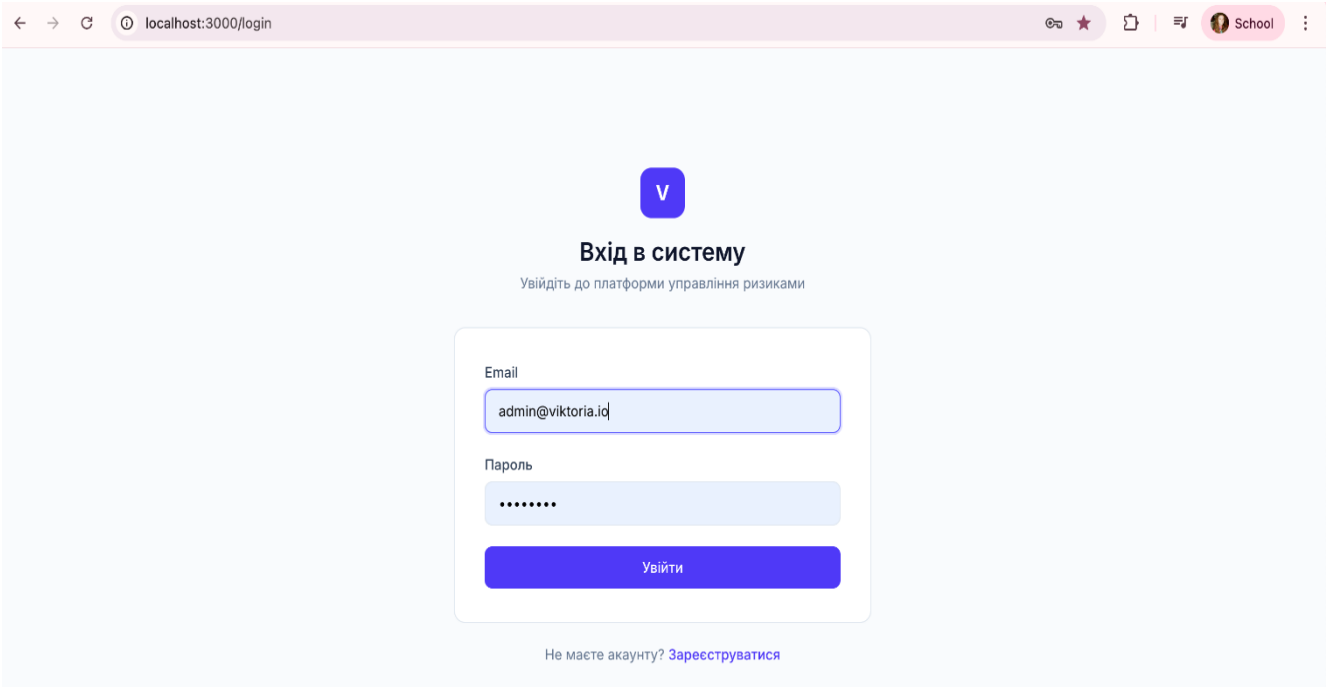


Рисунок 4.1 — Вікно автентифікації користувача

Після успішної авторизації менеджер переходить до панелі управління портфелем, де ініціалізується новий об'єкт дослідження. За допомогою спеціалізованого діалогового вікна створюється проєкт, наприклад, «Реконструкція вузлової електричної підстанції 110 кВ» із загальним бюджетом 25 млн грн (рисунок 4.2). Встановлений бюджет є ключовим параметром, що слугує базовим орієнтиром для розрахунку всіх абсолютних фінансових показників у модулях кількісного аналізу.

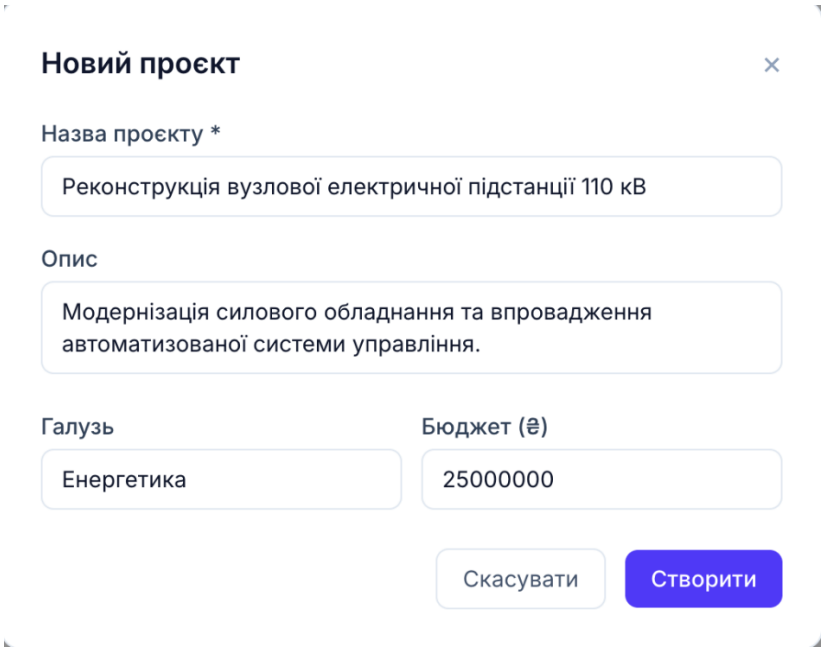


Рисунок 4.2 — Ініціалізація нового проєкту та введення базових параметрів

Наступний етап реалізації проєкту передбачає наповнення бази даних інформацією про потенційні загрози. Використовуючи інтерфейс створення нового ризику, користувач вносить детальні параметри кожної події (рисунки 4.3, 4.4). Окрім базових ідентифікаційних атрибутів, таких як назва, опис та категорія, для кожної загрози визначається комплекс дискретних експертних оцінок за п’ятибальною шкалою, зокрема: рівні ймовірності, впливу та складності виявлення. Додатково, для забезпечення коректної роботи алгоритмів імітаційного моделювання та економічного аналізу, задаються діапазони очікуваних фінансових втрат, часових затримок (у днях), а також плановий бюджет на заходи реагування, включаючи показники залишкових ризиків після їх впровадження.

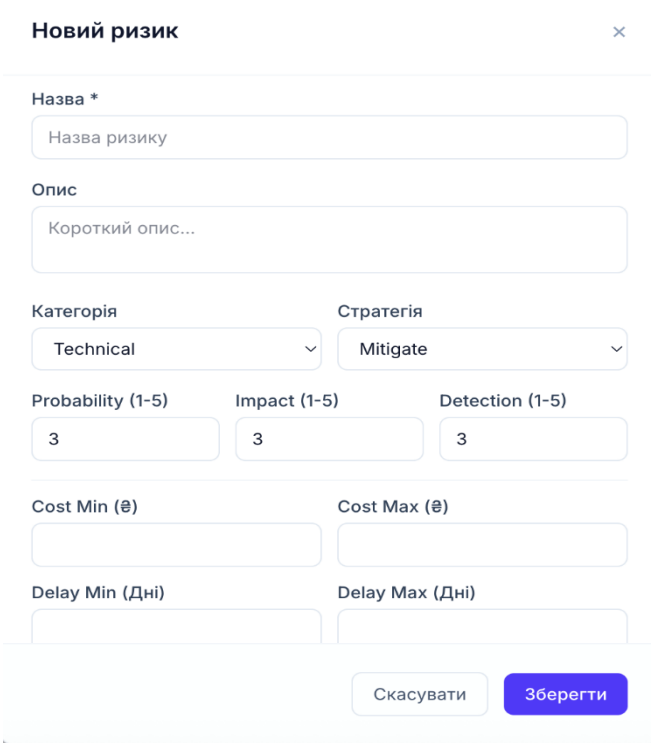


Рисунок 4.3 — Форма додавання нового ризику з кількісними параметрами (частина 1)

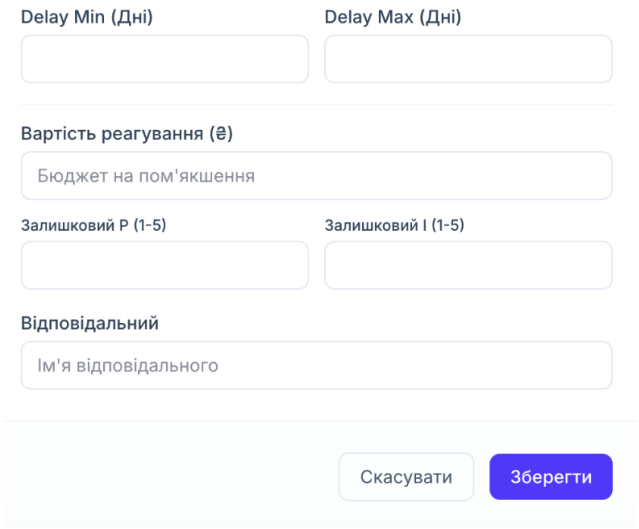


Рисунок 4.4 — Форма додавання нового ризику з кількісними параметрами (частина 2)

Після обробки вхідних даних система генерує зведений реєстр ризиків

проєкту. Даний інтерфейс містить структурований перелік ідентифікованих загроз із відображенням розрахункових показників, обраних стратегій реагування (зокрема, уникнення або мінімізації наслідків), поточних статусів та визначених відповідальних осіб (рисунок 4.5). Реєстр функціонує як централізований інструмент для управління ризиковими записами.

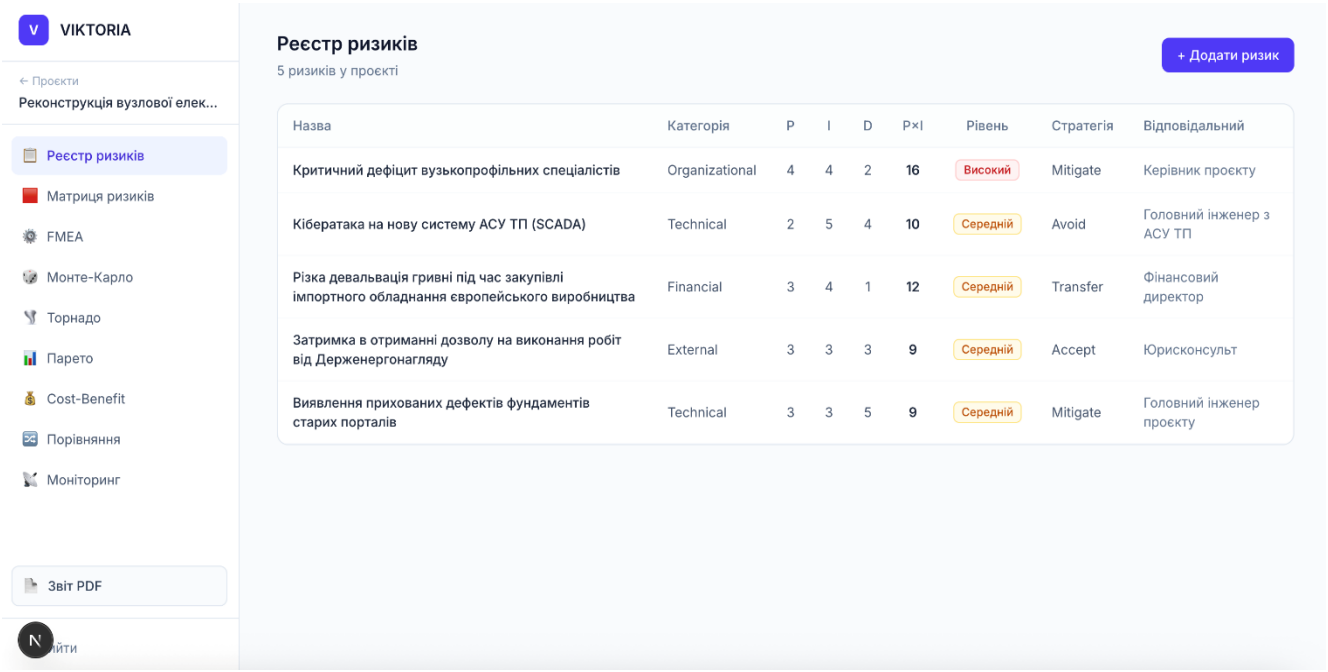


Рисунок 4.5 — Загальний вигляд реєстру ризиків проєкту

Первинна комплексна оцінка загроз відображається за допомогою матриці ймовірності та впливу. Інтерактивна діаграма розсіювання автоматично розраховує добуток цих показників, розподіляючи виявлені ризики за кольоровими зонами критичності — від низького до екстремального рівня (рисунок 4.6). Зокрема, на матриці ідентифіковано ризик критичного дефіциту вузькопрофільних фахівців з оцінкою 16 балів, що відповідає червоній зоні високої небезпеки та вимагає негайного розроблення плану заходів із реагування.

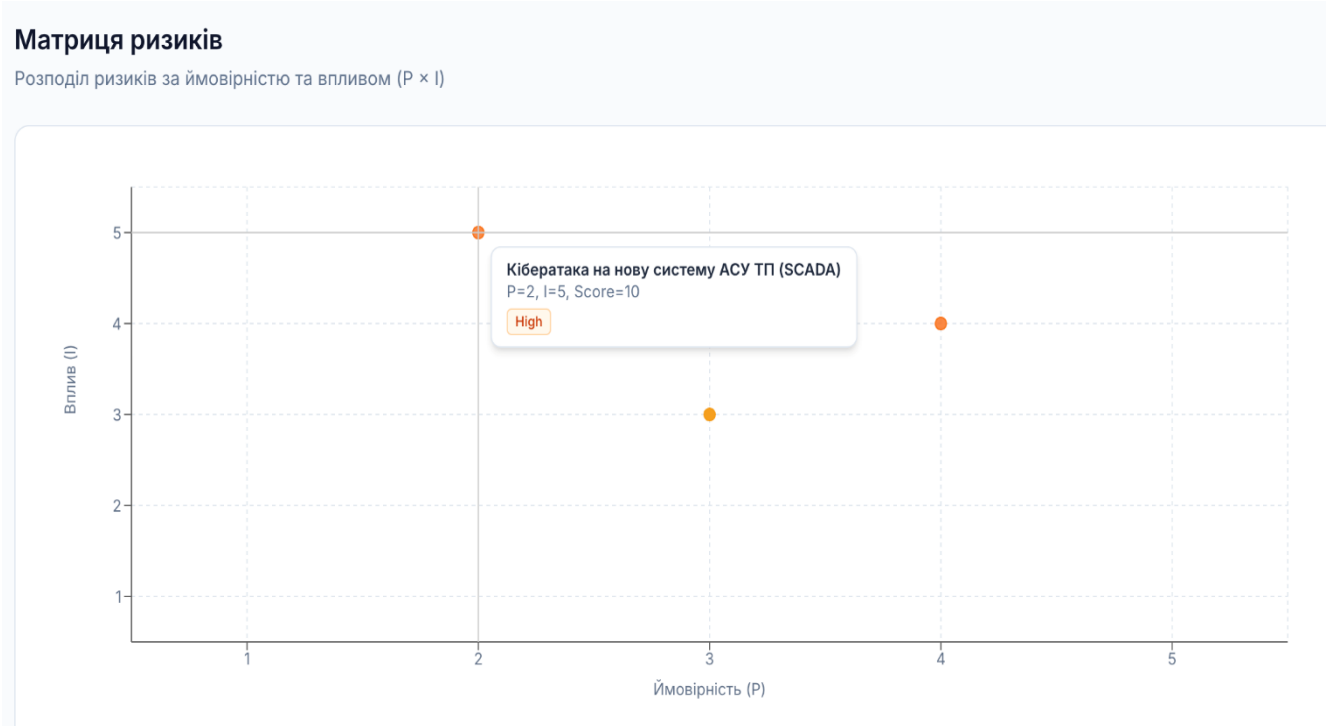


Рисунок 4.6 — Візуалізація розподілу загроз на матриці ризиків

Поглиблений якісний аналіз реалізовано в межах модуля оцінки видів та наслідків відмов, результатом якого є формування ранжованого списку за показником пріоритетного числа ризику (рисунок 4.7). Інтеграція третього виміру — фактора виявлюваності — дозволяє суттєво скоригувати оцінку пріоритетності загроз. Згідно з алгоритмом, до найбільш критичних віднесено виявлення прихованих дефектів фундаментів (показник 45) та кібератаки на системи управління (показник 40). Попри середні значення за матрицею ризиків, вказані загрози становлять найвищу небезпеку через низьку ймовірність їх ідентифікації до моменту виникнення інциденту.

Зони: Low Medium High Extreme

Ризик	P	I	P×I	Зона
Критичний дефіцит вузькопрофільних спеціалістів	4	4	16	High
Кібератака на нову систему АСУ ТП (SCADA)	2	5	10	High
Різка девальвація гривні під час закупівлі імпортного обладнання європейського виробництва	3	4	12	High
Затримка в отриманні дозволу на виконання робіт від Держенергонагляду	3	3	9	Medium
Виявлення прихованих дефектів фундаментів старих порталів	3	3	9	Medium

Рисунок 4.7 — Результати ранжування ризиків за методикою аналізу видів та наслідків відмов

Перехід до етапу кількісного аналізу передбачає застосування методу Монте-Карло для математичного моделювання стохастичних процесів. Обчислювальне ядро виконує 10 000 незалежних ітерацій випадкових сценаріїв з метою прогнозування спектра потенційних фінансових та часових втрат проєкту. Результати моделювання відображаються у вигляді розгорнутої статистики та детальної гістограми розподілу збитків (рисунок 4.8).

Рисунок 4.8 — Гістограма розподілу збитків за результатами симуляції Монте-Карло

Згідно з отриманими даними, середній розмір очікуваних збитків становить

приблизно 6,59 млн грн, що еквівалентно 26 % від загального бюджету. Показник вартості під ризиком (VaR) визначено на рівні 13,44 млн грн. Дана метрика є фундаментальною для прийняття управлінських рішень та вказує на необхідність формування резервного фонду в зазначеному обсязі для забезпечення покриття збитків із рівнем довіри 95 %. Крім того, система прогнозує середнє відхилення термінів реалізації проєкту на 62 дні.

Аналіз чутливості проєкту до впливу окремих факторів наочно ілюструється за допомогою діаграми «торнадо». Цей інструментарій не враховує імовірнісні характеристики подій, а здійснює ранжування ризиків виключно за масштабом їхнього потенційного фінансового впливу на бюджет у межах реалізації найгіршого сценарію, формуючи характерний горизонтальний каскадний графік (рисунок 4.9).

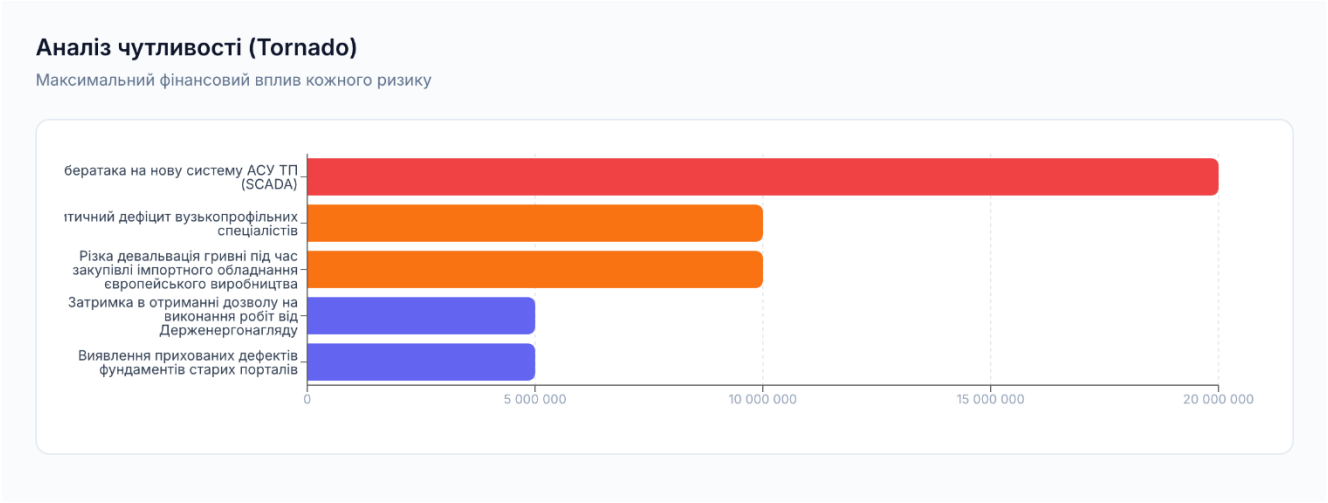


Рисунок 4.9 — Діаграма торнадо для оцінки чутливості проєкту до фінансових втрат

Застосування методу економічного аналізу Парето забезпечує математично обґрунтовану ідентифікацію критичної групи ризиків. Побудовані кумулятивна крива та гістограма дозволяють автоматично визначити загрози, сукупний вплив яких перевищує порогове значення у 80% (рисунок 4.10). У межах цього проєкту чотири ідентифіковані чинники формують критичну масу, що надає менеджменту

змогу оптимізувати управлінський фокус та раціонально розподілити ресурси, уникнувши розпорошення уваги на несуттєві фактори.

Рисунок 4.10 — Діаграма Парето та визначення критичної групи ризиків

Економічна доцільність інвестицій у заходи із забезпечення безпеки визначається за допомогою модуля аналізу «витрати — вигоди». Система враховує потенційні первинні збитки та порівнює їх із залишковими показниками після впровадження превентивних заходів. Програмний комплекс автоматично агрегує вартість стратегій реагування (870 тис. грн) та обчислює очікувану економічну вигоду, що перевищує 4,4 млн грн (рисунок 4.11). Завдяки розрахунку індивідуального показника окупності інвестицій (ROI) для кожного виду ризику система ідентифікує стратегії з високою ефективністю, а також визначає випадки, у яких прийняття ризику є доцільнішим за впровадження додаткових заходів мінімізації.

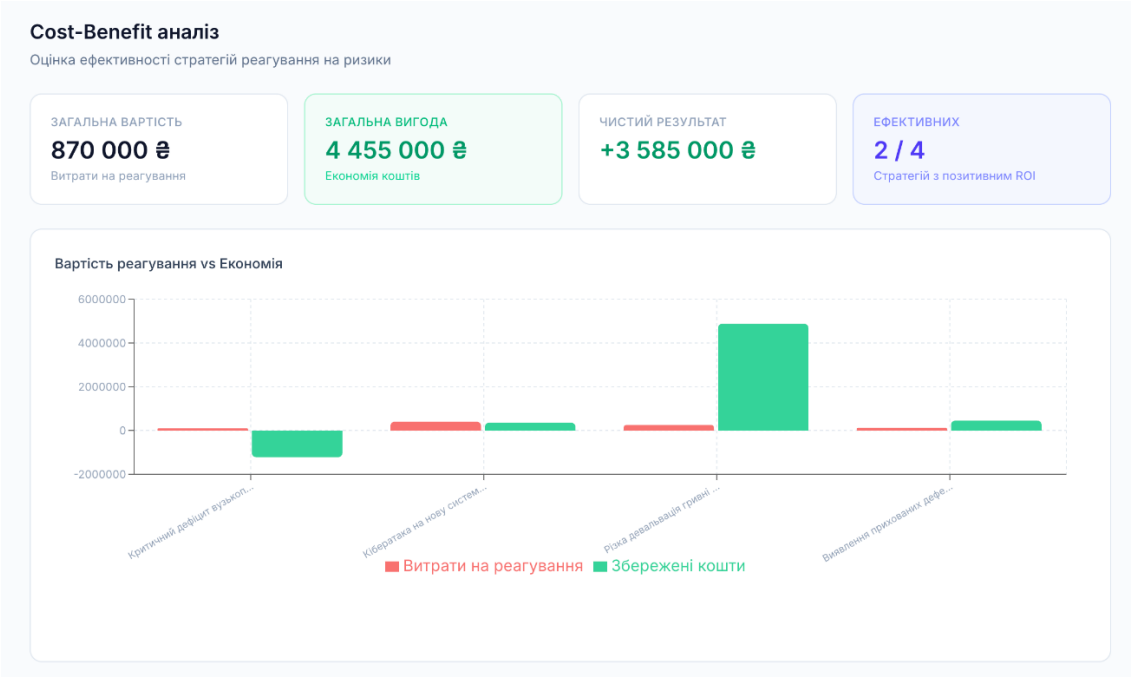


Рисунок 4.11 — Порівняльний аналіз окупності стратегій реагування

Процес валідації та перевірки достовірності результатів завершується автоматичним розрахунком коефіцієнта рангової кореляції Спірмена (рисунок 4.12). Метою цього етапу є оцінка узгодженості рейтингів, отриманих за допомогою класичної матриці та методу FMEA. Система відображає розрахункове значення коефіцієнта на рівні -0,3250. Встановлений показник слабкої оберненої кореляції свідчить про суттєві розбіжності в результатах обох методів, що зумовлено вагомим впливом фактора виявлюваності на визначення пріоритетів. Це підтверджує наукову доцільність та обґрунтованість інтегрованого використання зазначених підходів.

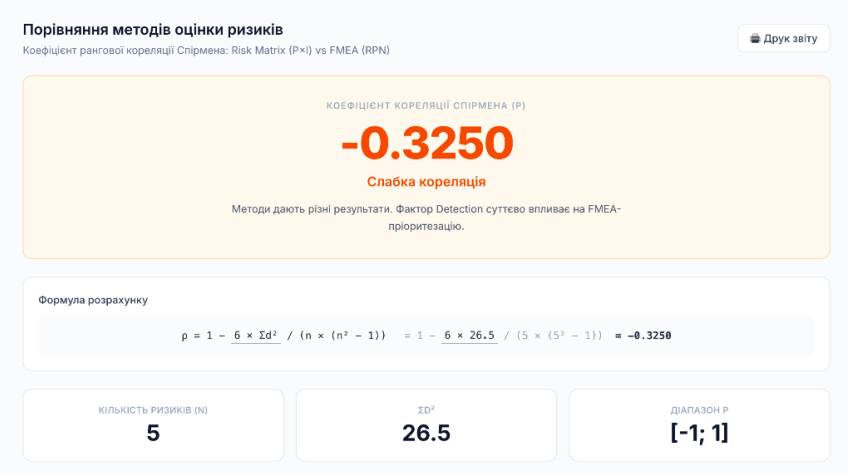


Рисунок 4.12 — Результати розрахунку коефіцієнта рангової кореляції Спірмена

Для забезпечення прозорості управління та цілісності даних платформа оснащена інтегрованою системою аудиту. Підсистема моніторингу автоматично реєструє всі дії користувачів із фіксацією точних часових міток: від створення нових записів до зміни статусів або перегляду бюджету на реагування (рисунок 4.13). Це забезпечує повний контроль над життєвим циклом ризику та унеможливлює внесення несанкціонованих змін без збереження відповідної історії операцій.

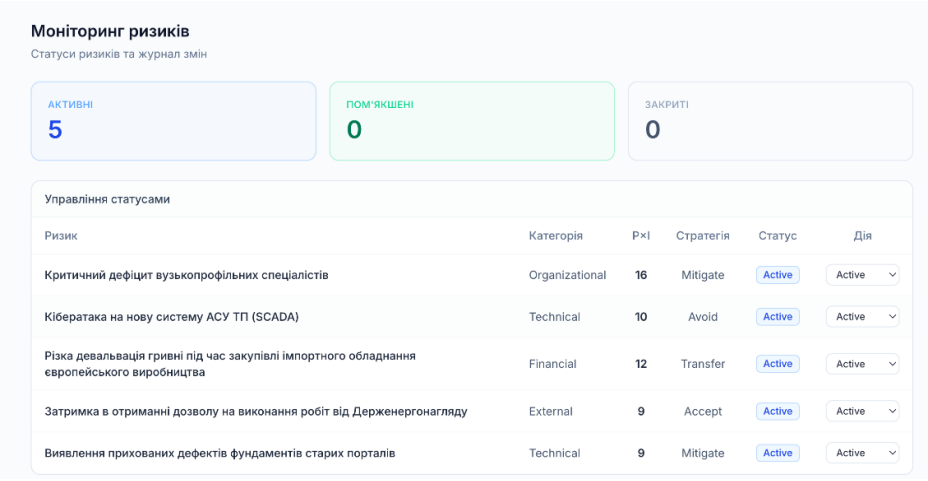
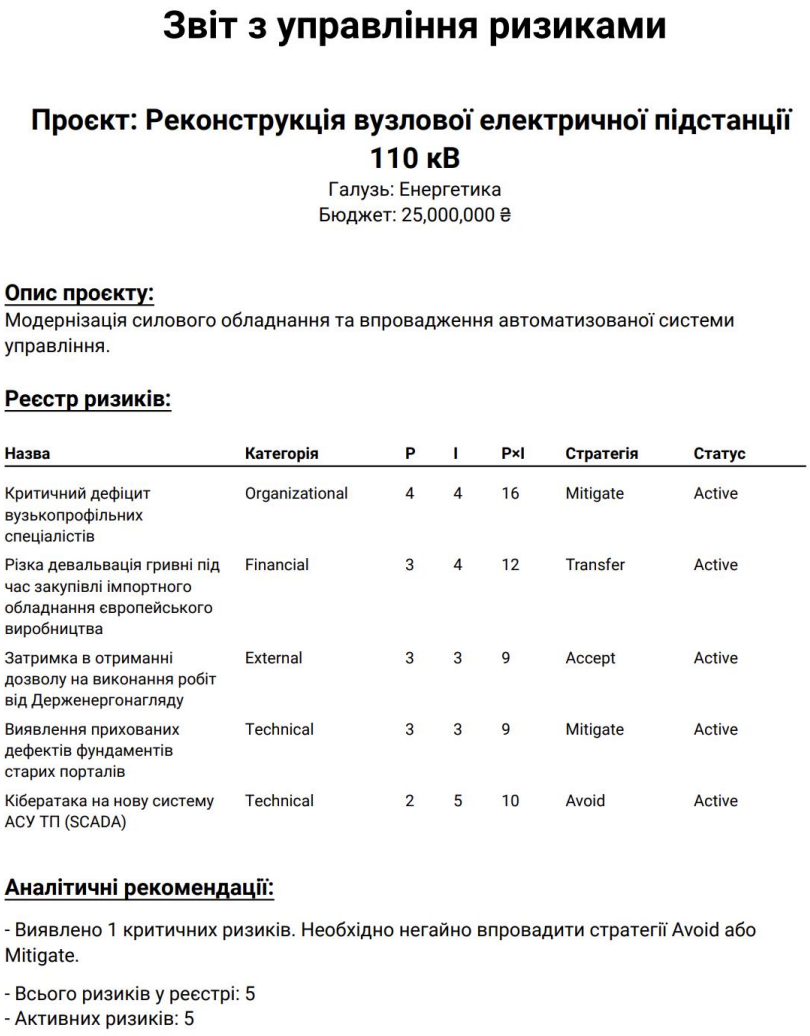


Рисунок 4.13 — Журнал аудиту та панель моніторингу статусів ризиків

Практичний етап експлуатації системи завершується формуванням

комплексного фінального звіту про стан управління ризиками. На запит користувача серверне ядро забезпечує миттєву генерацію документа у форматі PDF із повною підтримкою кириличних шрифтів (рисунок 4.14). Звіт консолідує загальні дані про проєкт, деталізований реєстр ризиків у табличній формі та автоматично сформовані аналітичні рекомендації щодо стратегій реагування на критичні загрози. Документ є повністю готовим до друку та представлення стейкголдерам чи інвесторам проєкту.



Звіт з управління ризиками

Проєкт: Реконструкція вузлової електричної підстанції 110 кВ

Галузь: Енергетика
Бюджет: 25,000,000 ₴

Опис проєкту:
Модернізація силового обладнання та впровадження автоматизованої системи управління.

Реєстр ризиків:

Назва	Категорія	P	I	P×I	Стратегія	Статус
Критичний дефіцит вузькопрофільних спеціалістів	Organizational	4	4	16	Mitigate	Active
Різка девальвація гривні під час закупівлі імпортного обладнання європейського виробництва	Financial	3	4	12	Transfer	Active
Затримка в отриманні дозволу на виконання робіт від Держенергонагляду	External	3	3	9	Accept	Active
Виявлення прихованих дефектів фундаментів старих порталів	Technical	3	3	9	Mitigate	Active
Кібератака на нову систему АСУ ТП (SCADA)	Technical	2	5	10	Avoid	Active

Аналітичні рекомендації:

- Виявлено 1 критичних ризиків. Необхідно негайно впровадити стратегії Avoid або Mitigate.

- Всього ризиків у реєстрі: 5
- Активних ризиків: 5

Рисунок 4.14 — Фрагмент згенерованого звіту у форматі PDF

Результати практичної експлуатації вебсистеми «VIKTORIA»

підтверджують її повну відповідність функціональним вимогам та ефективність у забезпеченні наскрізної автоматизації процесів ризик-менеджменту. Впроваджені алгоритми демонструють високу продуктивність обчислювального ядра, здатного виконувати багатовимірний аналіз та масштабне стохастичне моделювання в режимі реального часу.

Інтеграція різнорідних методологій — від матриць ризиків до імітаційного моделювання Монте-Карло та аналізу «витрати-вигоди» (Cost-Benefit) — у межах єдиного цифрового середовища дозволяє відмовитися від використання несистематизованих електронних таблиць та ручних розрахунків. Формування точних метрик, візуалізація даних та автоматизована підготовка аналітичних звітів забезпечують надійне обґрунтування для прийняття стратегічних управлінських рішень. Таким чином, система надає керівникам проєктів, аналітикам та інвесторам ефективний інструментарій для мінімізації фінансових і часових ризиків, що сприяє успішній реалізації комплексних проєктів, зокрема у високотехнологічних та енергетичних секторах.

ВИСНОВКИ

Під час виконання дипломної роботи було розроблено вебсистему аналізу й управління ризиками на основі кількісних методів із застосуванням сучасних вебтехнологій та імовірнісних алгоритмів для автоматизації процесів прогнозування фінансових і часових втрат проєкту.

Отримано наступні результати:

1. За результатами аналізу підходів, методів та алгоритмів розв’язання поставленої задачі обґрунтовано доцільність застосування матриці ризиків, методології FMEA, імітаційного моделювання за методом Монте-Карло, аналізу Парето, діаграми Торнадо, аналізу «витрати-вигоди» (Cost-Benefit), а також методології рангової кореляції Спірмена.

2. З огляду на технічний аналіз програмного забезпечення, обґрунтовано вибір мови програмування TypeScript, модульного серверного фреймворку NestJS, клієнтського фреймворку Next.js, реляційної системи управління базами даних PostgreSQL із застосуванням інструментарію Prisma ORM, а також технології контейнеризації Docker для розгортання системи.

3. Розроблено програмний комплекс «VIKTORIA» — вебсистему для аналізу та управління ризиками з використанням кількісних методів. Функціонал системи забезпечує автоматизацію математичних розрахунків фінансових та часових втрат, формування реєстру загроз, інтерактивну візуалізацію аналітичних показників та автоматичне генерування звітності у форматі PDF.

4. Коректність функціонування розробленого програмного забезпечення підтверджено результатами тестування. Верифіковано точність роботи математичного ядра при виконанні стохастичних симуляцій, підтверджено надійність механізмів автентифікації та аудиту бази даних, а також перевірено стабільність клієнт-серверної взаємодії за різних рівнів навантаження.

Подальший розвиток системи передбачає розширення спектра

підтримуваних імовірнісних розподілів (зокрема нормального та трикутного) для методу Монте-Карло, впровадження алгоритмів машинного навчання для автоматичної ідентифікації типових ризиків на основі історичних даних, а також розробку публічного API для інтеграції з корпоративними системами управління проєктами (Jira, MS Project).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Настанова до зводу знань з управління проектами (A Guide to the Project Management Body of Knowledge). 7-ме вид. Project Management Institute, 2021. 370 с.

2. ДСТУ ISO 31000:2018 (ISO 31000:2018, IDT). Керування ризиками. Настанови. [Чинний від 2019-01-01]. Київ : Держспоживстандарт України, 2018. 21 с..

3. Клейн Д. Побудова систем на NestJS : масштабовані Node.js застосунки. О'Рейлі, 2023. 280 с.

4. Чинна Т., Дакетт Д. Розробка на React та Next.js : сучасний фронтенд. Сан-Франциско : No Starch Press, 2024. 350 с.

5. Вітвер Н. Методи Монте-Карло в управлінні ризиками та фінансах. 2-ге вид. Нью-Йорк : Wiley, 2022. 420 с..

6. Жигірь А. А. Методи кількісної оцінки ризику економічного розвитку підприємства. Ефективна економіка. 2014. № 10. URL: http://www.economy.nayka.com.ua/?op=1&z=3413.

7. Степанюк З. А. Огляд сучасних методів кількісної оцінки ризиків та впровадження інформаційних технологій в систему ризик-менеджменту. Економіка. Менеджмент. Бізнес. 2017. № 4 (22). С. 176–182.

8. Грицай О. І., Дефір І. В., Козак О. Є. Огляд кількісних методів оцінки ризиків зовнішньоекономічної діяльності. Економіка та суспільство. 2024. Вип. 68. DOI: https://doi.org/10.32782/2524-0072/2024-68-104.

9. Вітлінський В. В., Верченко П. І. Аналіз, моделювання та управління економічним ризиком : навч.-метод. посіб. Київ : КНЕУ, 2000. 292 с..

10. Слободянюк О., Кібік О., Котлубай В. Ризикологія : навч.-метод. посіб. Одеса : Фенікс, 2025. 78 с..

11. Технології розробки програмного забезпечення. Практикум : навч. посіб. / В. О. Тихоход та ін. Київ : КПІ ім. Ігоря Сікорського, 2024. 139 с.

12. Волівач А. П., Хімічева Г. І. Ідентифікація ризиків навчального процесу ЗВО за вимогами міжнародного стандарту ДСТУ ISO 31010:2013 : thesis. 2019. URL: https://er.knutd.edu.ua/handle/123456789/14637 (дата звернення: 19.05.2026).

13. Томашевський В. М. Моделювання систем : підруч. для студентів ВНЗ. Київ : ВНV, 2005. 352 с.

14. Слюсарчук П. В. Теорія ймовірностей та математична статистика : підруч. для студ. вищ. навч. закл. Ужгород : Карпати, 2005. 184 с.

15. Тарасюк Г. М. Бізнес-план: розробка, обгрунтування та аналіз : навч. посіб. для студентів ВНЗ. Київ : Каравела, 2008. 280 с.

16. Лавріщева К. М. Розвиток ідей академіка В.М. Глушкова з питань технології програмування. Вісник Національної академії наук України. 2013. № 9. С. 66–83.

17. Фрімен А. TypeScript для професіоналів. Київ : Діалектика, 2021. 624 с.

18. Іванюта Т. М., Заїчкова Л. О. Економічний аналіз та оцінка інвестиційних проєктів : навч. посіб. Київ : Центр навчальної літератури, 2019. 256 с.

19. ДСТУ IEC/ISO 31010:2013 (IEC/ISO 31010:2009, IDT). Керування ризиком. Методи загального оцінювання ризику. [Чинний від 2015-01-01]. Київ : Мінекономрозвитку України, 2015. 96 с.

20. Єріна А. М. Статистичне моделювання та прогнозування. Київ : КНЕУ, 2014. 348 с.

ДОДАТОК А

Програмна реалізація обчислювального ядра системи кількісного та якісного аналізу ризиків

Технологічний стек:

– мова програмування: TypeScript;

– серверний фреймворк для розробки модульної архітектури: NestJS;

– інструментарій об’єктно-реляційного відображення (ORM): Prisma;

– засоби імітаційного моделювання та статистичних обчислень (метод Монте-Карло, коефіцієнт кореляції Спірмена): стандартні математичні модулі TypeScript (Math).

```
import { Injectable, NotFoundException } from '@nestjs/common';
import { PrismaService } from '../prisma/prisma.service';
import { ProjectsService } from '../projects/projects.service';

@Injectable()
export class AnalysisService {
  constructor(
    private prisma: PrismaService,
    private projectsService: ProjectsService
  ) {}

  // 1. Risk Matrix
  async getMatrix(userId: number, projectId: number) {
    await this.projectsService.findOne(userId, projectId);
    const risks = await this.prisma.risk.findMany({ where: { projectId } });
```

```
  return risks.map(risk => {
    const score = risk.probability * risk.impact;
    let zone = 'Low';
    if (score >= 5 && score <= 9) zone = 'Medium';
    if (score >= 10 && score <= 16) zone = 'High';
    if (score >= 17) zone = 'Extreme';

    return {
      id: risk.id,
      title: risk.title,
      probability: risk.probability,
      impact: risk.impact,
      score,
      zone
    };
  });
}

// 2. FMEA
async getFMEA(userId: number, projectId: number) {
  await this.projectsService.findOne(userId, projectId);
  const risks = await this.prisma.risk.findMany({ where: { projectId } });

  const fmea = risks.map(risk => ({
    id: risk.id,
    title: risk.title,
    probability: risk.probability,
    impact: risk.impact,
```

```
      detection: risk.detection,
      rpn: risk.probability * risk.impact * risk.detection
    }));

    return fmea.sort((a, b) => b.rpn - a.rpn);
  }

  // 3. Monte Carlo
  async getMonteCarlo(userId: number, projectId: number) {
    const project = await this.projectsService.findOne(userId, projectId);
    if (!project.budget) {
      throw new NotFoundException('Для Монте-Карло потрібен бюджет проєкту');
    }

    const risks = await this.prisma.risk.findMany({ where: { projectId } });

    // Mapping 1-5 to real percentages
    const probMap = { 1: 0.1, 2: 0.3, 3: 0.5, 4: 0.7, 5: 0.9 };
    const impactMap = { 1: 0.05, 2: 0.1, 3: 0.2, 4: 0.4, 5: 0.8 };

    const iterations = 10000;
    const lossResults: number[] = [];
    const delayResults: number[] = [];

    for (let i = 0; i < iterations; i++) {
      let totalLoss = 0;
      let totalDelay = 0;
      for (const risk of risks) {
```

```
            const prob = probMap[risk.probability];

            // Randomly check if risk occurs
            if (Math.random() <= prob) {
              // Calculate financial loss using Uniform distribution
              if (risk.costMin != null && risk.costMax != null) {
                totalLoss += risk.costMin + Math.random() * (risk.costMax - risk.costMin);
              } else {
                // Fallback to static mapping if specific costs aren't set
                const impactPerc = impactMap[risk.impact];
                totalLoss += project.budget * impactPerc;
              }

              // Calculate time delay using Uniform distribution
              if (risk.delayMin != null && risk.delayMax != null) {
                totalDelay += risk.delayMin + Math.random() * (risk.delayMax - risk.delayMin);
              }
            }
          }
          lossResults.push(totalLoss);
          delayResults.push(totalDelay);
        }

        // Sort to find VaR (Value at Risk) 95%
        lossResults.sort((a, b) => a - b);
        const varIndex = Math.floor(iterations * 0.95);
        const var95 = lossResults[varIndex];
```

```
    const avgLoss = lossResults.reduce((a, b) => a + b, 0) / iterations;

    // Time stats
    delayResults.sort((a, b) => a - b);
    const avgDelay = delayResults.reduce((a, b) => a + b, 0) / iterations;
    const maxDelay = delayResults[iterations - 1] || 0;

    // Optional: Return a histogram instead of 1000 raw values for better frontend rendering
    const histogram = this.buildHistogram(lossResults, 20);

    return {
      iterations,
      budget: project.budget,
      avgLoss,
      var95,
      avgDelay,
      maxDelay,
      histogram,
    };
  }

  private buildHistogram(data: number[], bins: number) {
    if (data.length === 0) return [];
    const min = data[0];
    const max = data[data.length - 1];
    if (min === max) return [{ binStart: min, binEnd: max, count: data.length }];

    const binSize = (max - min) / bins;
```

```
  const histogram = Array.from({ length: bins }, (_, i) => ({
    binStart: min + i * binSize,
    binEnd: min + (i + 1) * binSize,
    count: 0
  }));

  for (const val of data) {
    const binIndex = Math.min(Math.floor((val - min) / binSize), bins - 1);
    histogram[binIndex].count++;
  }

  return histogram;
}

// 4. Tornado (Sensitivity)
async getTornado(userId: number, projectId: number) {
  const project = await this.projectsService.findOne(userId, projectId);
  const risks = await this.prisma.risk.findMany({ where: { projectId } });
  const budget = project.budget || 100000; // fallback if no budget
  const impactMap = { 1: 0.05, 2: 0.1, 3: 0.2, 4: 0.4, 5: 0.8 };

  const sensitivity = risks.map(risk => ({
    id: risk.id,
    title: risk.title,
    maxFinancialImpact: budget * impactMap[risk.impact]
  }));

  return sensitivity.sort((a, b) => b.maxFinancialImpact - a.maxFinancialImpact);
}
```

```
// 5. Pareto
async getPareto(userId: number, projectId: number) {
  await this.projectsService.findOne(userId, projectId);
  const risks = await this.prisma.risk.findMany({ where: { projectId } });

  // Calculate score and sort
  const data = risks.map(risk => ({
    id: risk.id,
    title: risk.title,
    score: risk.probability * risk.impact
  })).sort((a, b) => b.score - a.score);

  const totalScore = data.reduce((acc, curr) => acc + curr.score, 0);
  let cumulativeScore = 0;

  return data.map(risk => {
    cumulativeScore += risk.score;
    return {
      ...risk,
      cumulativePercentage: (cumulativeScore / totalScore) * 100
    };
  });
}

// 6. Cost-Benefit Analysis
async getCostBenefit(userId: number, projectId: number) {
  const project = await this.projectsService.findOne(userId, projectId);
  const risks = await this.prisma.risk.findMany({ where: { projectId } });
```

```
const probMap = { 1: 0.1, 2: 0.3, 3: 0.5, 4: 0.7, 5: 0.9 };
const impactMap = { 1: 0.05, 2: 0.1, 3: 0.2, 4: 0.4, 5: 0.8 };
const budget = project.budget || 100000;

const results = risks.map(risk => {
 // 1. Initial Expected Loss
 const prob = probMap[risk.probability];
 let initialLoss = 0;
 if (risk.costMin != null && risk.costMax != null) {
  initialLoss = (risk.costMin + risk.costMax) / 2; // Expected value
 } else {
  initialLoss = budget * impactMap[risk.impact];
 }
 const expectedInitialLoss = prob * initialLoss;

 // 2. Residual Expected Loss
 let expectedResidualLoss = expectedInitialLoss; // Default if no mitigation
 if (risk.residualProbability && risk.residualImpact) {
   const resProb = probMap[risk.residualProbability];
   const resLoss = budget * impactMap[risk.residualImpact];
   expectedResidualLoss = resProb * resLoss;
 }

 const mitigationCost = risk.mitigationCost || 0;
 const benefit = expectedInitialLoss - expectedResidualLoss;
 const roi = mitigationCost > 0 ? ((benefit - mitigationCost) / mitigationCost) *
100 : 0;
 const isEffective = benefit > mitigationCost;
```

```
      return {
        id: risk.id,
        title: risk.title,
        expectedInitialLoss,
        expectedResidualLoss,
        mitigationCost,
        benefit,
        roi,
        isEffective
      };
    });

    return results;
  }
}
import { Injectable } from '@nestjs/common';
import { PrismaService } from '../prisma/prisma.service';
import { ProjectsService } from '../projects/projects.service';

@Injectable()
export class ComparisonService {
  constructor(
    private prisma: PrismaService,
    private projectsService: ProjectsService
  ) {}

  async getSpearmanCorrelation(userId: number, projectId: number) {
    await this.projectsService.findOne(userId, projectId);
```

```
const risks = await this.prisma.risk.findMany({ where: { projectId } });

const n = risks.length;
if (n === 0) {
  return { correlation: 0, details: [] };
}

// Sort descending to assign ranks (highest score = rank 1)
const sortedByMatrix = [...risks].sort((a, b) => (b.probability * b.impact) - (a.probability * a.impact));
const sortedByFmea = [...risks].sort((a, b) => (b.probability * b.impact * b.detection) - (a.probability * a.impact * a.detection));

// Assign fractional ranks for tied values
const getRanks = (sortedArray, scoreFn) => {
  const ranks = {};
  let i = 0;
  while (i < sortedArray.length) {
    let j = i;
    while (j < sortedArray.length && scoreFn(sortedArray[i]) === scoreFn(sortedArray[j])) {
      j++;
    }
    const avgRank = (i + 1 + j) / 2; // e.g. ranks 1,2,3 for 3 tied items -> (1+3)/2 = 2
    for (let k = i; k < j; k++) {
      ranks[sortedArray[k].id] = avgRank;
    }
    i = j;
```

```
    }
    return ranks;
  };

  const matrixRanks = getRanks(sortedByMatrix, r => r.probability * r.impact);
  const fmeaRanks = getRanks(sortedByFmea, r => r.probability * r.impact * r.detection);

  let sumDSquared = 0;
  const details = risks.map(risk => {
    const rankMatrix = matrixRanks[risk.id];
    const rankFmea = fmeaRanks[risk.id];
    const d = rankMatrix - rankFmea;
    const dSquared = d * d;
    sumDSquared += dSquared;

    return {
      id: risk.id,
      title: risk.title,
      matrixScore: risk.probability * risk.impact,
      fmeaScore: risk.probability * risk.impact * risk.detection,
      rankMatrix,
      rankFmea,
      d,
      dSquared
    };
  });

  let correlation = 1;
```

```
    if (n > 1) {
      correlation = 1 - (6 * sumDSquared) / (n * (n * n - 1));
    }

    return {
      correlation,
      details: details.sort((a, b) => a.rankMatrix - b.rankMatrix)
    };
  }
}
```

ДОДАТОК Б

Вебсистема аналізу та управління ризиками на основі кількісних методів

АПРОБАЦІЯ

3-тя Міжнародна науково-практична конференція
«Science and Technology: New Horizons of Development»:
збірник наукових праць / International
Scientific Unity. – Прага, Чехія, 3–5 червня 2026 року. –
Секція: Комп'ютерна інженерія

Аркушів 7

Київ — 2026

Конференція – «Science and Technology: New Horizons of Development»
Секція – Комп'ютерна інженерія

ВЕБСИСТЕМА АНАЛІЗУ ТА УПРАВЛІННЯ РИЗИКАМИ НА ОСНОВІ КІЛЬКІСНИХ МЕТОДІВ

Канавець Вікторія
студентка 4 курсу
vika.kanavets@gmail.com
Кафедра цифрових технологій в енергетиці
НТУУ «Київський політехнічний інститут імені Ігоря Сікорського», Україна

Вступ. Сучасні умови провадження інвестиційної, виробничої та проєктної діяльності в енергетичному та високотехнологічному секторах вимагають впровадження інноваційних інструментів управління ризиками. Традиційні методи, зокрема класична матриця ризиків та методологія FMEA, забезпечують

переважно якісний або напівкількісний аналіз, що зумовлює суб'єктивність оцінювання та обмежує точність прогнозування фінансових збитків чи часових відхилень [1, 2]. Водночас використання методів математичного моделювання та імовірнісного аналізу в межах стандартних табличних процесорів є ресурсомістким, що призводить до фрагментарності даних та зниження ефективності аналітичних процесів [6, 7]. З огляду на це, розробка інтегрованого вебсередовища для оперативного розрахунку, пріоритизації та моніторингу ризиків є стратегічно важливим науково-практичним завданням.

Аналіз літературних джерел та проблематики розробки ПЗ. Аналіз ринку спеціалізованого програмного забезпечення засвідчує, що переважна більшість наявних інструментів автоматизації характеризується вузькою галузевою спеціалізацією або суттєвими архітектурними обмеженнями [7, 8]. Зокрема, програмний комплекс «@RISK» від компанії «Lumivero» забезпечує реалізацію стохастичного моделювання, проте функціонує виключно як надбудова до табличного процесора MS Excel, що суттєво обмежує можливості візуалізації першопричин. Система «RiskyProject», поєднуючи кількісні методи з календарним плануванням, орієнтована переважно на лінійні графіки Ганта, що є недостатньо ефективним для сучасних гнучких методологій управління. Платформа «BowTieXP» зосереджена на візуалізації бар'єрів безпеки, залишаючись при цьому ізольованою від імовірнісного фінансового моделювання бюджету. Корпоративні рішення, такі як «Sphera» чи «DNV Synergi Life», є високовартісними, закритими (proprietary) та адаптованими передусім для аналізу вже виниклих інцидентів, а не для проактивного моделювання ризиків.

Основними викликами в розробці програмного забезпечення для сфери ризик-менеджменту є [7]:

1)Відсутність гнучких вебплатформ з відкритим кодом (open-source), здатних інтегрувати структурні якісні методи ідентифікації ризиків із комплексними стохастичними обчисленнями на основі єдиної моделі даних [6, 8].

2)Десктопна ізольованість академічних та інженерних рішень, що обмежує можливості багатокористувацького доступу, гнучкого розмежування прав доступу та впровадження наскрізних систем аудиту змін у режимі реального часу.

Мета та завдання. Метою роботи є проєктування та розробка вебсистеми аналізу й управління ризиками на основі кількісних методів із застосуванням сучасних вебтехнологій та імовірнісних алгоритмів для автоматизації процесів прогнозування фінансових і часових втрат проєкту.

Для досягнення поставленої мети визначено такі завдання:

1) Аналіз ринкових рішень, підходів, методів та алгоритмів кількісної і якісної оцінки ризиків.

2) Обґрунтування вибору технологічного стека для розробки вебсистеми «VIKTORIA».

3) Проєктування та розробка програмного забезпечення для платформи

управління ризиками.

4) Комплексне тестування та верифікація розробленої системи.

Вхідні та вихідні дані. Інформаційна модель вебсистеми побудована на трирівневій ієрархічній структурі, що охоплює облікові записи користувачів, параметри портфеля проєктів та деталізований реєстр ризиків. Вхідний масив даних включає автентифікаційні відомості та ключові проєктні індикатори, де базовим показником для кількісного моделювання є загальний бюджет. Кожен запис у реєстрі ризиків структурований за понад 16 параметрами, зокрема: опис, класифікація, відповідальні особи, оцінка фінансових і часових втрат, витрати на заходи реагування, а також дискретні експертні оцінки ймовірності, впливу та виявлюваності (за п'ятибальною шкалою) [1, 2].

Вихідні дані формуються шляхом застосування семи математико-статистичних методів аналізу, що включають:

1)автоматизовану сегментацію ризиків відповідно до зон критичності;

2)розрахунок очікуваних фінансових втрат та показника вартості під ризиком (VaR) з рівнем довіри 95% за результатами стохастичного моделювання [5];

3)визначення коефіцієнтів рангової кореляції Спірмена для верифікації узгодженості методів оцінювання [9];

4)обчислення коефіцієнтів окупності інвестицій у заходи безпеки (ROSI/ROI) для визначення ефективності стратегій мінімізації ризиків [9].

Фінальним етапом обробки є формування структурованих наборів даних для візуалізації результатів за допомогою інтерактивних діаграм розподілу втрат, кумулятивних кривих Парето та діаграм чутливості Торнадо, а також автоматична генерація аналітичних звітів у форматі PDF.

Архітектура платформи. Вебсистему спроєктовано з використанням розподіленої клієнт-серверної архітектури. Клієнтська частина базується на фреймворку Next.js 16.2 із застосуванням парадигми App Router та технології Server-Side Rendering (SSR), а також бібліотеці React 19.2 [4]. Стилізацію інтерфейсу реалізовано за допомогою Tailwind CSS 4, тоді як візуалізацію інтерактивних графіків забезпечує бібліотека Recharts 3.8.

Серверна частина (Backend API) побудована на базі модульного фреймворку NestJS 11.x із використанням мови TypeScript 5.7 [3]. Архітектура сервера передбачає декомпозицію логіки на ізольовані модулі: AuthModule (автентифікація за допомогою Passport та JWT), ProjectsModule (CRUD-операції з проєктами з урахуванням каскадного видалення), RisksModule (управління реєстром ризиків із вбудованим автоматичним аудитом), а також спеціалізовані модулі аналітичних обчислень. Взаємодію з реляційною СУБД PostgreSQL 15 забезпечує Prisma ORM 7.8 [3].

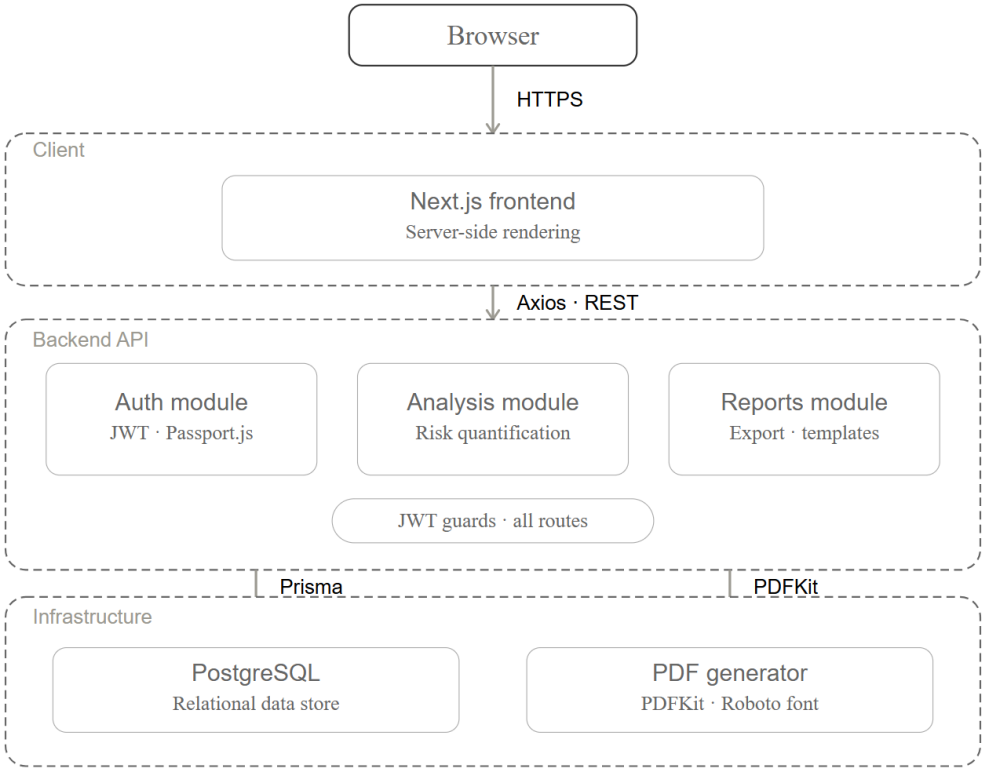


Рисунок 1. Загальна схема архітектури додатку.

Методи якісного аналізу. Первинна оцінка та ранжування загроз у системі здійснюються шляхом застосування методів якісного аналізу, що забезпечує формалізацію експертних висновків.

1. **Кількісна матриця ризиків**. Фундаментальний інструмент для переходу від якісного до кількісного аналізу. Рівень ризику визначається як функція двох незалежних змінних. Розрахунковий бал (Score) обчислюється за такою формулою:

$$Score = Probability \times Impact \quad (1)$$

де Probability — експертна оцінка ймовірності настання ризикової події (за шкалою від 1 до 5), а Impact — експертна оцінка рівня її впливу на проєкт. На підставі отриманого значення ризики розподіляються за чотирма категоріями критичності: Low (низька), Medium (середня), High (висока) та Extreme (екстремальна).

2. **Аналіз видів та наслідків відмов (FMEA)**. З метою усунення обмежень стандартної двовимірної матриці, яка не враховує здатність команди своєчасно ідентифікувати загрозу, було інтегровано показник виявлюваності (Detection). Розрахунок пріоритетного числа ризику (RPN) здійснюється за такою формулою:

$$RPN = Probability \times Impact \times Detection \quad (2)$$

де Detection — це показник здатності ідентифікувати ризик до моменту його реалізації (інвертована шкала: 1 — легка ідентифікація, 5 — прихована загроза).

Використання цього критерію дозволяє ефективно виявляти критичні «сліпі зони» проєкту.

Методи кількісного та статистичного аналізу. Система використовує комплекс кількісних та статистичних методів для врахування стохастичного характеру невизначеності, оптимізації розподілу ресурсів і верифікації отриманих оцінок.

1. **Метод імітаційного моделювання Монте-Карло**. Застосовується для трансформації дискретних експертних оцінок у статистичні розподіли ймовірностей. Обсяг потенційних збитків моделюється як випадкова величина, що підпорядковується рівномірному розподілу (Uniform Distribution). Значення фінансових втрат або часових затримок (X) на кожній ітерації розраховується за наступною формулою:

$$X = C_{min} + R \times (C_{max} - C_{min}) \quad (3)$$

де C_{min} та C_{max} — задані експертами нижня та верхня межі діапазону фінансових втрат (або відхилень у розкладі);

R — випадкове число, згенероване алгоритмом в інтервалі [0; 1].

На основі 10 000 ітерацій розраховується ключова метрика — ризикова вартість (VaR 95%), що визначає максимальний рівень збитків із рівнем довіри 95% [5].

2. **Аналіз «витрати-вигоди» (Cost-Benefit Analysis).** Цей інструмент надає стратегічне обґрунтування доцільності впровадження превентивних заходів. Кількісна оцінка фінансової ефективності реалізується шляхом розрахунку показника рентабельності інвестицій у заходи реагування (ROI) за наступною формулою:

$$ROI = \frac{Benefit - MitigationCost}{MitigationCost} \times 100\% \quad (4)$$

де $Benefit$ — очікувана вигода. Вона розраховується як різниця між початковими (прогнозованими до впровадження заходів) збитками та залишковими збитками (Residual Risk), які залишаться після реалізації стратегії;

$MitigationCost$ — пряма вартість впровадження стратегії реагування (наприклад, купівля ліцензій, наймання додаткових фахівців, страхування).

3. Рангова кореляція Спірмена. Апарат непараметричної статистики, що застосовується для математичної верифікації впливу фактора виявлення (Detection) на пріоритетність ризиків. Коефіцієнт кореляції ρ обчислюється за формулою:

$$p = 1 - \frac{6 \sum d_i^2}{n(n^2-1)} \quad (5)$$

де d_i — різниця між рангами і-го ризику в матричному рейтингу Score та рейтингу RPN;

n — загальна кількість ідентифікованих ризиків у реєстрі проєкту.

4. **Аналіз Парето та діаграма Торнадо.** Для оптимізації фокусу на пріоритетних завданнях у систему інтегровано метод аналізу Парето (принцип

80/20), що забезпечує автоматичне ранжування ризиків за показниками RPN або EMV з розрахунком їхньої кумулятивної частки. Візуалізація даних за допомогою діаграми Парето дозволяє оперативно ідентифікувати критичні зони впливу та формувати ефективні стратегії мінімізації загроз.

Крім того, застосовується інструмент одновимірного аналізу чутливості — діаграма Торнадо, яка ранжує чинники відповідно до ступеня їхнього впливу на фінансові показники проєкту при варіації окремих параметрів. Систематизація ризиків за рівнем невизначеності уможливлює оцінку діапазону коливань для кожного сценарію: від оптимістичного до песимістичного. Комплексне використання зазначених методів створює аналітичне підґрунтя для прийняття обґрунтованих стратегічних рішень.

Практичні результати та обговорення. Практичну апробацію та тестування обчислювального ядра системи здійснено на основі імітаційних даних, змодельованих відповідно до типового енергетичного проєкту «Реконструкція вузлової електричної підстанції 110 кВ» із базовим бюджетом 25 млн грн. До тестового реєстру було включено 5 критичних загроз.

За результатами моделювання методом Монте-Карло встановлено, що середній розмір очікуваних збитків становить 6 591 403 грн (26,4 % від загального бюджету), а показник ризикової вартості (VaR 95%) досягає 13 448 392 грн. Отримані дані забезпечують наукове обґрунтування для формування резервного фонду проєкту в розмірі ~13,45 млн грн, що дозволяє покрити збитки з рівнем довіри 95 %. Середній прогнозований термін затримки реалізації проєкту становить 62 дні, при цьому в межах реалізації найгіршого сценарію затримка може сягати 152 днів (див. рисунок 2).

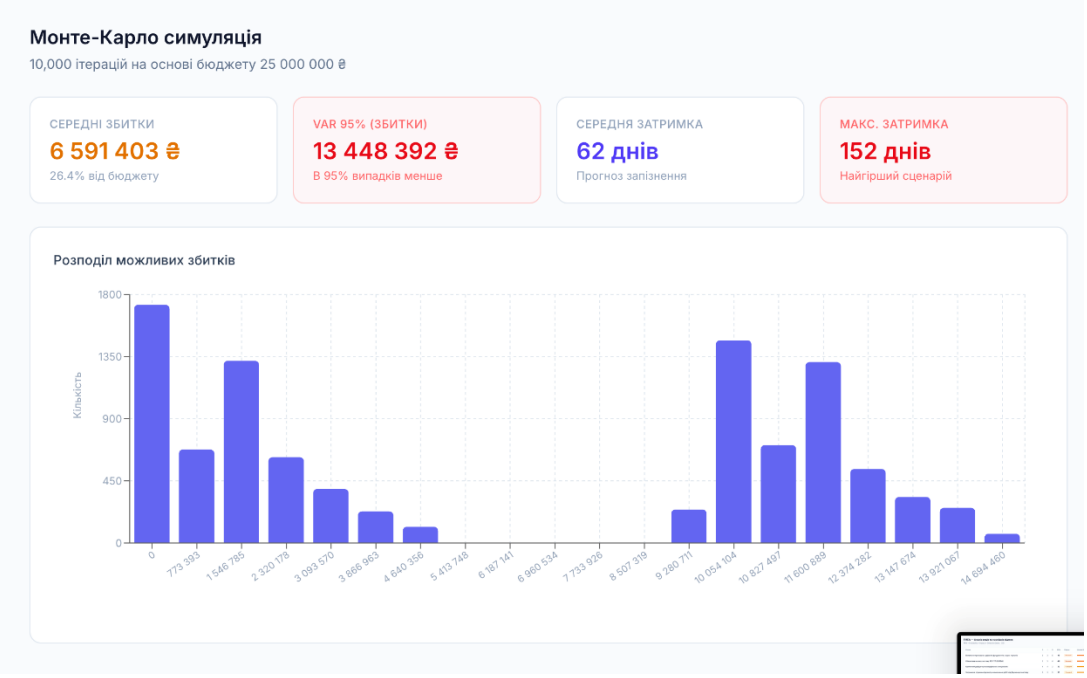


Рисунок 2. Демонстративний приклад симуляції Монте-Карло.

У ході порівняльного аналізу методів Score та RPN обчислювальне ядро

зафіксувало коефіцієнт рангової кореляції Спірмена на рівні ρ=−0,3250. Наявність слабкої оберненої кореляції математично підтверджує, що інтеграція виміру виявлення (Detection) суттєво оптимізує структуру пріоритетності загроз, дозволяючи ідентифікувати латентні ризики (зокрема, «Виявлення прихованих дефектів фундаментів») та перемістити їх у категорію критичних. Це доводить наукову доцільність та ефективність обраної архітектури ядра системи. (див. рисунок 3).

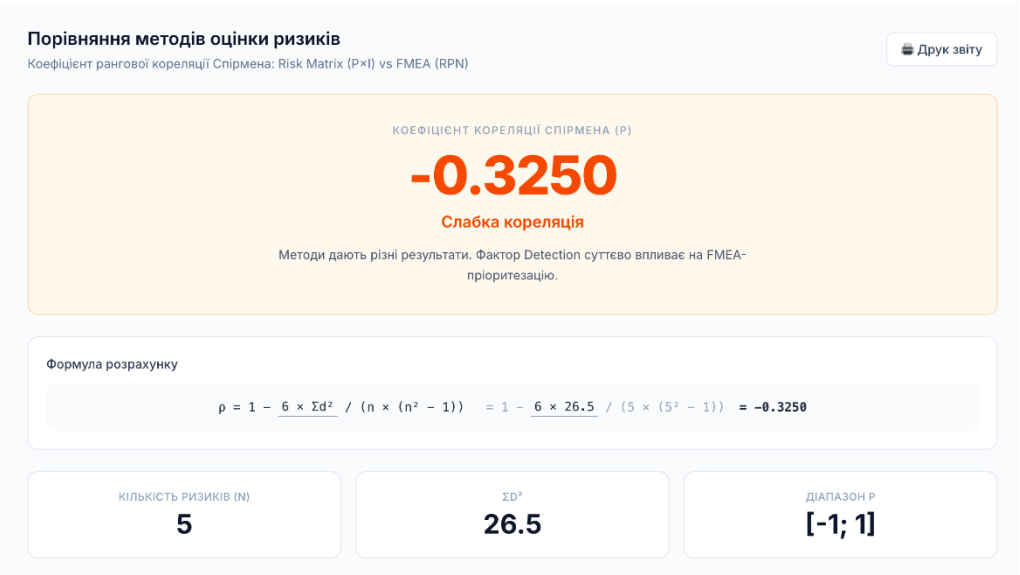


Рисунок 3. Вікно результатів розрахунку коефіцієнта Спірмена.

Окрім того, у межах тестування було верифіковано функціональність підсистеми Event Sourcing (Audit Log), призначеної для автоматичної реєстрації модифікацій ризиків, а також модуля генерації документів, який забезпечує формування аналітичних звітів у форматі PDF (з підтримкою кириличних шрифтів), готових до надання стейкхолдерам.

Висновки. Обґрунтовано та реалізовано високопродуктивну клієнт-серверну архітектуру на базі NestJS, Next.js, PostgreSQL та Prisma ORM, що забезпечує стабільне виконання ресурсомістких ітераційних розрахунків у вебсередовищі. Розроблено аналітичне ядро вебсистеми «VIKTORIA», що здійснює наскрізну автоматизацію семи методів якісного оцінювання, стохастичного моделювання, чутливості, економічної окупності та статистичної валідації на основі єдиного джерела даних, дозволяючи повністю відмовитися від несистематизованих електронних таблиць. Практичне тестування системи на прикладі інженерно-енергетичного проєкту підтвердило точність математичних симуляцій Монте-Карло (VaR 95% визначено на рівні 13.44 млн грн) та верифікувало наукову новизну інтегрованого FMEA-підходу через розрахунок коефіцієнта Спірмена (ρ=−0.3250). Подальший розвиток системи передбачає інтеграцію нормального і трикутного розподілів та розробку публічного API.

Список використаних джерел

1.Project Management Institute. (2021). *Настанова до зводу знань з управління проектами (A Guide to the Project Management Body of Knowledge)* (7-ме вид.). Project Management Institute.

2.Держспоживстандарт України. (2018). *ДСТУ ISO 31000:2018 (ISO 31000:2018, IDT). Керування ризиками. Настанови*. Держспоживстандарт України.

3.Клейн, Д. (2023). *Побудова систем на NestJS : масштабовані Node.js застосунки*. O'Рейлі.

4.Чинна, Т., & Дакетт, Д. (2024). *Розробка на React та Next.js : сучасний фронтенд*. No Starch Press.

5.Вітвер, Н. (2022). *Методи Монте-Карло в управлінні ризиками та фінансах* (2-ге вид.). Wiley.

6.Жигірь, А. А. (2014). Методи кількісної оцінки ризику економічного розвитку підприємства. *Ефективна економіка*, (10). http://www.economy.nayka.com.ua/?op=1&z=3413

7.Степанюк, З. А. (2017). Огляд сучасних методів кількісної оцінки ризиків та впровадження інформаційних технологій в систему ризик-менеджменту. *Економіка. Менеджмент. Бізнес*, 4(22), 176–182.

8.Грицай, О. І., Дефір, І. В., & Козак, О. Є. (2024). Огляд кількісних методів оцінки ризиків зовнішньоекономічної діяльності. *Економіка та суспільство*, (68). https://doi.org/10.32782/2524-0072/2024-68-104

9.Вітлінський, В. В., & Верченко, П. І. (2000). *Аналіз, моделювання та управління економічним ризиком*. КНЕУ.