

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

Краудфандингова платформа на основі технології Blockchain з можливістю
передбачення успішності проєктів за допомогою машинного навчання
(тема)

Виконав:
студент 4 курсу, групи ПЗП-20-10
Жмур Д. М.

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітньо-професійна програма Програмна інженерія
(повна назва освітньої програми)

Керівник: доц. Кириченко І. В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

З.В. Дудар

2024 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра програмної інженерії

Рівень вищої освіти перший (бакалаврський)

Спеціальність 121 – Інженерія програмного забезпечення

(код і повна назва)

Тип програми освітньо-професійна

Освітньо-професійна програма Програмна інженерія

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«__» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Жмуру Дмитру Михайловичу

(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) Краудфандингова платформа на основі технології Blockchain з можливістю передбачення успішності проєктів за допомогою машинного навчання

затверджена наказом по університету від «20» травня 2024 р. № 471Ст

2. Термін здачі студентом закінченої роботи «14» червня 2024 р.

3. Вихідні дані до роботи методи розробки програмних продуктів, методи розробки браузерних додатків, мови програмування.

4. Зміст пояснювальної записки (перелік питань, що потрібно розробити)

Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проектування програмного забезпечення, опис прийнятих програмних рішень, тестування розробленого програмного забезпечення, висновки, додатки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	20.05.2024	<i>виконано</i>
2	Створення специфікації ПЗ	22.05.2024	<i>виконано</i>
3	Проектування ПЗ	24.05.2024	<i>виконано</i>
4	Розробка ПЗ	28.05.2024	<i>виконано</i>
5	Тестування ПЗ	30.05.2024	<i>виконано</i>
6	Оформлення пояснювальної записки	05.06.2024	<i>виконано</i>
7	Підготовка презентації та доповіді	06.06.2024	<i>виконано</i>
8	Попередній захист	09.06.2024	<i>виконано</i>
9	Нормоконтроль, рецензування	06.06.2024	<i>виконано</i>
10	Здача роботи у електронний архів	10.06.2024	<i>виконано</i>
11	Допуск до захисту у зав. кафедри	11.06.2024	<i>виконано</i>

Дата видачі завдання «1» травня 2024 р.

Студент _____ Жмур Д. М.
(підпис)

Керівник роботи _____ доц. кафедри ПІ Кириченко І. В.
(підпис)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи бакалавра, 68 с., 22 рис., 1 табл., 7 додатків, 15 джерел.

БЛОКЧЕЙН, МАШИННЕ НАВЧАННЯ, ШТУЧНИЙ ІНТЕЛЕКТ, КРАУДФАНДИНГОВА ПЛАТФОРМА, ПРОГНОЗУВАННЯ, ПЕРЕДБАЧЕННЯ.

Об'єкт розробки – краудфандингова платформа, що ґрунтується на технології Blockchain і використовує методи машинного навчання для передбачення успішності проєктів.

Ціль роботи – створення інноваційної програмної системи для залучення фінансування на реалізацію цікавих ідей, що використовує принципи децентралізації й аналіз даних за допомогою штучного інтелекту.

Методи дослідження та перелік апаратури – для розробки системи було використано методи програмування на мовах високого рівня, зокрема Java для розробки блокчейну та серверної частини, TypeScript для інтерфейсу користувача. Апаратне забезпечення включає комп'ютери з потужними процесорами та відповідною пам'яттю для обробки великих обсягів даних.

В результаті розробки створено серверну, клієнтську та десктопну частини для краудфандингової платформи на основі технології Blockchain із можливістю передбачення успішності проєктів за допомогою машинного навчання.

Економічна та соціально-економічна ефективність роботи – впровадження краудфандингової платформи на основі Blockchain і машинного навчання може сприяти зростанню ефективності у сфері збору коштів для інноваційних проєктів, зниженню ризиків для інвесторів та загалом підвищенню ефективності використання ресурсів у суспільстві.

BLOCKCHAIN, MACHINE LEARNING, ARTIFICIAL INTELLIGENCE, CROWDFUNDING PLATFORM, FORECASTING, PREDICTION.

The object of development is a crowdfunding platform based on Blockchain technology and using machine learning methods to predict the success of projects.

The aim of the work is to create an innovative software system for raising funds for the implementation of interesting ideas, using the principles of decentralisation and data analysis with the help of artificial intelligence.

Research methods and hardware - to develop the system, we used high-level programming methods, in particular Java for the development of the blockchain and server side, TypeScript for the user interface. The hardware includes computers with powerful processors and adequate memory to process large amounts of data.

The development resulted in the creation of server, client, and desktop parts for a crowdfunding platform based on Blockchain technology with the ability to predict the success of projects using machine learning.

Economic and socio-economic efficiency - the implementation of a crowdfunding platform based on Blockchain, and machine learning can help increase efficiency in raising funds for innovative projects, reduce risks for investors, and generally increase the efficiency of resource use in society.

Я, Жмур Дмитро Михайлович, студент гр. ПЗП-20-10, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Краудфандингова платформа на основі технології Blockchain з можливістю передбачення успішності проєктів за допомогою машинного навчання», що буде представлена до екзаменаційної комісії для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIAr KhNURE. Усі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови до допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ.....	8
1 Аналіз предметної галузі.....	9
1.1 Аналіз предметної галузі.....	9
1.2 Виявлення проблем і актуалізація рішень.....	10
1.3 Визначення аналогів.....	12
1.4 Постановка задачі.....	18
1.4.1 Визначення завдань.....	18
1.4.2 Визначення цільової аудиторії.....	20
2 Формування вимог до програмної системи.....	22
3 Архітектура та проектування.....	24
3.1 UML проектування ПЗ.....	24
3.2 Проектування структури зберігання даних.....	24
3.3 Проектування архітектури ПЗ.....	27
3.5 План роботи та моделювання компонентів.....	29
4 Опис прийнятих програмних рішень.....	31
4.1 Розробка блокчейну.....	31
4.2 Розробка клієнтської частини.....	36
4.3 Розробка серверної частини.....	38
5 Тестування програмного забезпечення.....	39
5.1 Мануальне тестування.....	39
5.2 Модульне тестування.....	44
Висновки.....	46
Перелік джерел посилання.....	47
Додаток А Звіт результатів перевірки унікальності тексту в базі ХНУРЕ.....	49
Додаток Б Перелік джерел посилання за науковими напрямками керівника та науковців кафедри програмної інженерії.....	50
Додаток В Приклад кодів програми.....	51
Додаток Г Специфікація програмного продукту.....	53

Додаток Д Тези XXVIII міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті»	58
Додаток Е Слайди презентації	61

ВСТУП

Сучасна епоха глобалізації та цифровізації стимулює стрімке розширення сфери краудфандингу як інноваційного механізму залучення фінансування для реалізації різноманітних проєктів. Застосування технології блокчейн у «фінансуванні натовпом» відкриває нові перспективи для забезпечення прозорості, безпеки та ефективності цього процесу, а використання методів машинного навчання уможливорює прогнозування успішності фінансованих задумів. На тлі цього виникає актуальність розробки відповідної платформи.

Провідні наукові установи та організації у цій галузі активно досліджують та розвивають методи та підходи до впровадження децентралізації у фінансові та соціальні процеси. Вчені та фахівці по всьому світу зосереджують увагу на розробці нових моделей краудфандингу, які забезпечують максимальну ефективність та мінімізують ризики для учасників. Це свідчить про значний інтерес до проблеми оптимізації процесів добровільного залучення коштів через використання сучасних технологій.

Ціль роботи полягає у створенні краудфандингової платформи, що базується на «ланцюжку блоків» та використовує методи машинного навчання для прогнозування успішності проєктів [1]. Для реалізації цього використовуються мови програмування TypeScript і Java, бази даних Cassandra та MySQL, середовища IntelliJ IDEA та WebStorm.

Ця робота має взаємозв'язок з попередніми дослідженнями у галузі краудфандингу, децентралізованих систем і штучного інтелекту. Вона спирається на відомі методи та практики, одночасно пропонуючи новий підхід до організації та функціонування платформ для групового спонсорювання цікавих ідей.

Іноваційність підходу полягає у розробці інтегрованої системи, яка поєднує такі напрямки, як краудфандинг, блокчейн та машинне навчання для розв'язання важливих проблеми, пов'язаних зі звітністю, надійністю та непередбачуваністю існуючих рішень для спільного фінансування проєктів.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1 Аналіз предметної галузі

За останні двадцять років з Інтернетом сталося чимало змін. Сучасні платформи соціальних мереж замінили Internet Relay Chat (IRC). Складні банківські онлайн послуги замінили базові цифрові платежі. Були випробувані абсолютно нові Інтернет-технології, такі як криптовалюта та блокчейн. Інтернет почав сприйматися як життєво важлива частина людської взаємодії, зв'язку та продовжує розвиток. Ми вже бачили, що таке Web 1.0 і 2.0, але Web 3.0 привносить зовсім новий досвід [2]. Він значною мірою опирається на машинне навчання, штучний інтелект і технологію блокчейну. Це оновлення стосується і процесу краудфандингу.

Краудфандинг (від англійського «crowdfunding» – фінансування від суспільства) представляє собою механізм залучення коштів з метою реалізації продукту, організації заходу, надання допомоги потребуючим, підтримки підприємництва та інших цілей. Нерідко виникає ситуація, коли для втілення в життя дійсно вартісних ідей відсутні необхідні ресурси. Через це не вдається реалізувати сотні стартап-проектів, винаходів, інноваційних ідей. Проте у світовій спільноті є значна кількість людей, які готові підтримати перспективну ідею та вкласти власні кошти в її втілення. Саме з метою забезпечення такої можливості були створені краудфандингові платформи, а цей проєкт є одним з них.

Однак у сучасному краудфандингу є багато недоліків. Найголовніша серед них – відсутність можливості відстежувати транзакції, через що користувачі не можуть мати впевненість у тому, що їхні кошти надійшли саме визначеному адресату. Інша проблема полягає у розпорядженні інвестованими коштами у тому разі, якщо цільова сума не була зібрана. Якщо зловмисник отримає доступ до бази транзакцій, то кошти будуть «повернені» не своїм адресантам. Ще одна проблема – відсутність інформації про перспективність проєктів, на які залучаються фінанси. Тому краудфандингові майданчики, які працюють на основі традиційних методів і технологій, породжують сумніви як у тих, хто має перспективні ідеї, так і у тих, хто

може допомогти з їх фінансуванням [1]. Це призводить до того, що приблизно 60% проєктів не досягають мети збору [3].

Незважаючи на згадані факти, краудфандинг розвивається, а остання розробка, з технологією блокчейну, зробила систему більш прозорою та підзвітною. Концепція краудфандингу Web 3.0 базується на трьох складових: блокчейн, токенизація активів і смарт-контракти.

Існуючі дослідження у галузі блокчейн-краудфандингу говорять про перспективність цього підходу. Насамперед серед переваг використання блокчейну називають децентралізацію – вона робить прозорим пряме джерело коштів, підвищує видимість проєкту для інвесторів і усуває непотрібні вузькі місця. Ще однією з переваг, яку блокчейн і криптографія приносять краудфандингу, є універсальна прийнятність платформи: «Щойно ваш проєкт запрацює за допомогою технології блокчейн, він стане видимим для кожного користувача Інтернету. Будь-який інвестор, який має доступ до Інтернету, зможе легко перевірити кампанію збору коштів і зробити пожертви».

У дослідженні «Prediction of Crowdfunding Project Success Probability using Machine Learning» [4] описано підхід керованого машинного навчання для визначення ймовірності успіху проєктів з платформи Kickstarter. У ньому описаний процес відбору даних, моделювання, обчислення фактичної оціненої ймовірності за допомогою методу плавного сортування, використання подальшої регресії для оцінки точності прогнозованих ймовірностей. Це дослідження може розглядатися як основа для виконання робіт у галузі блокчейн-краудфандингу.

1.2 Виявлення проблем і актуалізація рішень

Перед постановкою завдань для розробки краудфандингової платформи на основі технології Blockchain з можливістю передбачення успішності проєктів за допомогою машинного навчання, важливо визначити основні проблеми, що існують у сфері краудфандингу, та з'ясувати, які рішення можуть бути актуальними для їх вирішення.

Проблеми у галузі краудфандингу:

- недостатня прозорість та довіра. Багато проєктів краудфіндингу стикаються з проблемами щодо прозорості використання зібраних коштів та недовіри з боку учасників;
- високі комісії та обмежені можливості фінансування. Традиційні платформи краудфіндингу можуть вимагати від проєктів високі комісії, а також обмежувати доступ до фінансування для певних категорій проєктів або учасників;
- відсутність прогнозування успішності проєктів. Часто важко передбачити успішність краудфіндингового проєкту самостійно, що може призводити до ризиків для інвесторів і невдач для авторів.

Актуалізація рішень:

- забезпечення прозорості та безпеки за допомогою технології Blockchain. Використання технології Blockchain дозволить забезпечити прозорість та недоторканість транзакцій, зменшити шанси на шахрайство та підвищити довіру користувачів;
- зниження комісій та обмежень для більшої доступності. Зменшення комісій та обмежень на платформі краудфіндингу сприятиме більшій доступності для різних категорій проєктів та учасників, що може позитивно вплинути на розвиток галузі. Використання технології Blockchain дозволить зменшити комісії за рахунок таких аспектів:
 - а) прозорість та недоторканість: блокчейн забезпечує прозорість та недоторканість всіх транзакцій, які відбуваються на платформі. Це означає, що будь-який учасник може перевірити історію транзакцій та впевнитися в їхній відповідності до правил та умов платформи, що сприяє зниженню ризиків та необґрунтовано високих комісій;
 - б) глобальний доступ: блокчейн-платформи мають потенціал для глобального доступу, оскільки вони не обмежені територіальними межами. Це може зменшити бар'єри для участі в краудфіндингу та забезпечити більший ринок для проєктів та інвесторів;

- використання аналітичних методів для передбачення успішності проєктів. Застосування аналітичних методів, таких як машинне навчання, для аналізу даних та передбачення успішності проєктів, допоможе інвесторам приймати більш обґрунтовані рішення щодо фінансування.

Виявлені проблеми у сфері краудфандингу вимагають комплексного підходу до їх вирішення. Використання технології Blockchain та машинного навчання може стати ефективними інструментами для забезпечення прозорості, безпеки та передбачуваності у цій галузі.

1.3 Визначення аналогів

Коли мова йде про краудфандинг, першим, що спадає на думку, є майданчик Kickstarter [5]. Його головною місією є допомога у реалізації творчих проєктів. Ця платформа надає можливість залучати кошти на 13 категорій ідей: комікси, ремесла, танці, дизайн, мода, фільми, їжа, ігри, журналістика, музика, фотографування, видавництво, технології, театр. На сайті можна фільтрувати проєкти за категорією та походженням і сортувати за такими властивостями: популярність, новизна, дата завершення, фінансування, підтримка, місцезнаходження відносно користувача (див. рис. 1).

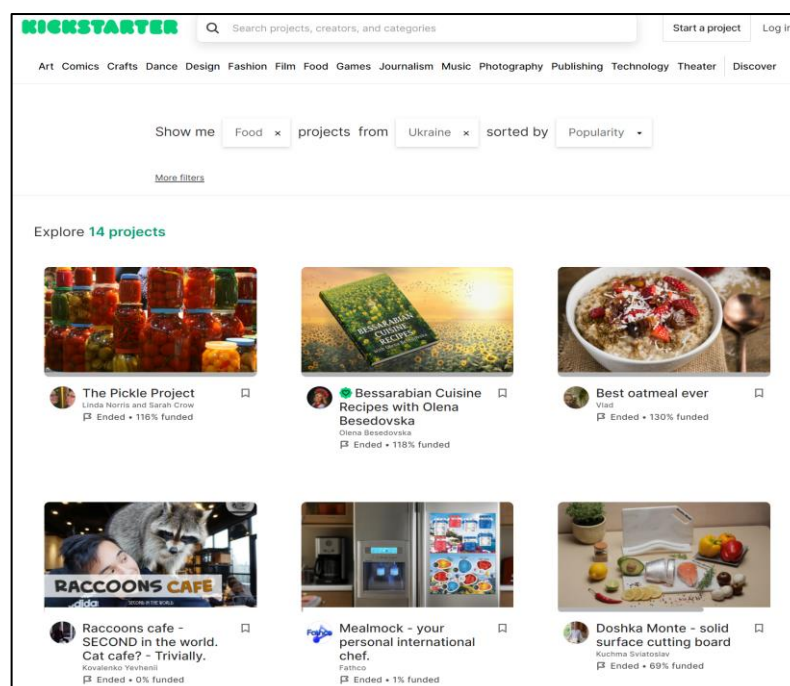


Рисунок 1 – Платформа Kickstarter (за даними [6])

Безумовна перевага цієї платформи – це популярність і впізнаваність: Kickstarter є одним із найвідоміших і найпопулярніших краудфандингових майданчиків у світі. Завдяки великій базі користувачів, проекти на Kickstarter мають можливість залучити значну кількість інвесторів. Велике розмаїття проектів є іншою позитивною рисою цього майданчика, однак кількість категорій обмежена. Головним недоліком платформи вважаються високі комісії: Kickstarter стягує 5% від залучених коштів за свої послуги, що може бути обтяжливим для авторів та інвесторів. Перекази здійснюються за допомогою Amazon Payments, яка стягує ще 3-5%. Крім того, фінансування на платформі здійснюється за принципом «все або нічого», тобто проєкт отримує фінансування тільки якщо досягає своєї мети. Іншим «мінусом» є закритість системи: Kickstarter не надає можливості відстеження кампанії збору коштів.

Найближчим же аналогом розроблюваної платформи є екосистема Tesra Space, яка дозволяє інвестувати в проєкти, купуючи їхні токени й отримуючи прибуток. Вона складається з трьох проєктів: власна валюта TesraCoin, платформа краудфандингу, децентралізована біржа Tesra DEX, яка ще створюється. Система заохочує залучати та здійснювати інвестиції у чотирьох галузях: ігри, озеленення, блокчейн, мистецтво (див. рис. 2).

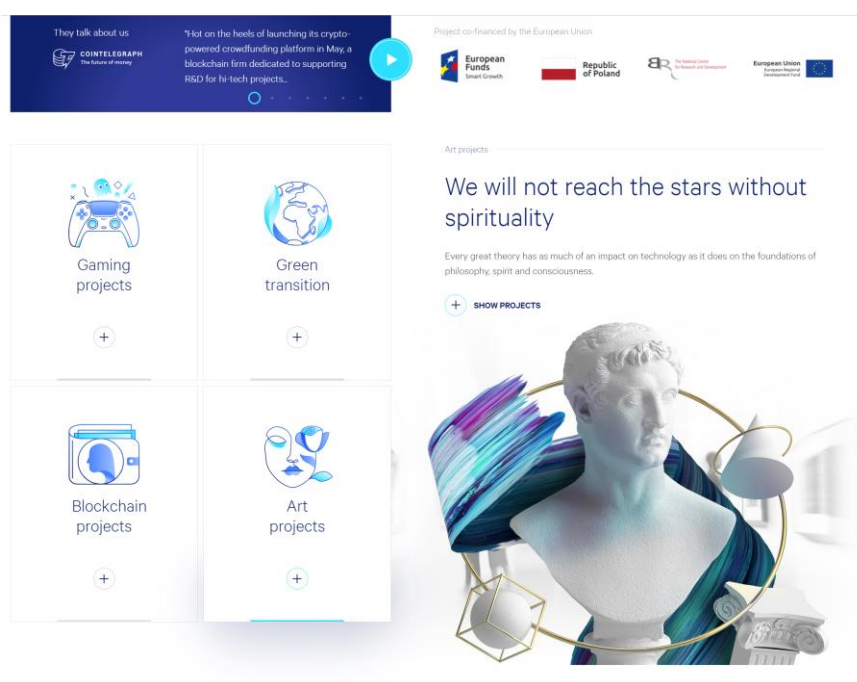


Рисунок 2 – Платформа Tesra Space (за даними [7])

Найголовнішими перевагами цієї платформи є відсутність оподаткування при використанні криптовалют, можливість прозоро виходити з інвестицій і стежити за ходом проєктів. Її недоліками можна назвати поганий UX-дизайн: хоча ще не розпочаті проєкти показані на загальній сторінці, перейти на їхню сторінку неможливо, а при натисканні на їхню назву загальна сторінка просто оновлюється; при натисканні на будь-яку категорію з домашньої сторінки, користувач просто перенаправляється на сторінку всіх проєктів і категорію знову потрібно обирати вручну. Ще одним мінусом є малий перелік категорій проєктів, на які можна збирати та інвестувати кошти.

Третій аналог – платформа Fundly (див. рис. 3).

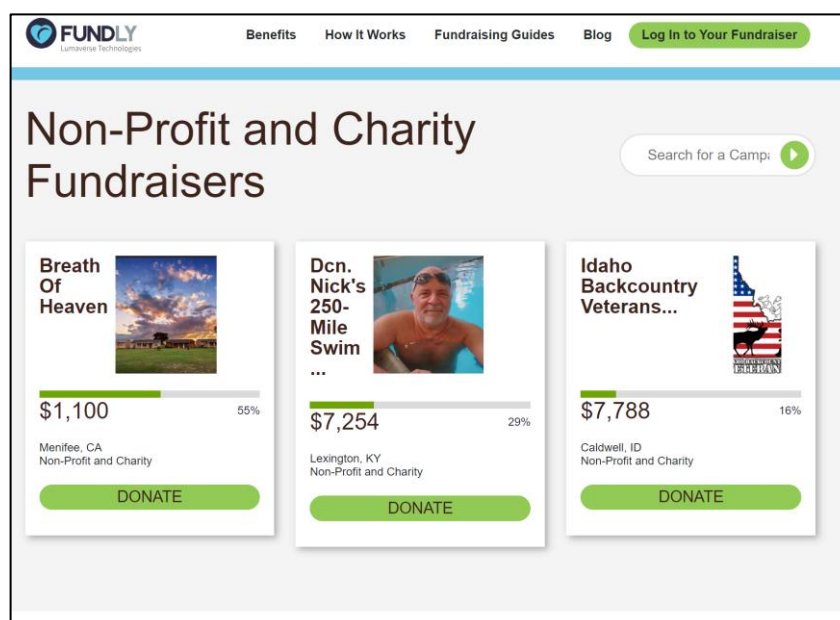


Рисунок 3 – Платформа Fundly (за даними [8])

Серед її переваг:

- прозорість: хоча Fundly підтримує анонімні перекази, на сайті можна переглянути історію кампанії збору коштів;
- соціальна інтеграція: Fundly добре інтегрований з соціальними мережами, що спрощує процес залучення коштів через особисті мережі контактів;
- фокус на особистих та благодійних кампаніях: Fundly підтримує широкий спектр благодійних та особистих кампаній.

Слід виділити такі недоліки:

- не є ідеальним вибором для комерційних або стартап проєктів: Fundly має обмежені можливості для бізнес-проєктів;
- комісії: платформа стягує щонайменше 5% комісії за транзакції, що може уповільнити кампанії;
- неінтуїтивний інтерфейс: існуючі елементи інтерфейсу не надають можливості перейти від сторінки проєкту до списку всіх проєктів; з головної сторінки перейти на список усіх проєктів категорії можливо, тільки спочатку обравши категорію, а потім натиснувши на зелену стрілку у пошуковому полі; на головну сторінку можна повернутися, лише натиснувши на логотип платформи.

GoFundMe — це платформа, яка надає можливість організувати збори коштів на широкий спектр подій, починаючи від святкування дня народження, і закінчуючи подоланням складних життєвих обставин, таких як нещасні випадки та хвороби [9]. Цей майданчик має приємну та зручну стартову сторінку (див. рис. 4).

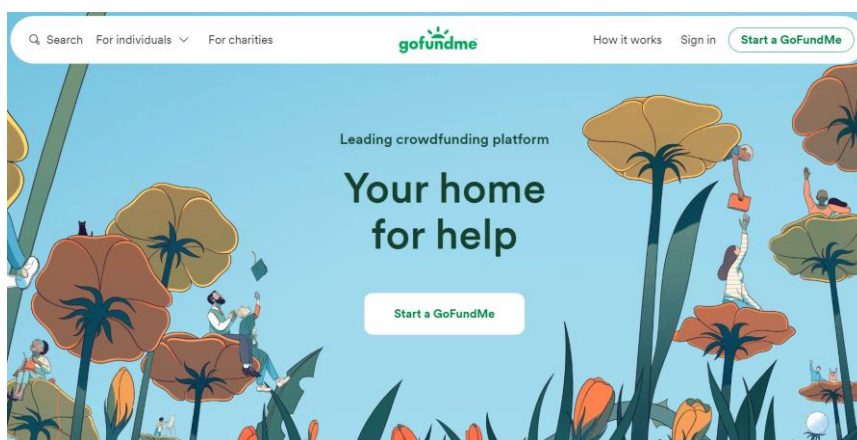


Рисунок 4 – Платформа GoFundMe (за даними [10])

Його переваги:

- особисті та благодійні проєкти: GoFundMe спеціалізується на особистих та благодійних кампаніях, дозволяючи користувачам збирати кошти для різноманітних потреб;
- добровільні комісії: GoFundMe надає можливість робити пожертви з нульовою комісією.

Недоліки:

- обмеження на типи проєктів: GoFundMe не підходить для комерційних або стартап проєктів, оскільки орієнтована на особисті та благодійні ініціативи;
- мінімальні можливості для бізнесу: GoFundMe не надає інструментів для розвитку бізнесу чи стартапів;
- невелика кількість доступних країн: GoFundMe надає можливість збору коштів лише у певних країн, список яких не містить України.

Indiegogo є ще одним відомим аналогом платформи, що розробляється. Майданчик надає можливість залучати кошти на 3 дуже широкі категорії речей: «технології та інновації», творчі роботи, громадські проєкти (див. рис. 5).

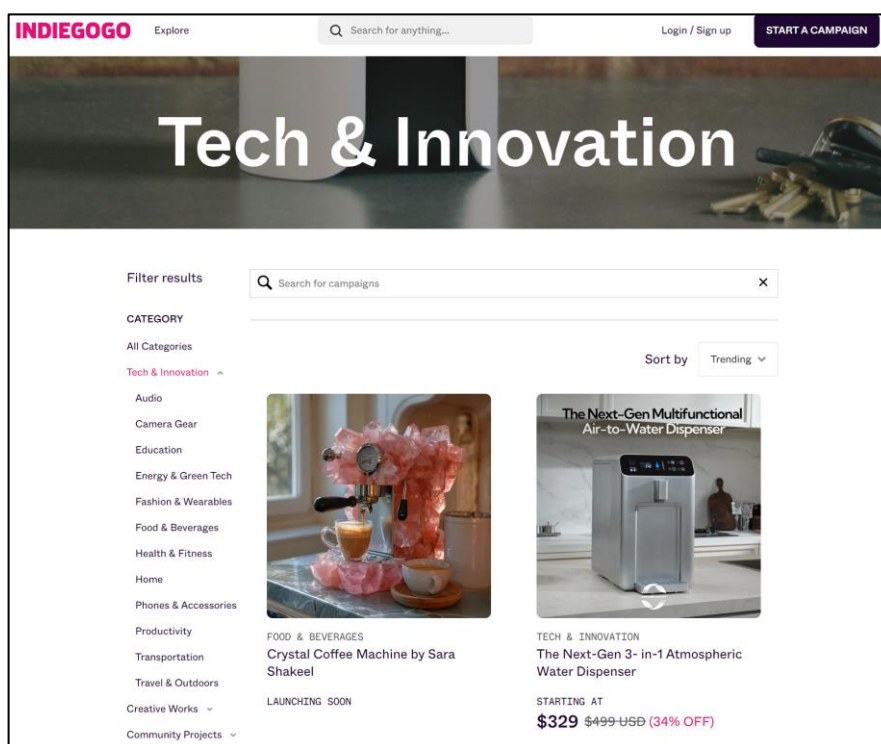


Рисунок 5 – Платформа Indiegogo (за даними [11])

Переваги:

- гнучкість фінансування: Indiegogo дозволяє авторам проєктів обирати між фіксованим та гнучким фінансуванням, що надає більше варіантів для залучення коштів;
- підтримка стартапів: Indiegogo пропонує додаткові послуги для стартапів, включаючи інкубаційні програми та доступ до ресурсів;

- система винагород: Indiegogo надає можливість налаштувати винагороди для кожної суми інвестиції;
- велике розмаїття проєктів: Indiegogo допомагає залучати кошти на проєкти у 28 категоріях, яких достатньо щоб віднести до них майже будь-яку ідею.

Недоліки:

- менша аудиторія: хоча Indiegogo є популярною платформою, її аудиторія все ще менша, порівняно з Kickstarter;
- комісії: Indiegogo стягує 4% комісії з успішних зборів і 9% за збереження коштів після неуспішної кампанії, що може зменшити чисту суму, яку отримують автори проєктів;
- велика кількість проєктів: хоча це хороший показник, за такої кількості аудиторії проєкт може легко «втратитися» серед сотень інших і не досягти своєї мети.

У таблиці 1 наведено порівняльний аналіз аналогів і нової платформи за такими показниками: прозорість, комісії, вигоди, наявність прогнозування успіху.

Таблиця 1 – Порівняльний аналіз розробки й аналогів (виконано самостійно)

Краудфандинговий майданчик	Прозорість	Комісії	Вигоди для інвесторів	Прогнозування успіху проєкту
Платформа, що розробляється	Доступ до даних через блокчейн	2%	Відшкодування	Є
Kickstarter	Немає	10%	Відшкодування та винагороди	Немає
Indiegogo	Немає	4-9%	Винагороди	Немає
GoFundMe	Історія кампанії	0%	Немає	Немає
Fundly	Історія кампанії	5%	Немає	Немає
Tecra Space	Доступ до даних через блокчейн	10%	Примноження інвестицій	Немає

Дані з таблиці говорять про наявність одних із найкращих показників по всім пунктам у спроектованій платформі. Лише майданчик GoFundMe має кращі умови для пожертв, однак це пояснюється тим, що він сконцентрований на благодійності, а не на комерційних проєктах. Платформа Tesra Space має вигідніші умови для інвесторів за рахунок токенизації активів, однак ці вигоди можуть нівелюватися непомірною комісією у 10%. У той же час лише нова платформа буде мати можливість передбачати успіх проєктів, що робить її унікальною та привабливою. Також, можливість примножити свої інвестиції може бути введена у подальших версіях продукту без підвищення комісій.

1.4 Постановка задачі

1.4.1 Визначення завдань

Постановка задачі проєкту «Краудфандингова платформа на основі технології Blockchain з можливістю передбачення успішності проєктів за допомогою машинного навчання» визначає основні цілі та завдання, які необхідно вирішити для досягнення успішного впровадження системи. Ці завдання визначають основні напрямки розробки та дії, які потрібно вжити для досягнення мети проєкту.

Головною метою проєкту є створення та впровадження краудфандингової платформи, яка надасть користувачам можливість створювати та фінансувати проєкти, а також використовувати технологію Blockchain для забезпечення безпеки та прозорості транзакцій. Ключові аспекти мети проєкту включають:

- створення безпечного середовища для фінансування проєктів за допомогою криптовалюти та технології Blockchain;
- реалізація системи передбачення успішності проєктів на основі аналізу даних та використання методів машинного навчання;
- забезпечення прозорості та надійності процесу краудфандингу за допомогою технології Blockchain.

Основні завдання проєкту включають:

- розробка краудфандингової платформи. Розробка інтерактивної та інтуїтивно зрозумілої веб-платформи, що дозволяє користувачам створювати, редагувати та фінансувати проєкти;
- інтеграція технології Blockchain. Впровадження технології Blockchain для забезпечення безпеки та прозорості транзакцій та фінансових операцій;
- реалізація системи передбачення успішності проєктів. Розробка та впровадження моделі передбачення успішності проєктів на основі аналізу даних, використовуючи методи машинного навчання;
- тестування та вдосконалення системи. Проведення тестування розробленої системи з метою виявлення помилок та недоліків. Внесення необхідних коректив та вдосконалень для забезпечення оптимального функціонування системи;
- оптимізація використання ресурсів. Забезпечення ефективного використання обчислювальних та мережевих ресурсів для забезпечення швидкої та надійної роботи платформи. Мінімізація витрат, пов'язаних з обробкою та зберіганням даних на блокчейні та серверах;
- забезпечення масштабованості. Розробка системи з урахуванням можливості масштабування, щоб забезпечити плавну роботу платформи при збільшенні обсягів користувачів та проєктів. Використання технічних рішень, які дозволяють ефективно розширювати функціональність та продуктивність системи;
- забезпечення сумісності та інтеграції. Впровадження стандартів та протоколів, що дозволяють інтегрувати платформу з іншими сервісами та додатками. Забезпечення сумісності інтерфейсів для взаємодії зі сторонніми системами та сервісами;
- забезпечення правової відповідності. Врахування правових аспектів, пов'язаних із зберіганням та обробкою особистих даних користувачів та фінансових транзакцій. Дотримання вимог законодавства щодо криптовалют та фінансових послуг у різних юрисдикціях.

1.4.2 Визначення цільової аудиторії

Цільова аудиторія для системи, що розроблюється, включає різноманітні групи користувачів, які мають різні цілі та потреби:

Проектні автори.

- індивідуальні підприємці, стартапи, митці та інші творчі особи, які шукають фінансування для реалізації своїх ідей та проєктів;
- компанії, які розробляють нові продукти або послуги та шукають ресурси для їхнього запуску на ринок;
- наукові дослідники, які шукають фінансування для реалізації дослідницьких проєктів та винаходів.

Інвестори.

- індивідуальні інвестори, які шукають можливості для інвестування в інноваційні та перспективні проєкти;
- великі фонди та інвестиційні компанії, які зацікавлені в ризикованих, але потенційно високодохідних можливостях;

Експерти та консультанти.

- фахівці з різних сфер, які можуть надати консультації та експертні оцінки проєктів;
- аналітики та експерти з машинного навчання, які можуть допомогти у вивченні та аналізі даних для передбачення успішності проєктів;
- регуляторні органи та правозахисні організації: органи державного регулювання та нагляду, які відповідають за забезпечення відповідності процесів краудфандингу правовим нормам та стандартам; організації, які займаються захистом прав споживачів та інвесторів.

Підприємства та бізнес-спільнота.

Крім індивідуальних проєктів, деякі підприємства можуть шукати фінансування через краудфандингові платформи для реалізації своїх проєктів або розширення бізнесу [12].

Активісти та благодійники.

Ця група користувачів може бути зацікавлена у фінансуванні проєктів, які мають соціальну чи екологічну спрямованість, або участі в благодійних ініціативах через краудфандинг.

Блогери.

Зйомки й організація деяких відео вимагає багато ресурсів. Насамперед, потрібно оплатити роботу операторів і монтажерів. Якщо це шоу, необхідно найняти саунд-продюсерів, стилістів, акторів, а також заохотити гостей. Крім того, реалізація ідеї блогера часто потребує спеціального обладнання. Саме тому їм у нагоді може стати залучення коштів від їхніх підписників, адже це в інтересах обох сторін: глядачі побачать довгоочікуване відео від свого улюбленця, а блогеру не доведеться викладати неякісний контент і накопичувати кошти половину своєї кар'єри для реалізації своїх ідей у блозі. Зрозуміло, що для таких цілей будуть використовуватися краудфандингові майданчики. Прозорість платформи дозволить взаємодіяти з відеохостингами, щоб забезпечувати покарання для тих блогерів, які обдурили своїх підписників, зібравши з них кошти та витративши їх не за призначенням.

Артисти.

Починаючі музиканти й актори не можуть собі дозволити встановлювати високі ціни на білети і через це не мають можливості виступати далеко за межами міста, у якому перебувають, тому що переїзд із інструментами та проживання в інших містах або країнах дорого коштує. Однак навіть у далеких місцях можуть знайтися заможні фанати цих артистів, які можуть профінансувати виступ на «рідній» сцені.

Цільова аудиторія системи «Краудфандингова платформа на основі технології Blockchain з можливістю передбачення успішності проєктів за допомогою машинного навчання» охоплює широкий спектр учасників, що забезпечить різноманітність та динаміку на платформі.

2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

У першій версії програмної системи передбачається створення краудфандингової платформи та інтеграція в неї блокчейн-складової, а також машинного навчання [13, с. 317]. Вимоги будуть розглядатися з точки зору функціональності, надійності, безпеки та зручності користування:

Функціональні вимоги.

Реєстрація та автентифікація користувачів:

- система повинна надавати можливість реєстрації нових користувачів та автентифікації вже зареєстрованих;
- користувачі повинні мати можливість управляти своїм акаунтом і змінювати в ньому дані.

Створення проєктів:

- користувачі повинні мати можливість створювати нові проєкти на платформі, вказуючи їхню назву, опис і ціль збору.

Фінансування проєктів:

- користувачі повинні мати можливість фінансувати існуючі проєкти на платформі;
- система повинна надавати можливість відстеження стану фінансування кожного проєкту;
- система повинна зберігати транзакції за допомогою технології Blockchain для забезпечення прозорості та безпеки.

Передбачення успішності проєктів:

- використовуючи техніки машинного навчання, система повинна прогнозувати успішність проєктів на основі різноманітних факторів;
- аналітичні дані та результати передбачення мають бути доступні користувачам у зручному для сприйняття вигляді.

Вимоги надійності.

Захист від шахрайства:

- система повинна містити механізми захисту від шахрайства та недопущення фінансових зловживань, використовуючи технологію Blockchain.

Вимоги безпеки.

Конфіденційність даних:

- система повинна забезпечувати конфіденційність особистих даних користувачів та даних проєктів. Доступ до конфіденційних даних повинен бути обмежений та контрольований.

Цілісність транзакцій:

- доступ до зміни інформації про здійснені перекази коштів має бути закритий.

Вимоги зручності:

- зручний інтерфейс. Інтерфейс користувача повинен бути зрозумілим і легким у використанні для різних категорій користувачів;
- швидка навігація. Навігація по системі має бути інтуїтивною, надаючи можливість швидко знайти потрібну інформацію.

Апаратні вимоги:

- сучасний процесор. Додаток для підтримки блокчейну повинен запускатися на системі, яка має щонайменше 6 логічних ядер (потоків);
- містка ОЗП. Додаток має коректно працювати на системі, яка має щонайменше 32 ГБ оперативної пам'яті. Якщо, пам'ять буде не настільки містка, вона швидко почне переповнюватися даними блоків і викидати OutOfMemoryError;
- швидкий Інтернет. Додаток повинен мати доступ до Інтернету на швидкості щонайменше 80 Мбіт/с. В іншому випадку він теж буде перешкоджати роботі блокчейну, тому що інші учасники будуть дуже довго чекати відповіді від нього. Система не повинна надавати можливість доєднатися до блокчейну з повільнішим Інтернетом.

3 АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ

3.1 UML проектування ПЗ

Краудфіндингова платформа на основі технології Blockchain з можливістю передбачення успішності проектів за допомогою машинного навчання є складною, децентралізованою та розподіленою системою, яка складається з декількох компонентів, що взаємодіють між собою.

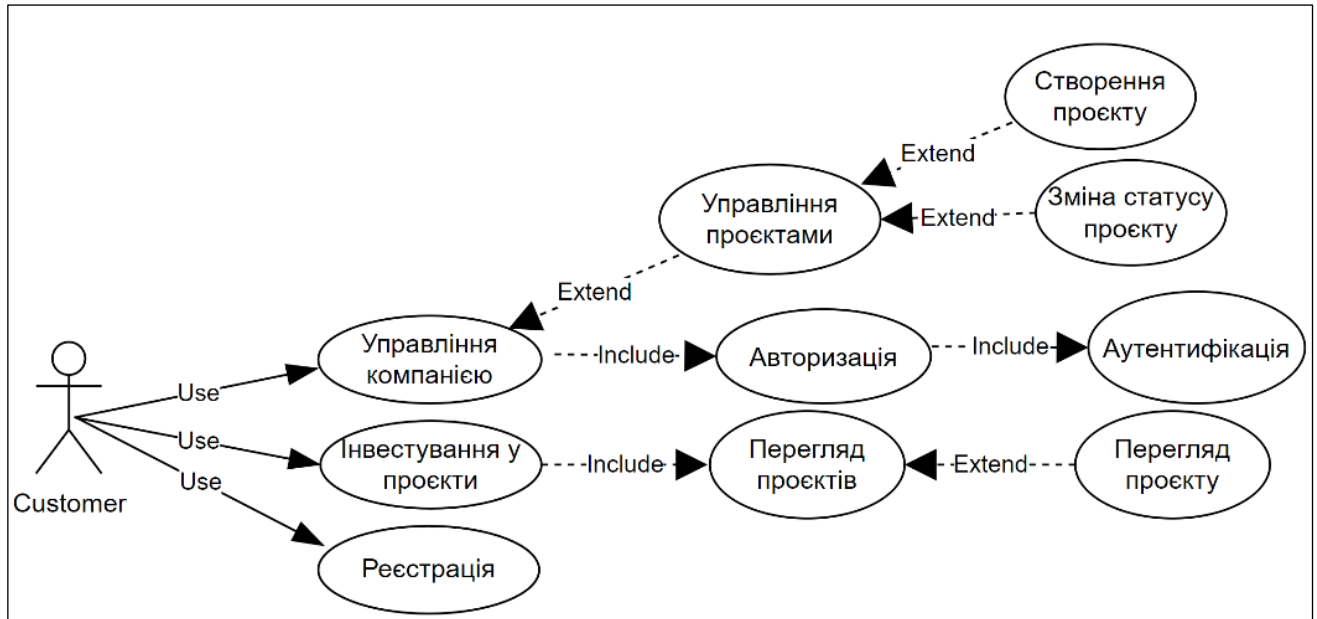


Рисунок 5 – Use Case діаграма (рисунок створено самостійно)

Система розроблена для підтримки наступних функціональних можливостей (див. рис. 5):

- створення та управління проектами краудфіндингу;
- збір коштів від інвесторів;
- прогнозування успішності проектів за допомогою машинного навчання;
- забезпечення безпечного та прозорого середовища для краудфіндингу.

3.2 Проектування структури зберігання даних

База даних буде розділена на 3 частини: блокчейн, управління блокчейном, управління командами (див. рис. 6).

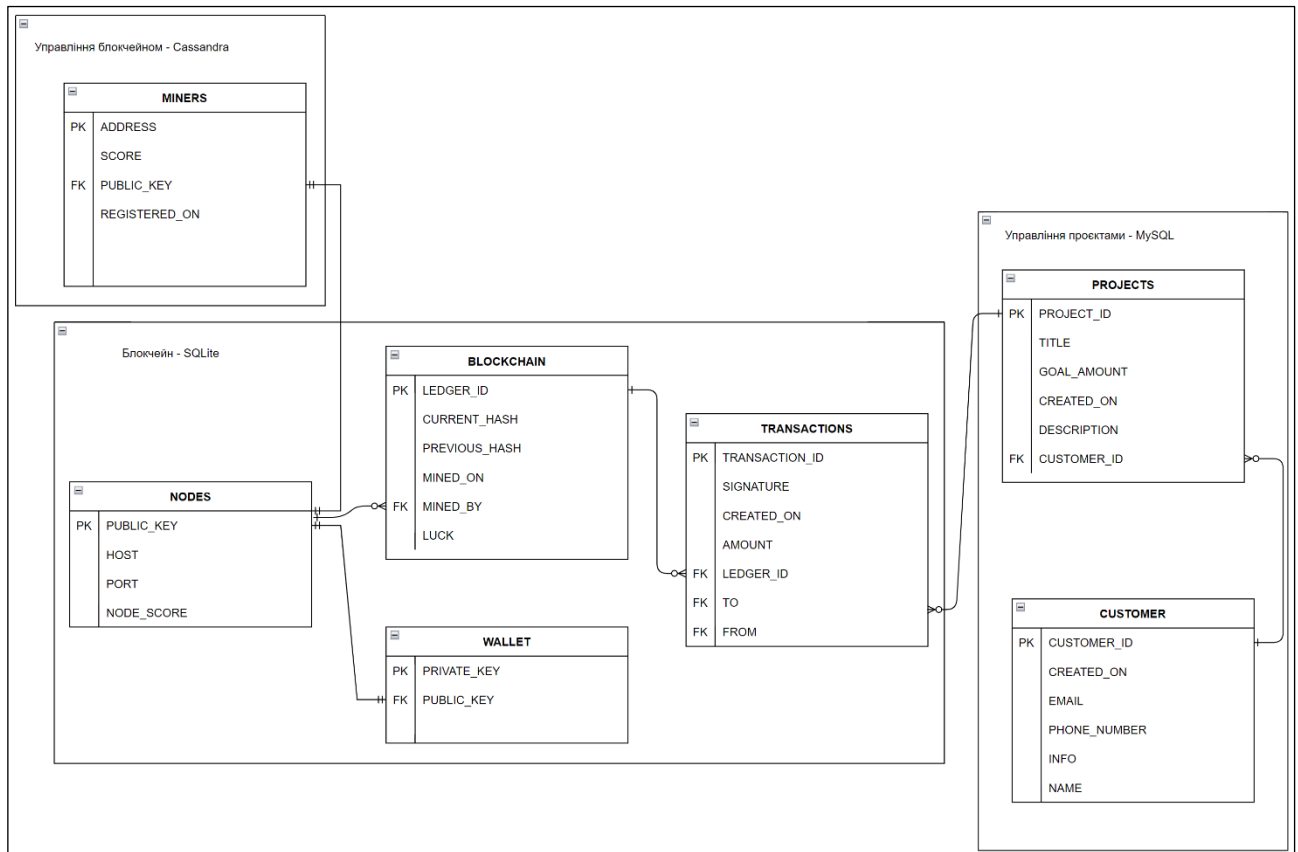


Рисунок 6 – ER-діаграма бази даних (рисунок створено самостійно)

Першу частину забезпечуватимуть вузли блокчейну. У ній зберігатиметься інформація про транзакції з прив'язкою до блоків. Також, у ній будуть зберігатися потрібні дані інших майнерів блокчейну, їхні публічні ключі, власні публічні та приватні ключі. Так як доступ до цієї бази даних буде здійснюватися з десктопного додатку, найкращим вибором СКБД у даному випадку буде SQLite, щоб не займати зайвого місця на комп'ютерах користувачів.

Друга частина буде слугувати для компонента «Blockchain Gateway Server», який здійснюватиме зв'язок із блокчейном. Тому у цій частині повинна зберігатися необхідна серверу інформація про вузли блокчейна. Вона повинна працюватися дуже швидко, щоб не спричиняти затримок у роботі блокчейну, що призведе до серйозної денормалізації. Для денормалізації з метою пришвидшення запитів найкраще підійде Cassandra.

Третя частина знадобиться для основної інформації, читання якої буде відбуватися найчастіше. Дані з різних таблиць будуть зв'язані тісно, але найчастіше

використовуватися будуть окремо. Тут важлива компактність, цілісність і доступність даних, тому хорошим вибором СКБД буде MySQL.

Тепер проведемо денормалізацію бази даних з метою пришвидшення швидкодії системи (див рис. 7).

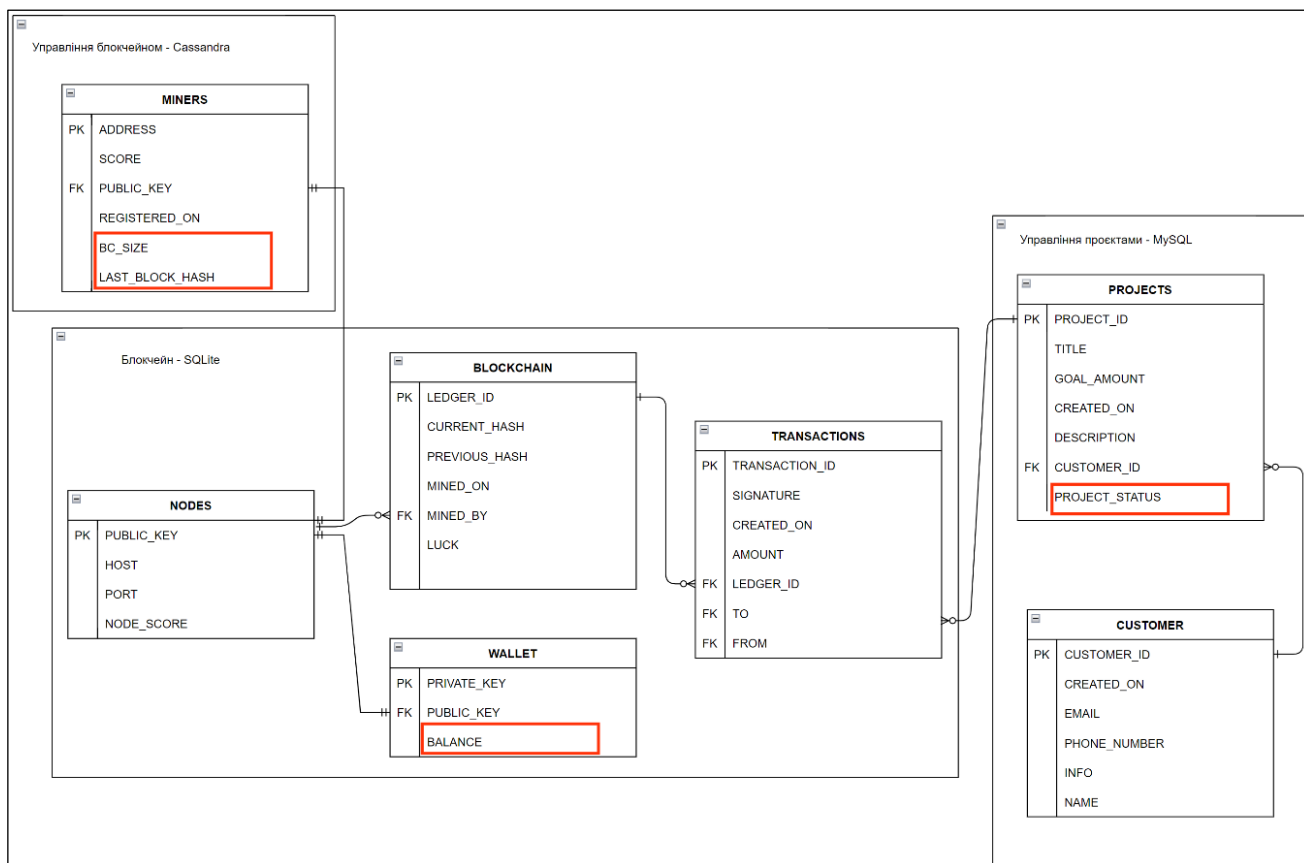


Рисунок 7 – ER-діаграма денормалізованої бази даних (рисунок створено самостійно)

Серверу потрібно знати, які вузли мають актуальний блокчейн, тому він буде постійно робити запити метаданих до різних вузлів перед тим, як зробити запит потрібної інформації. Щоб цього уникнути введемо такі характеристики блокчейну, як його розмір і хеш останнього блоку до таблиці «Miners». Ці характеристики будуть оновлюватися самими вузлами кожного разу, коли вони отримують новий блок.

Майнери захочуть знати свій баланс перед тим як виводити винагороду на свій рахунок. Щоб не перебирати кожного разу свої створені блоки та не перераховувати винагороду, що також створить незручність визначення того, яка

винагорода вже була виведена, а яка ще ні, можна ввести атрибут балансу до таблиці «Wallet» і перераховувати його кожного разу після створення блоку та виведення коштів.

Користувачам корисно буде відразу дізнатися стан збору коштів на проєкт перед тим, як переходити на його сторінку. Щоб не робити запит до блокчейну для всіх проєктів на сторінці кожного разу, коли хтось переглядає їхній список, введемо атрибут статусу проєкту до таблиці «Projects» і оновлюватимемо його для окремого проєкту лише після переказу коштів на його рахунок.

3.3 Проектування архітектури ПЗ

Платформа побудована на основі мікросервісної архітектури, де кожен компонент є окремим сервісом, що виконує певну функцію. Це дозволяє масштабувати систему та полегшує розробку й обслуговування. Компоненти системи (див. рис. 8):

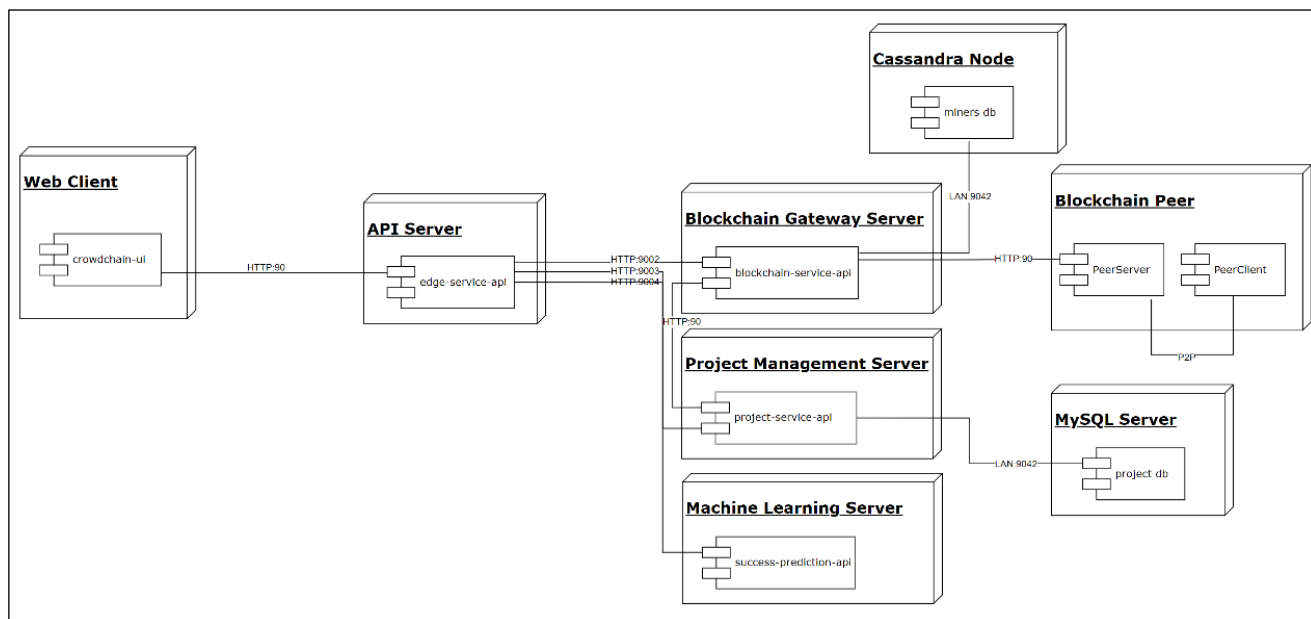


Рисунок 8 – Діаграма розгортання системи (рисунок створено самостійно)

- **WEB-клієнт** – це веб-додаток, який використовується користувачами для взаємодії з системою. Він надає інтерфейс для створення та управління даними проєктів, інвестування в проєкти та перегляду прогнозів їхньої успішності;

- API-сервер – це компонент, який надає RESTful API для доступу до даних та функціональних можливостей системи. Він використовується веб-клієнтом для взаємодії з іншими компонентами;
- сервер передбачення успішності – це компонент, який використовує машинне навчання для прогнозування успішності проєктів. Він може використовувати різні фактори, такі як назва проєкту, його опис, історія попередніх проєктів автора і дані з відкритих джерел про інші проєкти для створення прогнозу;
- сервер шлюзу блокчейну – це компонент, який використовується для взаємодії з Blockchain. Він дозволяє системі записувати та читати дані з розподіленої бази даних;
- учасник блокчейну – це десктопний-додаток, який використовується користувачами для участі в колективному зберіганні даних транзакцій, таких як: ключ інвестора, назва проєкту, сума переказу;
- сервер керування проєктами – це компонент, який використовується для оперування даними проєктів краудфандингу. Він зберігає інформацію про проєкти, такі як: назва й опис проєкту, команда проєкту та цільовий обсяг коштів;
- сервер MySQL – це база даних, яка використовується для зберігання даних проєктів, їхніх команд та інвесторів;
- вузол Cassandra – це база даних, яка використовується для зберігання даних про Blockchain. Cassandra має високий рівень доступності, що робить її ідеальною для швидкого доступу до даних.

3.4 Проектування системи

Система спроектована з урахуванням наступних принципів:

- модульність: система складається з декількох незалежних компонентів, що полегшує розробку та обслуговування;
- масштабованість: система може бути масштабована для підтримки більшої кількості користувачів і функціональних можливостей;

- надійність: система розроблена для забезпечення високого рівня доступності та стійкості до відмов;
- безпека: система використовує різні заходи безпеки, включаючи протокол OAuth 2.1 для управління обліковими даними та доступом, захищаючи дані користувачів та інвестиції.

Система краудфандингової платформи на основі Blockchain з можливістю передбачення успішності проектів за допомогою машинного навчання є складною та потужною системою, що може бути використана для підтримки широкого спектру краудфандингових проектів. Система розроблена з урахуванням принципів модульності, масштабованості, надійності та безпеки.

3.5 План роботи та моделювання компонентів

На першому етапі роботи розроблено додаток для підтримки блокчейну. Спочатку створені класи-моделі, які становлять ядро блокчейну: Block (клас блока), Transaction (клас транзакції), Wallet (клас гаманця). Після цього налаштовано зберігання даних, для того щоб після перезапуску додатків дані не втрачалися. З цією метою створено стартовий клас, який розширює клас Application. У стартовому класі перевизначимо метод `init()`, у якому встановлюємо з'єднання з базою даних і завантажуються звідти інформація про блокчейн, яка містилася в додатку на момент останнього виходу. Далі створено UI-компоненти для відстеження транзакцій: в'ю (англ. View) та контролер (англ. Controller). Потім налаштовано мережу та мультипотоківість, для чого створено чотири класи: UI (потік для інтерфейсу користувача), MiningThread (потік для майнінгу), PeerClient (потік для отримання інформації про блокчейн), BlockchainServerWrapper (клас для надання даних про блокчейн). Наприкінці створено 2 класи сервісного рівня, які виконують всю бізнес-логіку: MiningService, ValidationService.

Другий етап присвячений створенню серверних додатків, який обробляє запити від користувачів, що взаємодіють із платформою через браузер. Для цього ініціалізовано додатки Spring Boot через сайт <https://start.spring.io/> і створено в них пакети під чотирьохрівневу архітектуру: repository, service, controller (див. рис. 9).

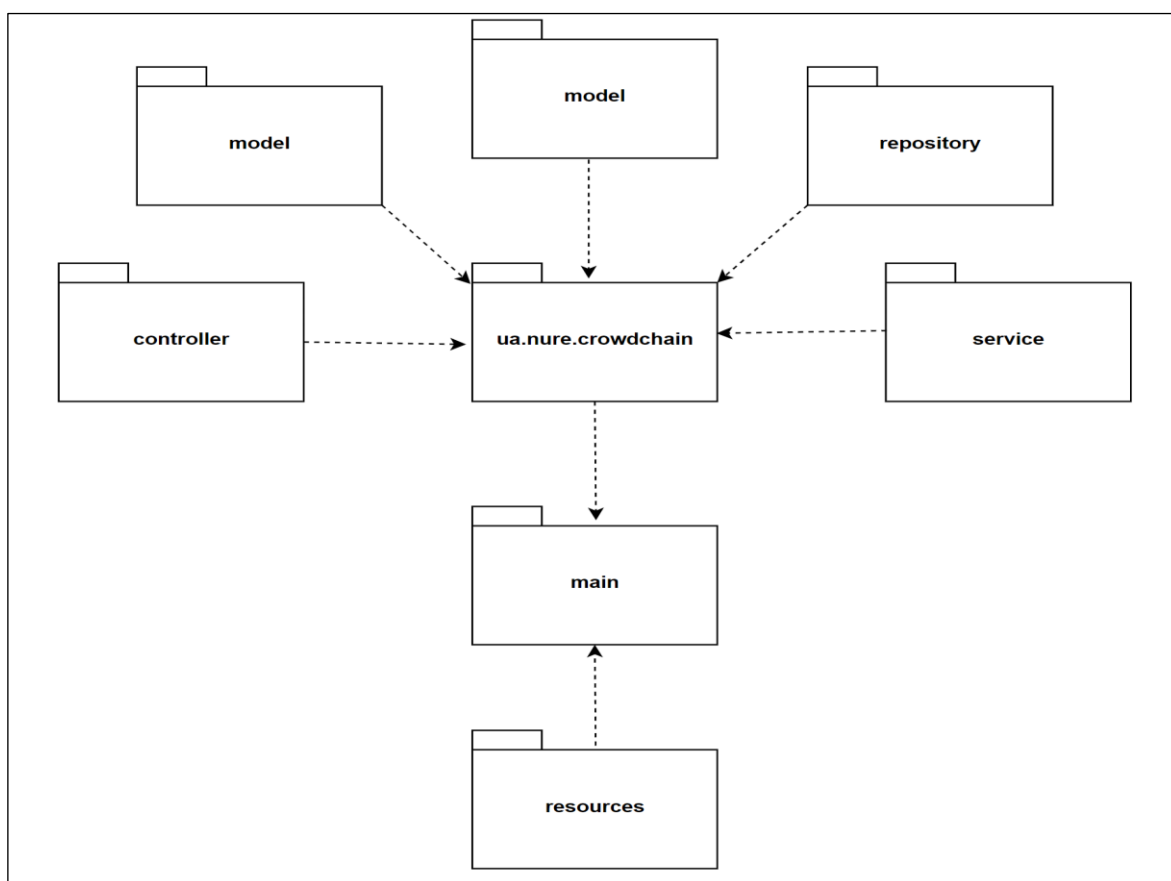


Рисунок 9 – Діаграма пакетів серверної частини (рисунок створено самостійно)

Також створено пакет `model`, який містить три класи, які відображають дані, що обробляються, та один, що відображає вузли блокчейну: `Project`, `Transaction`, `Customer`, `Miner`. У пакетах `repository`, `service`, `controller` створено відповідні класи для роботи з кожною із чотирьох моделей. У пакет `service` також помістили класи, пов'язані з машинним навчанням.

Четвертий рівень (`view`) винесено в окремий Angular-додаток, через який із системою взаємодіє більшість користувачів.

4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

4.1 Розробка блокчейну

Для зберігання даних транзакцій застосована технологія блокчейну. З цією метою було розроблено десктопний додаток, який після запуску на машині користувача підключається до сервера, з якого зчитує інформацію про вже існуючих учасників блокчейну. На цьому ж сервері знаходиться пул транзакцій, який учасники блокчейну намагаються зчитати кожні 10 секунд, поки не отримають непустий список. Після того, як нові транзакції були отримані, учасники розпочинають процес майнінгу, який завершується після видобутку валідного блоку. Нижче представлений код створення нового блоку:

```

while (!stopped.get()) {
    Block newBlock = new Block(currentBC);
    latestBlock.ifPresentOrElse(
        (block) -> newBlock.setPrevHash(block.getCurrHash()),
        () -> newBlock.setPrevHash(new byte[]{0}));
    newBlock.setTransactionLedger(new
ArrayList<>(newBlockTransactions));
    newBlock.setTimeStamp(LocalDateTime.now().toString());
    newBlock.setMinedBy(minersWallet.getPublicKey().getEncoded());
    newBlock.setCurrHash(new Crypto()
        .applySHA256(newBlock.toString(),
minersWallet.getPrivateKey()));
    if (validationService.isProved(newBlock, latestBlock,
latestPrevBlock)) {
        currentBC.add(newBlock);
        newBlockTransactions.clear();
        return newBlock;
    }
}

```

Валідність блоку визначається таким чином: якщо хеш попереднього і попередній хеш даного блоку не збігаються, то блок невалідний; якщо попередній блок був видобутий за більш ніж одну годину, то хеш даного блоку повинен містити на один ведучий нуль менше за хеш попереднього блоку; якщо попередній блок був видобутий за менш ніж 20 хвилин, то хеш даного блоку повинен містити на один ведучий нуль більше за хеш попереднього блоку; якщо попередній блок був видобутий за 20 – 60 хвилин, то хеш даного блоку повинен містити рівно стільки ж

ведучих нулів; якщо даний блок є першим або другим у ланцюжку, то його хеш повинен містити 4 ведучих нулів. Код перевірки валідності наданий нижче:

```

    if (!Arrays.equals(
        prevBlock.map(Block::getCurrHash).orElse(new byte[] {0}),
        block.getPrevHash()) || !crypto.isVerified(block)) {
        return false;
    }
    if (prevBlock.isPresent() && prevPrevBlock.isPresent()) {
        long difficulty =
LocalDateTime.parse(prevBlock.get().getTimeStamp()).getSecond() -
        LocalDateTime.parse(prevPrevBlock.get().getTimeStamp()).getSecond();
        if (difficulty > MAX_MINING_DIFFICULTY && getLeadingZerosLength(block)
- getLeadingZerosLength(prevBlock.get()) == 1) {
            return true;
        } else if (difficulty < MIN_MINING_DIFFICULTY &&
getLeadingZerosLength(block) - getLeadingZerosLength(prevBlock.get()) == 1)
        {
            return true;
        } else {
            return MIN_MINING_DIFFICULTY <= difficulty &&
difficulty <= MAX_MINING_DIFFICULTY && getLeadingZerosLength(block) ==
getLeadingZerosLength(prevBlock.get());
        } else {
            return getLeadingZerosLength(block) ==
START_ZEROS_QUANTITY;
        }
    }

```

Якщо валідність блоку не була підтверджена, видобуток продовжується. У протилежному випадку видобутий блок відправляється на сервер, щоб повідомити його про наявність найактуальнішої версії ланцюжка блоків, а також іншим учасникам. Після цього оновлюється баланс і список блоків у локальній базі даних. Далі викладено код опрацювання видобутого валідного блоку:

```

requestHandler.notifyServer(minedBlock);
new Thread(() -> requestHandler.notifyNodes(minedBlock)).start();
blockchainRepository.save(minedBlock);
walletRepository.updateBalance(
    walletRepository.getWallet().getBalance() +
        minedBlock.getTransactionLedger().stream()
            .mapToDouble(transaction ->
transaction.getAmount() * 0.015).sum());

```

Коли учасник блокчену приймає новий блок від іншого вузла, він перевіряє його на відповідність локальному ланцюжку за допомогою того самого алгоритму

перевірки валідності. Якщо валідність не була підтверджена, то у відповідь буде надісланий увесь локальний блокчейн. У протилежному випадку отриманий блок буде збережено, а його майнер отримає вищий рейтинг. Код опрацювання отриманого блоку представлений нижче:

```

Block receivedBlock = (Block) ois.readObject();
LinkedList<Block> currentBC = blockchainRepository.findAll();
    Optional<Block> currentLastBlock = currentBC.isEmpty()
        ? Optional.empty()
        : Optional.ofNullable(currentBC.getLast());
    Optional<Block> currentPrevBlock = currentBC.size() < 2
        ? Optional.empty()
        :
Optional.ofNullable(currentBC.get(currentBC.size() - 2));
    if (validationService.isProved(receivedBlock,
currentLastBlock, currentPrevBlock)) {
        blockchainRepository.save(receivedBlock);
        byte[] senderPK = receivedBlock.getMinedBy();
        Node node = nodeRepository.findById(senderPK)
            .orElse(new Node(senderPK,
socket.getInetAddress().getHostAddress(), socket.getPort(), 0));
        node.setScore(node.getScore() + 1);
        nodeRepository.save(node);
        oos.writeObject(new LinkedList<>());
    } else {
        oos.writeObject(currentBC);
    }
}

```

Коли майнер отримує у відповідь ланцюжок блоків, він порівнює його зі своїм локальним. Далі наведений код опрацювання отриманого блокчейну:

```

ObjectOutputStream oos = new
ObjectOutputStream(socket.getOutputStream());
ObjectInputStream ois = new
ObjectInputStream(socket.getInputStream());
oos.writeObject(minedBlock);
LinkedList<Block> returnedBC = (LinkedList<Block>) ois.readObject();
LinkedList<Block> currentBC =
BlockchainRepository.getInstance().findAll();
    if (!returnedBC.isEmpty() && !currentBC.equals(returnedBC)) {
        new Thread(() -> {
            if
(!currentBC.equals(ValidationService.getInstance().compareChains(currentBC,
returnedBC))) {
                BlockchainRepository.getInstance().removeAll();
                BlockchainRepository.getInstance().saveAll(returnedBC);
            })}.start();
    }
}

```

Порівняння двох ланцюжків відбувається так: спочатку порівнюється довжина обох ланцюжків і, якщо другий ланцюжок довший і валідний, то він є переможцем; якщо хоч одна умова не виконується, то перевіряється валідність першого блокчейну і, якщо вона підтверджується, то перший ланцюжок виграв; якщо ж і він є невалідним, то обидва ланцюжки признаються неправильними. Програмна реалізація порівняння ланцюжків надана нижче:

```
public LinkedList<Block> compareChains(LinkedList<Block> firstChain,
LinkedList<Block> secondChain) {
    if (firstChain.size() < secondChain.size() &&
isProved(secondChain)) {
        System.out.println("Second chain won");
        return secondChain;
    } else if (isProved(firstChain)) {
        System.out.println("First chain won");
        return firstChain;
    } else {
        System.out.println("Both chains are invalid");
        return new LinkedList<>();
    }
}
```

Ланцюжок визнається валідним тільки у тому випадку, якщо кожен його окремий блок є валідним. Метод перевірки валідності реалізовано таким чином:

```
public boolean isProved(LinkedList<Block> blockChain) {
    try {
        for (int i = 0; i < blockChain.size(); i++) {
            Optional<Block> prevBlock = i > 0 ?
Optional.of(blockChain.get(i - 1)) : Optional.empty();
            Optional<Block> prevPrevBlock = i > 1 ?
Optional.of(blockChain.get(i - 2)) : Optional.empty();
            if (!isProved(blockChain.get(i), prevBlock,
prevPrevBlock)) {
                return false;
            }
        }
    } catch (Exception e) {
        System.err.println("Problem with blockchain validation: "
+ e.getMessage());
        e.printStackTrace();
        return false;
    }
    return true;
}
```

Кожен вузол блокчейну надає серверу API для отримання зібраної суми на певний проєкт за його ідентифікатором, реалізація представлена наступним кодом:

```

server = HttpServer.create(new InetSocketAddress(8080), 0);
server.createContext("/api/transactions", exchange -> {
String queryParams = exchange.getRequestURI().getQuery();
StringBuilder body = new StringBuilder();
if (queryParams != null && queryParams.startsWith("to")) {
    System.out.println("Query param: " + queryParams);
    String receiver = queryParams.replace("to=", "");
    body.append(transactionRepository.findAmountSumByReceiver(receiver.getBytes()));
}
exchange.sendResponseHeaders(200, body.toString().getBytes().length);
exchange.getResponseBody().write(body.toString().getBytes());
exchange.getResponseBody().close();
});

public Double findAmountSumByReceiver(byte[] to) {
    try (DatabaseHandler dbHandler = new DatabaseHandler(BLOCKCHAIN_CONNECTION_URL)) {
        ResultSet resultSet = dbHandler.runScript("SELECT SUM(AMOUNT) FROM TRANSACTIONS WHERE \"TO\" = ?", new LinkedList<>() {{
            add(Pair.with(to, JDBCType.VARCHAR));
        }});
        return resultSet.getDouble(1);
    } catch (Exception e) {
        System.err.println("Problem while finding all transactions by ledger id: " + e.getMessage());
        e.printStackTrace();
    }
    return 0D;
}

```

У якості локальної бази даних використовується СКБД SQLite. Звернення до неї відбувається через інтерфейс JDBC у методі runScript класу DatabaseHandler. Для цього при створенні екземпляру класу встановлюється з'єднання з файлом «blockchain.db». При виконанні методу створюється екземпляр класу Statement, якщо запит не містить параметрів, або екземпляр класу PreparedStatement, якщо запит містить параметри. Таке розділення необхідне для того, щоб унеможливити SQL-ін'єкції [14], але водночас зберегти швидкодію програми. Далі для кожного параметру перевіряється його тип, а значення передається до виконання запиту. Після виконання, якщо результат містить дані, то вони зберігаються у полі resultSet.

Так як результат містить дані тільки після команди «Select», то перевірка стосується саме початку рядка запиту (див. додаток В).

Інтерфейс додатку розроблено за допомогою бібліотеки JavaFX. Він надає таблицю для перегляду історії переказів у системі та кнопку для оновлення даних у таблиці, якщо вони застаріли (див. додаток В).

У результаті користувач завжди може побачити найбільш актуальні дані у своєму додатку (див. рис. 10).

Timestamp	Amount	Sender	Recipient	Type
2021-01-01 12:0...	10.0	Alice	Bob	DONATE

Рисунок 10 – Інтерфейс додатку для підтримки блокчейну (рисунок створено самостійно)

4.2 Розробка клієнтської частини

Інтерфейс самої платформи представлений браузерним додатком, розробленим із використанням мови програмування TypeScript і фреймворку Angular. Його бібліотека «Angular Material» надає великий набір готових компонентів, які зручно використовувати у клієнтському додатку.

```

<mat-sidenav-container class="main-container">
  <mat-sidenav-content class="sidenav-content">
    <mat-toolbar class="nav-container" color="primary">
      <div fxFlex fxLayout fxLayoutAlign="start center">
        <div>

```

```
        <button mat-button routerLink="/">Home</button>
        <button mat-button routerLink="/projects">Projects</button>
    </div>
    <div fxFlex></div>
    <div><button mat-button>Sign In</button></div>
</div>
</mat-toolbar>
<main>
    <router-outlet></router-outlet>
</main>
</mat-sidenav-content>
</mat-sidenav-container>
```

Такий підхід дозволяє будувати браузерні програми типу Single Page Application, тобто додатки з однією сторінкою, контент якої динамічно змінюється. На рисунках 11 і 12 зображений різний вміст цієї сторінки.

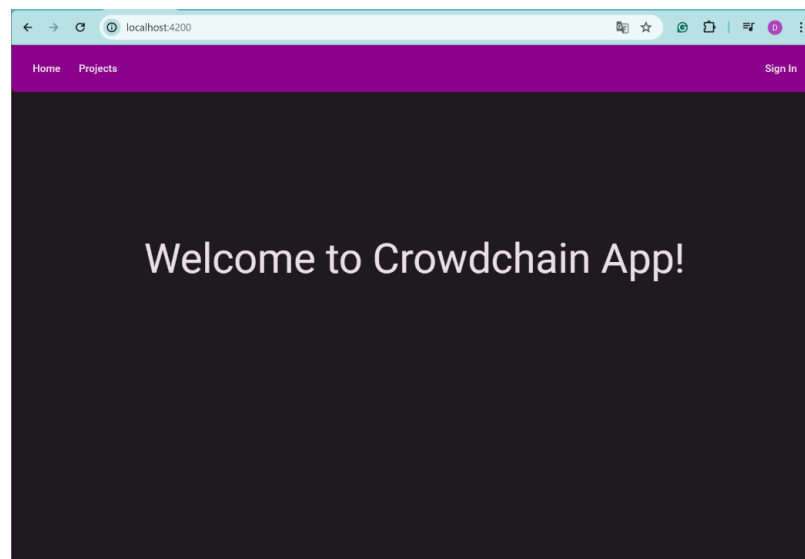


Рисунок 11 – Домашня сторінка платформи (рисунок створено самостійно)

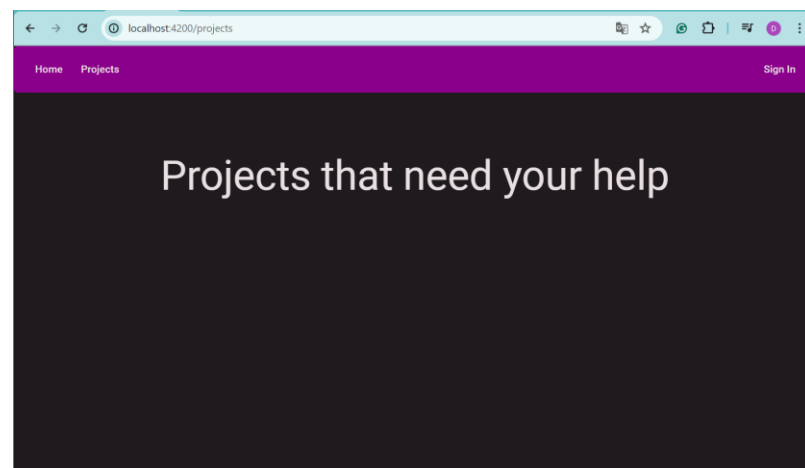


Рисунок 12 – Сторінка проєктів (рисунок створено самостійно)

Цей додаток взаємодіє з серверною частиною за допомогою класу `HttpClient` з бібліотеки `@angular/common/http`. Налаштування взаємодії з сервером виконане в наступному методі:

```
export class ProjectService {
  private apiUrl = `http://${SERVER_URL}:9000/api/projects`
  constructor(private http: HttpClient) { }
  getProject(id: string): Observable<any> {
    console.log(id);
    const url = `${this.apiUrl}/${id}`;
    console.log(url);
    return this.http.get<any>(url);
  }

  getProjects(): Observable<any[]> {
    return this.http.get<any[]>(this.apiUrl);
  }
}
```

4.3 Розробка серверної частини

Клієнтський додаток відправляє запити до шлюзового мікросервіса, який перенаправляє їх до іншого відповідного мікросервіса. Налаштування шлюзового сервісу наведено нижче:

```
spring:
  cloud:
    gateway:
      mvc:
        routes:
          - id: project-service-api
            uri: ${project-service.url}
            predicates:
              - Path=/api/projects/**
          - id: blockchain-service-api
            uri: ${blockchain-service.url}
            predicates:
              - Path=/api/donate
          - id: success-prediction-api
            uri: ${success-prediction-service.url}
            predicates:
              - Path=/api/**
```

Усі мікросервіси побудовані на базі Spring Boot – потужного фреймворку для створення автономних серверних додатків, написаних мовою Java.

5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Мануальне тестування

Для тестування програмного забезпечення застосовувався мануальний підхід. Він дозволив перевірити функціонування системи з точки зору користувача, оцінити її зручність, а також взаємодію всіх складових. Щоб провести таке тестування, потрібно запустити всі компоненти забезпечення та перейти на сторінку проєктів (див. рис. 13).

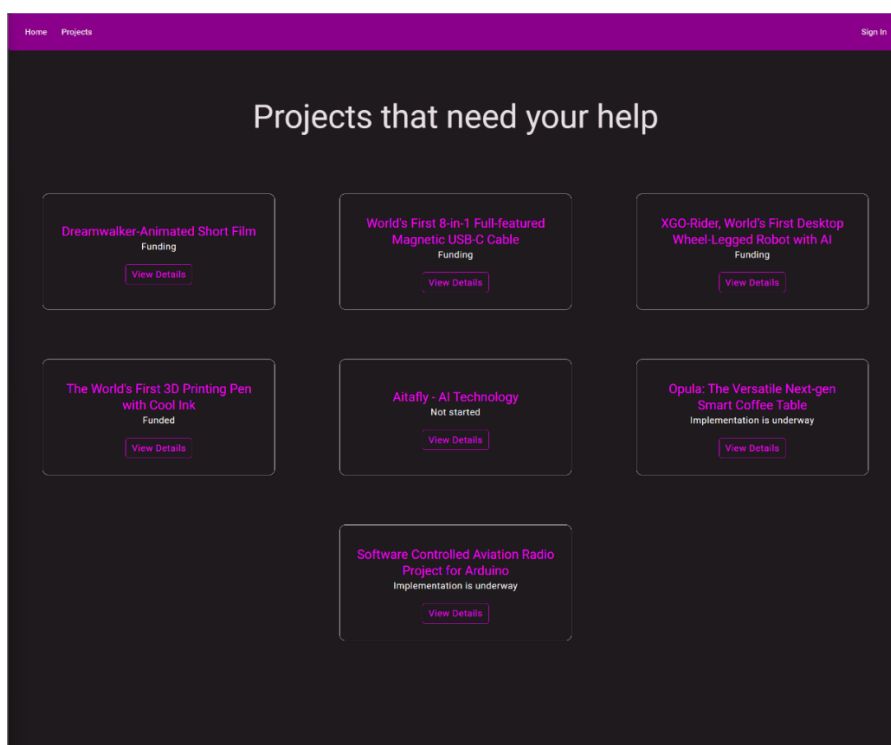


Рисунок 13 – Завантажені проєкти (рисунок створено самостійно)

Якщо програмне забезпечення запустилося без проблем, з'являться «картки» проєктів із назвами та статусами. Тепер можна обрати проєкт зі списку та натиснути кнопку «View Details». На відкритій сторінці видно також опис проєкту, ціль збору, зібрану суму та 2 кнопки: «Success probability» – для підрахунку ймовірності успіху та «Donate» – для переходу на сторінку введення платіжних даних (див. рис. 14).

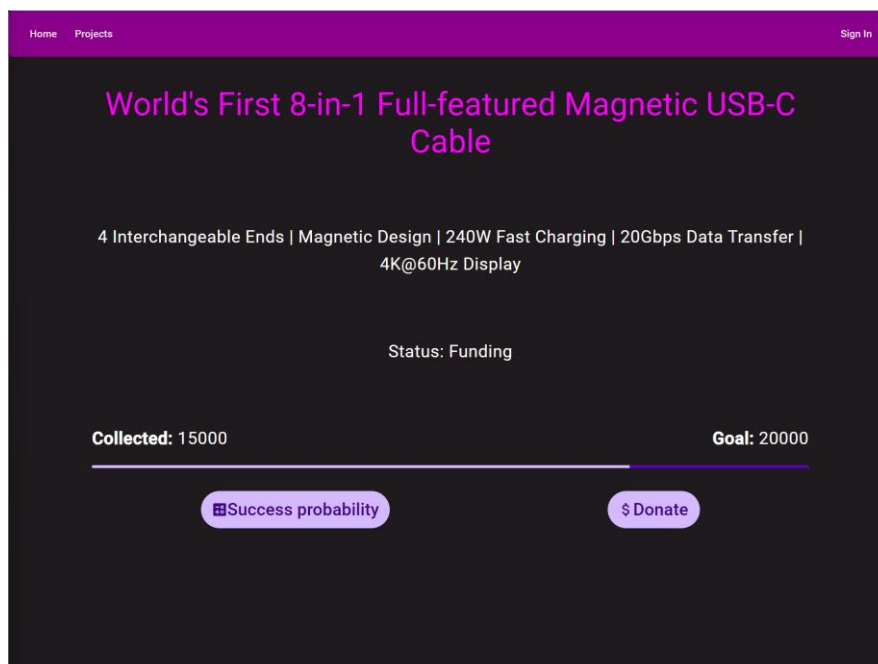


Рисунок 14 – Сторінка проєкту (рисунок створено самостійно)

Якщо натиснути на кнопку «Success probability», буде надіслано запит до мікросервіса прогнозування успіху. Якщо серверна частина відпрацювала без помилок, отримане значення буде виведено знизу, а кнопка «Success probability» зникне (див. рис. 15).

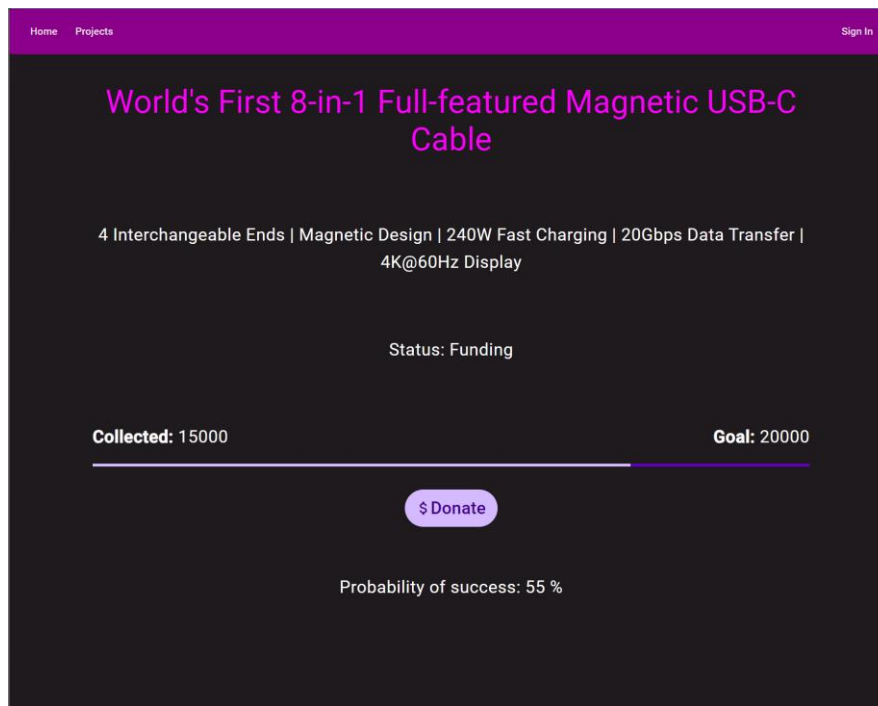


Рисунок 15 – Відображення ймовірності успіху (рисунок створено самостійно)

Якщо натиснути на кнопку «Donate» відкриється форма введення платіжних даних (див. рис. 16) з відключеною кнопкою «Donate».

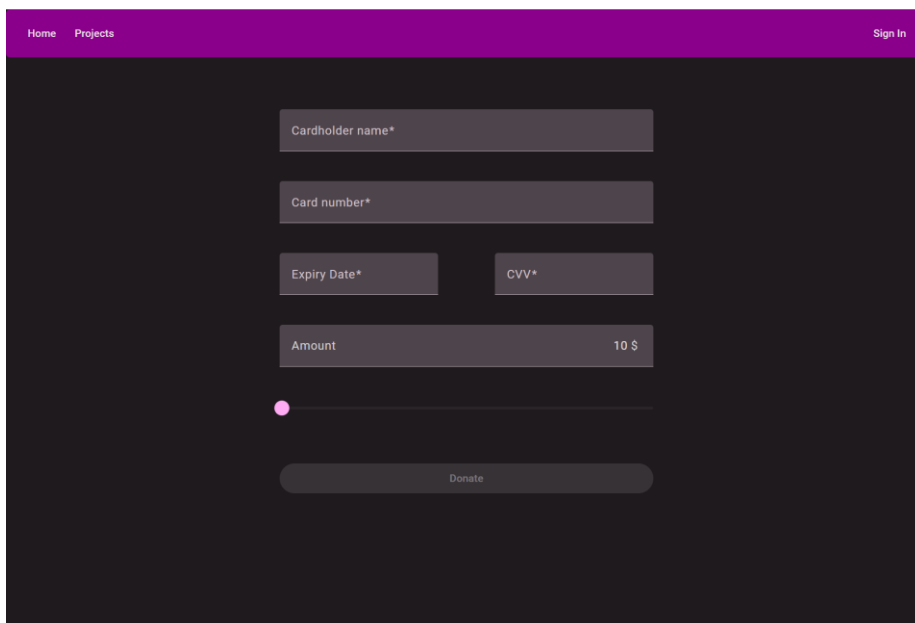
The image shows a dark-themed web form for entering payment information. At the top, there is a purple navigation bar with 'Home' and 'Projects' on the left, and 'Sign In' on the right. The form fields are: 'Cardholder name*' (empty), 'Card number*' (empty), 'Expiry Date*' (empty), 'CVV*' (empty), and 'Amount' (set to '10 \$'). Below the 'Amount' field is a horizontal progress indicator with a single purple dot on the left. At the bottom of the form is a grey 'Donate' button that is disabled.

Рисунок 16 – Форма введення платіжних даних (рисунок створено самостійно)

Якщо ввести дані у неправильному форматі, кнопка так і залишиться відключеною, а біля полів з'являться повідомлення про помилки (див. рис. 17).

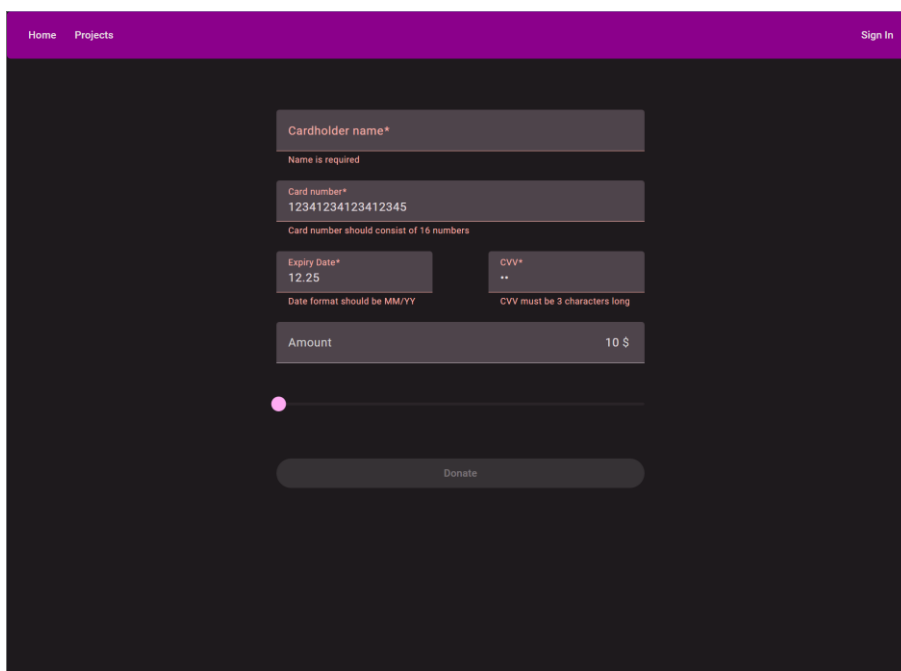
This image shows the same donation form as in Figure 16, but with error messages. The 'Cardholder name*' field has the text 'Name is required' below it. The 'Card number*' field contains '12341234123412345' and has the message 'Card number should consist of 16 numbers' below it. The 'Expiry Date*' field contains '12.25' and has the message 'Date format should be MM/YY' below it. The 'CVV*' field contains '..' and has the message 'CVV must be 3 characters long' below it. The 'Amount' field remains '10 \$'. The 'Donate' button is still disabled.

Рисунок 17 – Форма після введення неправильних даних (рисунок створено самостійно)

Якщо ввести правильні дані, то кнопка стане активною (див. рис. 18).

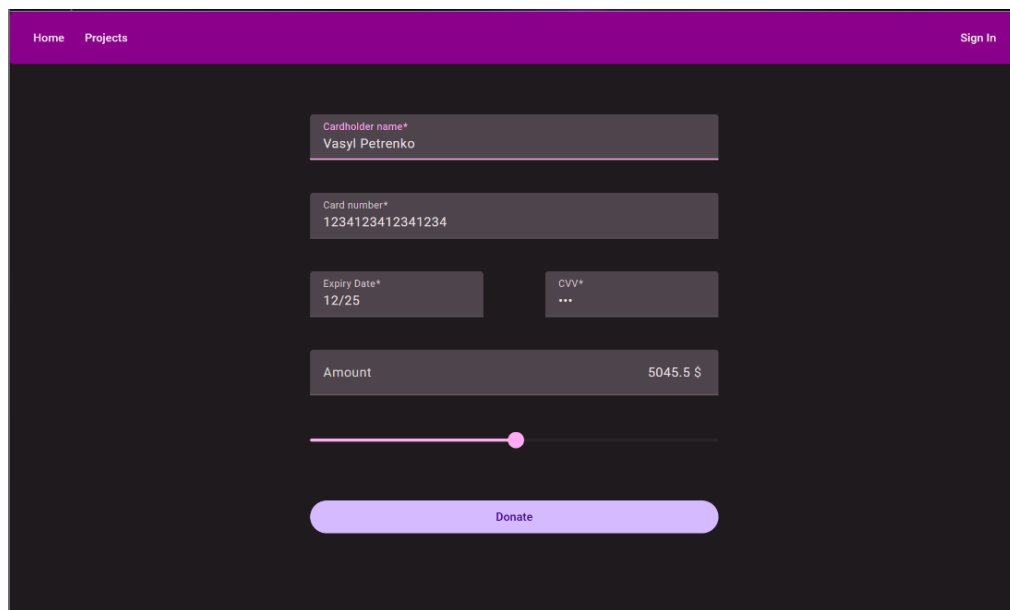
A screenshot of a web application's donation form. The form is set against a dark background with a purple header. The header contains 'Home' and 'Projects' on the left and 'Sign In' on the right. The form fields are: 'Cardholder name*' with the value 'Vasyl Petrenko', 'Card number*' with '1234123412341234', 'Expiry Date*' with '12/25', and 'CVV*' with '***'. Below these is an 'Amount' field showing '5045.5 \$' with a slider control. At the bottom is a purple 'Donate' button.

Рисунок 18 – Форма після введення правильних даних (рисунок створено самостійно)

Тепер можна натиснути на кнопку «Donate» і дані з форми будуть надіслані до мікросервісу блокчейну, де з цих даних буде зформована транзакція, яка потрапить до пулу транзакцій, а згодом і до самого блокчейну. Користувач при цьому буде перенаправлений на сторінку подяки (див. рис. 19).

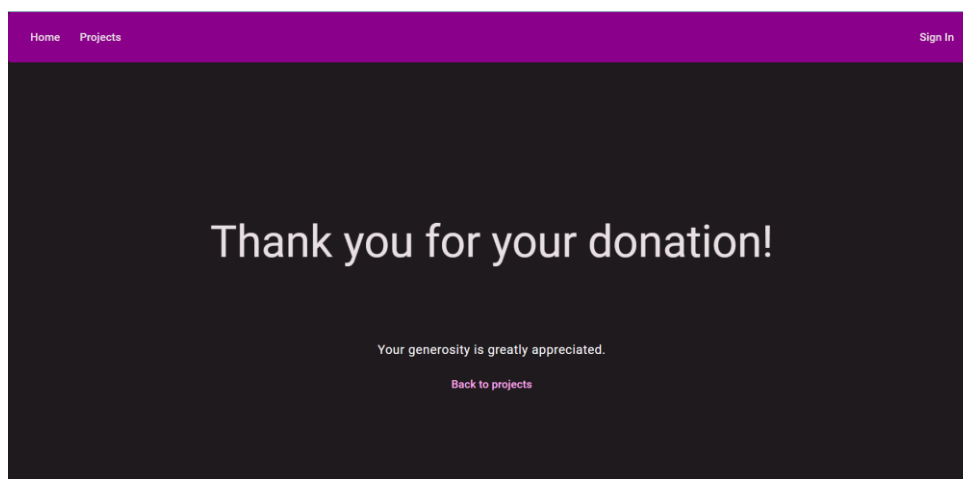


Рисунок 19 – Сторінка подяки (рисунок створено самостійно)

Тепер якщо знову відкрити сторінку цього проєкту, можна зауважити, що кількість зібраних коштів і статус проєкту оновилися (див. рис. 20).

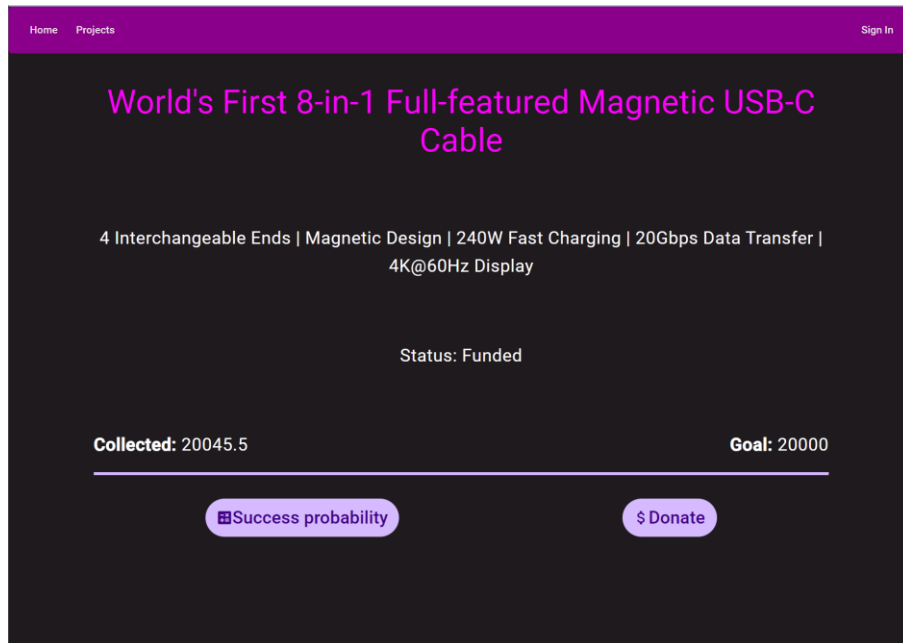


Рисунок 20 – Оновлена сторінка проєкту

Також, у таблиці додатку для підтримки роботи блокчейну з'явилася нова транзакція (див. рис. 21).

Block 20

Timestamp	Amount	Sender	Recipient	Type
2024-06-06T01:49:13.426...	5045.5	Vasyl Petrenko	World's First 8-in-1 Full-featured Magnetic USB-C Cable	DONATE

20/20

Рисунок 21 – Оновлений блокчейн (рисунок створено самостійно)

Таким чином було протестовано основний сценарій користувача. Система поведилася так, як від неї очікувалося, недоліків не виявлено.

5.2 Модульне тестування

Мікросервіс блокчейну – це головна сполучна ланка системи, тому він був також окремо перевірений методами модульного тестування. За допомогою бібліотек Jupiter і Assertions були протестовані клас контролера та клас сервісу. Нижче наведені тести.

```

@SpringBootTest
public class ControllerTest {
    @MockBean
    private PaymentService paymentService;
    @Autowired
    private PaymentController paymentController;

    @Test
    public void testPaymentServiceSuccess() {
        Mockito.when(paymentService.service(any(TransactionRequest.class))).t
henReturn(new Transaction());
        ResponseEntity<ResponseMessage> response =
paymentController.donate(new TransactionRequest());
        assertThat(response.getBody().getMessage()).isEqualTo("Transaction was
added to transaction pool");
    }

    @Test
    public void testPaymentServiceFailure() {
        Mockito.when(paymentService.service(any(TransactionRequest.class))).t
henReturn(null);
        ResponseEntity<ResponseMessage> response =
paymentController.donate(new TransactionRequest());
        assertThat(response.getBody().getMessage()).isEqualTo("Transaction was
not added to transaction pool");
    }
}

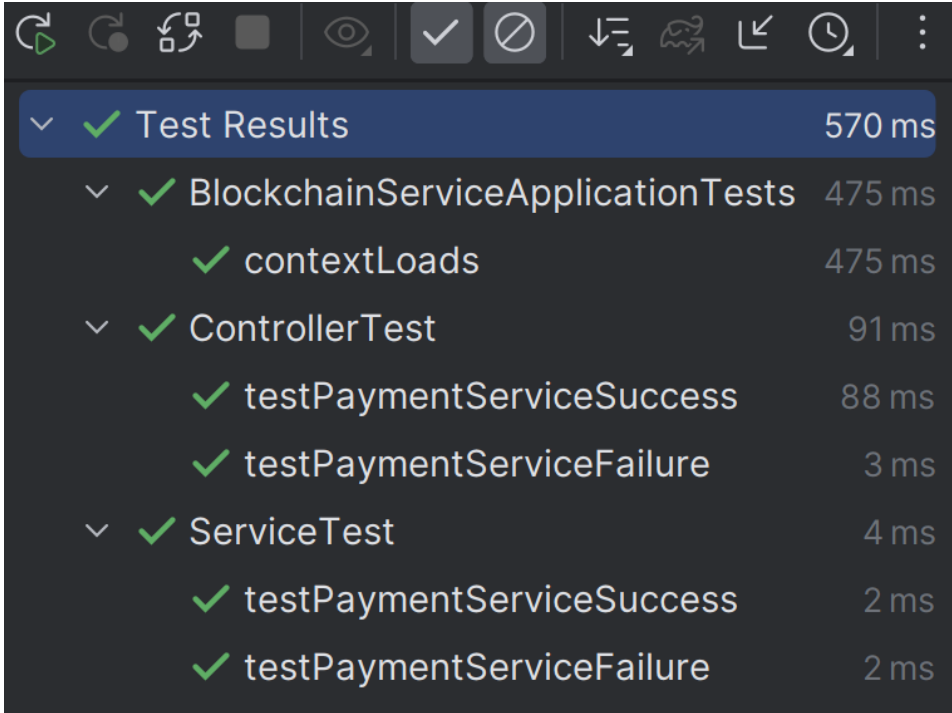
@SpringBootTest
public class ServiceTest {
    @Autowired
    private PaymentService paymentService;

    @Test
    public void testPaymentServiceSuccess() {
        assertThat(paymentService.service(new
TransactionRequest())).isEqualTo(new Transaction());
    }

    @Test
    public void testPaymentServiceFailure() {
        assertThat(paymentService.service(null)).isNull();
    }
}

```

Результати модульного тестування наведені на рисунку 22.



Test Category	Test Name	Execution Time (ms)
Test Results		570 ms
BlockchainServiceApplicationTests	contextLoads	475 ms
ControllerTest	testPaymentServiceSuccess	91 ms
	testPaymentServiceFailure	88 ms
	testPaymentServiceFailure	3 ms
ServiceTest	testPaymentServiceSuccess	4 ms
	testPaymentServiceSuccess	2 ms
	testPaymentServiceFailure	2 ms

Рисунок 22 – Результати модульного тестування мікросервісу блокчейну (рисунок створено самостійно)

Таким чином було окремо перевірено кожен значущий елемент мікросервісу та доведено його відповідність заданим сценаріям.

Отже, перевірка системи показала її спроможність відповідати на виклики та потреби користувачів, а виявлені в процесі тестування швидкодія, зручність і прозорість дозволяють говорити про деяку перевагу над конкурентами.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було проаналізовано предметну галузь і доцільність розробки «Краудфандингова платформа на основі технології Blockchain з можливістю передбачення успішності проєктів за допомогою машинного навчання», визначено цілі та засоби роботи, сформульовано вимоги до програмної системи, спроектовано архітектуру платформи, встановлено зв'язок з іншими дослідженнями з теми краудфандингу, блокчейну та штучного інтелекту.

Розроблено платформу, яка дозволить підприємцям-початківцям швидко збирати кошти для реалізації своїх ідей. При цьому будь-який інвестор зможе зручно надіслати гроші, будучи впевненим, що вони підуть саме визначеному адресату, адже інформацію про свої транзакції може отримати кожен користувач, який завантажить блокчейн на свій комп'ютер [15, с. 2]. Окрім того, що інвестори таким чином захищають свої інвестиції, вони також отримують частину комісії за усі нові транзакції, які їм вдалося помістити до блокчейну. Більш того, технологія блокчейну надасть змогу гарантовано отримати свої гроші назад у випадку, якщо необхідна сума не була зібрана. Також, усі користувачі мають змогу отримати інформацію про вірогідність успіху проєкту, визначену штучним інтелектом, який буде самостійно навчатися на даних, які зберігаються в системі. Передбачається, що на основі цієї інформації будуть прийматися рішення про доцільність збору та передачі коштів.

Інтеграція блокчейн-складової та машинного навчання у сферу групового фінансування проєктів сприяє зростанню цієї індустрії, зниженню ризиків для інвесторів і робить використання коштів у суспільстві більш ефективним і прозорим. При цьому вказані технології не накладають обмежень на галузі залучення коштів чи теми фінансованих проєктів.

Отже, розробка платформи цілком виправдана та потребує дуже малих ресурсів порівняно з користю, яку вона принесе.

Матеріали кваліфікаційної роботи пройшли апробацію на науково-технічній конференції (див. додаток Ж).

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Жмур Д. М. Краудфандингова платформа на основі технології блокчейн із можливістю передбачення успішності проектів за допомогою машинного навчання. Інформаційні інтелектуальні системи : матеріали XXVIII міжнар. молодіж. форуму «Радіоелектроніка та молодь у XXI столітті», [Харків], 16–18 квіт. 2024 р. / Харківський національний університет радіоелектроніки. – Харків, 2024. – С. 112–113.
2. Що таке Web 3.0 та чим воно важливо? Binance Academy. URL: <https://academy.binance.com/uk/articles/the-evolution-of-the-internet-web-3-0-explained> (date of access: 02.03.2024).
3. Ко С. Н., Cheng M. Y. Dynamic Prediction of Project Success Using Artificial Intelligence. Journal of Construction Engineering and Management. 2007. P. 316-321. DOI:10.1061/(ASCE)0733-9364(2007)133:4(316) (date of access: 02.03.2024).
4. Teotia A. Prediction of Crowdfunding Project Success Probability using Machine Learning. 2019.
5. Kickstarter. Wikipedia. URL: <https://uk.wikipedia.org/wiki/Kickstarter> (date of access: 25.05.2024).
6. Kickstarter [Електронний ресурс] – URL: <https://www.kickstarter.com/> (date of access: 25.05.2024).
7. Tecra Space [Електронний ресурс] – URL: <https://tecra.space> (date of access: 13.05.2024).
8. Fundly [Електронний ресурс] – URL: <https://fundly.com/> (data of access: 25.05.2024).
9. GoFundMe. Wikipedia. URL: <https://uk.wikipedia.org/wiki/GoFundMe> (date of access: 25.05.2024).
10. GoFundMe [Електронний ресурс] – URL: <https://www.gofundme.com/> (date of access: 25.05.2024).
11. Indiegogo [Електронний ресурс] – URL: <https://www.indiegogo.com/> (date of access: 25.05.2024).

12. Терещенко Г. Ю. Using blockchain technology in international bussiness relationships. 23-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті», Збірник Матеріалів форуму., т. 10. Харків, 2019. С. 91–92.

13. Li Y., Reddy C. K., Rakesh V. Project Success Prediction in Crowdfunding Environments. ResearchGate. URL: https://www.researchgate.net/publication/310820239_Project_Success_Prediction_in_Crowdfunding_Environments (date of access: 05.03.2024).

14. Кириченко І., Калашников П. Захист веб-додатків від SQL-ін'єкцій. Science in the Environment of Rapid Changes : Scientific and Practical Conference, Brussels, 6–8 February 2023. P. 335–337.

15. Buzharovski S. Introducing Blockchain with Java: Program, Implement, and Extend Blockchains with Java. Apress Media LLC, 2022. 173 p.

ДОДАТОК А

Звіт результатів перевірки унікальності тексту в базі ХНУРЕ



Ім'я користувача:
Олійник Олена Володимирівна каф. ПІ

ID перевірки:
1016336034

Дата перевірки:
08.06.2024 19:04:47 EEST

Тип перевірки:
Doc vs Library

Дата звіту:
09.06.2024 08:21:17 EEST

ID користувача:
100012353

Назва документа: 2024_Б_ПІ_ПЗПІ_20_10_Жмур_Д_М_скорочений

Кількість сторінок: 46 Кількість слів: 7297 Кількість символів: 62950 Розмір файлу: 1.64 MB ID файлу: 1016136780

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

3.55%
Схожість

Найбільша схожість: 1.47% з джерелом з Бібліотеки (ID файлу: 1008214841)

Пошук збігів з Інтернетом не проводився

3.55% Джерела з Бібліотеки

315

Сторінка 48

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

9

Підозріле форматування

17
сторінок

ДОДАТОК Б

Перелік джерел посилання за науковими напрямками керівника та науковців
кафедри програмної інженерії

1. Жмур Д. М. Краудфандингова платформа на основі технології блокчейн із можливістю передбачення успішності проєктів за допомогою машинного навчання. Інформаційні інтелектуальні системи : матеріали XXVIII міжнар. молодіж. форуму «Радіоелектроніка та молодь у XXI столітті», [Харків], 16–18 квіт. 2024 р. / Харківський національний університет радіоелектроніки. – Харків, 2024. – С. 112–113.

12. Терещенко Г. Ю. Using blockchain technology in international bussiness relationships. 23-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті», Збірник Матеріалів форуму., т. 10. Харків, 2019. С. 91–92.

14. Кириченко І., Калашников П. Захист веб-додатків від SQL-ін'єкцій. Science in the Environment of Rapid Changes: Scientific and Practical Conference, Brussels, 6–8 February 2023. P. 335–337.

ДОДАТОК В

Приклад кодів програми

В.1 Лістинг коду для доступу до бази даних

```

public ResultSet runScript(String sql, LinkedList<Pair<Object,
SQLType>> args) throws SQLException {
    preparedStatement = connection.prepareStatement(sql);
    int counter = 1;
    for (Pair<Object, SQLType> entry : args) {
        switch (entry.getValue1()) {
            case BLOB:
                preparedStatement.setBytes(counter++,      (byte[])
entry.getValue0());
                break;
            case INTEGER:
                preparedStatement.setInt(counter++,      (int)
entry.getValue0());
                break;
            case DOUBLE:
                preparedStatement.setDouble(counter++,    (double)
entry.getValue0());
                break;
            case VARCHAR:
                preparedStatement.setString(counter++,    (String)
entry.getValue0());
                break;
            default:
                throw new SQLException("Unknown SQLType");
        }
    }

    if (sql.trim().startsWith("SELECT")) {
        resultSet = preparedStatement.executeQuery();
    } else {
        preparedStatement.executeUpdate();
    }

    return resultSet;
}

public ResultSet runScript(String sql) throws SQLException {
    statement = connection.createStatement();

    if (sql.trim().startsWith("SELECT")) {
        resultSet = statement.executeQuery(sql);
    } else {
        statement.executeUpdate(sql);
    }

    return resultSet;
}

```

В.2 Лістинг коду інтерфейсу додатку для підтримки блокчейну

```

<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Pagination?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<VBox      prefHeight="501.0"      prefWidth="503.0"      spacing="10"
xmlns="http://javafx.com/javafx/21"      xmlns:fx="http://javafx.com/fxml/1"
fx:controller="ua.nure.crowdchainnode.controller.TransactionsViewController
">

    <Pagination fx:id="blockPagination" />

    <TableView fx:id="transactionsTable">
        <columns>
            <TableColumn      fx:id="timestampColumn"
prefWidth="106.00000250339508" text="Timestamp" />
            <TableColumn      fx:id="amountColumn"
prefWidth="53.3333740234375" text="Amount" />
            <TableColumn      fx:id="senderColumn"      prefWidth="108.0"
text="Sender" />
            <TableColumn      fx:id="recipientColumn"   prefWidth="160.0"
text="Recipient" />
            <TableColumn      fx:id="typeColumn"   minWidth="6.666748046875"
prefWidth="75.33349609375" text="Type" />
        </columns>
    </TableView>

    <Button layoutY="54.0" mnemonicParsing="false" onAction="#refresh"
prefHeight="36.0" prefWidth="96.0" text="REFRESH" textFill="#00e134">
        <font>
            <Font name="System Bold" size="16.0" />
        </font>
        <VBox.margin>
            <Insets left="205.0" />
        </VBox.margin>
    </Button>
</VBox>

```

ДОДАТОК Г

Специфікація програмного продукту

1 ВСТУП

1.1 Огляд продукту

Краудфандингова онлайн-платформа на основі блокчейн-технологій, яка надає можливість користувачам створювати, переглядати та фінансувати проекти, переглядати історію кампанії збору коштів. Додатково система використовує машинне навчання для передбачення успішності проектів.

1.2 Мета

Забезпечити прозорий, надійний та безпечний спосіб залучення фінансування для проектів з передбаченням їхньої успішності.

1.3 Межі

Система розроблена для інтернет-користувачів, що бажають залучити або вкласти кошти у реалізацію інноваційних ідей. Вона не підтримує офлайн-транзакції.

1.4 Посилання

- [Blockchain technology] (<https://www.blockchain.com/>);
- [Machine Learning] (https://en.wikipedia.org/wiki/Machine_learning);
- [Crowdfunding] (<https://osvita.diia.gov.ua/courses/crowdfunding-how-to-raise-money-for-your-project>).

1.5 Означення та аббревіатури

- Blockchain – технологія розподіленого реєстру на основі ланцюжка блоків;
- ML (Machine Learning, машинне навчання) – це галузь штучного інтелекту (ШІ) та інформатики, яка зосереджена на використанні даних і

алгоритмів, щоб дозволити ШІ імітувати спосіб навчання людей, поступово підвищуючи його точність;

- API – інтерфейс програмування застосунків;
- Краудфандинг (від англ. Crowdfunding) – це співпраця людей, які добровільно об'єднують свої гроші чи інші ресурси, щоб підтримати зусилля інших людей або організацій.

2 ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

Платформа дозволить залучити більше користувачів завдяки прозорості та безпеці блокчейну, а також покращити ефективність використання коштів завдяки прогнозам успішності на основі машинного навчання.

2.2 Функції продукту

- створення та управління проєктами;
- фінансування проєктів;
- прогнозування ймовірності успіху;
- надання історії кампанії та інших даних проєктів.

2.3 Характеристики користувачів

- стартапери;
- інвестори.

2.4 Загальні обмеження

2.4.1 Операційне середовище

Клієнтська частина розроблена у середовищі розробки WebStorm 2024.1. Серверні та десктопний додатки написані з використанням середовища IntelliJ IDEA 2023.2.2. Взаємодія з базою даних під час розробки виконувалась через платформу Docker.

2.4.2 Технологія розробки

TypeScript, Angular, Java, Spring, JavaFX.

2.5 Припущення й залежності

Припущення:

- розроблена платформа буде мати більше довіру та впевненість інвесторів
- розроблена платформа збільшить кількість стартапів у світі

Залежності:

- використання стабільного інтернет-з'єднання;
- використання потужного комп'ютера для доступу до блокчейну.

3 КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача

- інтуїтивно зрозумілий та зручний у використанні;
- наявність панелі управління для авторів проєктів та інвесторів.

3.1.2 Апаратний інтерфейс

- підтримка десктопних пристроїв.

3.1.3 Програмний інтерфейс

- RESTful API для інтеграції з іншими системами.

3.1.4 Комунікаційний протокол

- HTTP/HTTPS.

3.1.5 Обмеження пам'яті

- щонайменше 32 ГБ оперативної пам'яті для доступу до блокчейну.

3.1.6 Операції

- реєстрація та аутентифікація користувачів;
- створення, управління та перегляд проєктів;
- здійснення фінансових транзакцій;
- перегляд історії кампанії і ймовірності успіху кожного проєкту.

3.1.7 Припущення й залежності

- сучасний процесор. Додаток для підтримки блокчейну повинен запускатися на системі, яка має щонайменше 6 логічних ядер (потоків);
 - містка ОЗП. Додаток має коректно працювати на системі, яка має щонайменше 32 ГБ оперативної пам'яті. Якщо, пам'ять буде не настільки містка, вона швидко почне переповнюватися даними блоків і викидати OutOfMemoryError;
 - швидкий Інтернет. Додаток повинен мати доступ до Інтернету на швидкості щонайменше 80 Мбіт/с. В іншому випадку він теж буде перешкоджати роботі блокчейну, тому що інші учасники будуть дуже довго чекати відповіді від нього. Система не повинна надавати можливість доєднатися до блокчейну з повільнішим Інтернетом.

3.2 Атрибути програмного продукту

3.2.1 Надійність

- система повинна забезпечувати безперебійну роботу 99.9% часу.

3.2.2 Доступність

- система має бути доступна з будь-якого місця в будь-який час;
- час відповіді від системи має складати не більше 3-х секунд.

3.3.3 Безпека

- усі блоки повинні бути захищені цифровим підписом;
- доступ до конфіденційних даних повинен бути обмежений;

- неможливість внесення змін до історії кампанії збору коштів повинна прагнути до безкінечності

3.3.4 Супроводжуваність

- можливість просто та швидко додавати нові функції та виправляти помилки.

3.3.5 Переносимість

- сумісність з різними операційними системами та браузерами.

3.3.6 Продуктивність

- швидке завантаження сторінок та обробка транзакцій.

3.4 Вимоги бази даних

- розподілена база даних для зберігання транзакцій;
- реляційна база для зберігання інформації про проекти та їхніх авторів, щоб зменшити займаний простір;
- підтримка високого рівня доступності та надійності бази даних для зберігання інформації про блокчейн.

3.5 Інші вимоги

- відповідність правовим нормам та регуляціям, що діють у країнах, де використовується система.

ДОДАТОК Д

Тези XXVIII міжнародного молодіжного форуму «Радіоелектроніка та молодь у
XXI столітті»



УДК 004.75:005.8]:004.85

DOI: <https://doi.org/10.30837/UYF.IIS.2024.112>**КРАУДФАНДИНГОВА ПЛАТФОРМА НА ОСНОВІ ТЕХНОЛОГІЇ БЛОКЧЕЙН ІЗ МОЖЛИВІСТЮ ПЕРЕДБАЧЕННЯ УСПІШНОСТІ ПРОЄКТІВ ЗА ДОПОМОГОЮ МАШИННОГО НАВЧАННЯ**

Жмур Д. М.

Науковий керівник – к.т.н., доц. Кириченко І. В.

Харківський національний університет радіоелектроніки, каф. ПІ,
м. Харків, Українаe-mail: dmytro.zhmur@nure.ua

This work is devoted to modernization of crowdfunding process. The most important problems and vulnerabilities of current crowdfunding were analyzed. The most significant reasons for mistrust of existing crowdfunding platforms were determined. The main non-compliance with the Web 3.0 was considered. Based on the information collected, a solution was proposed: new blockchain-based crowdfunding platform. Blockchain will be used for reliable storage of critical data, which is transaction information. Also, proposed platform will be one of the learning systems. Machine learning will be used to teach the web service to determine the probability of project success. It is assumed that described platform will not have critical vulnerabilities and will fully comply with Web 3.0.

Краудфандинг (з англ. crowdfunding – фінансування натовпом) – це механізм залучення фінансування з метою реалізації проєкту, проведення заходу, допомоги нужденним, підтримки бізнесу тощо. Часто трапляється, що для реалізації справді хороших ідей немає коштів. З цієї причини вмирають сотні стартапів, винаходів, фільмів. Однак у світі знайдеться багато людей, які готові підтримати гарну ідею і вкласти гроші в її реалізацію. Саме для цього існують краудфандингові майданчики і саме тому створюється цей проєкт.

Однак сучасний краудфандинг має багато проблем. Найголовніша серед них – неможливість відстежувати транзакції, через що користувачі не можуть бути впевнені, що їхні кошти надійшли саме туди, куди мали. Інша вразливість полягає у розпорядженні зібраними коштами у випадку, якщо ціль збору не була досягнута. Якщо зловмисник отримає доступ до бази транзакцій, то кошти будуть «повернені» зовсім не тим, хто їх надсилав. Ще одна проблема – нерозуміння того, чи ідея, на яку збирають гроші, дійсно варта уваги. Тому краудфандингові платформи, створені за допомогою традиційних методів і технологій, сіють сумніви як у тих, хто має перспективні ідеї, так і у тих, хто може допомогти з їх фінансуванням. Через це лише 40% проєктів досягають мети збору [3].

Проте за останні двадцять років Інтернет помітно змінився. Ми перейшли від Internet Relay Chat (IRC) до сучасних платформ соціальних мереж. Від базових цифрових платежів до найскладніших банківських

онлайн-послуг. Ми навіть випробували абсолютно нові інтернет технології, такі як криптовалюта та блокчейн. Інтернет став життєво важливою частиною людської взаємодії, зв'язку та продовжує розвиватися. До цих пір ми бачили Web 1.0 і 2.0, але Web 3.0 привносить зовсім новий досвід [4]. Він значною мірою опирається на машинне навчання, штучний інтелект (AI) та технологію блокчейну. Це оновлення не оминає і процес краудфандингу. Краудфандинг розвивається і остання розробка, з технологією блокчейну, зробила систему більш прозорою та підзвітною. У Web 3.0 краудфандинг зазнає ще більших змін. Блокчейн дозволяє не лише відстежувати транзакції, а й впроваджувати смарт-контракти, тож користувачі зможуть токенизувати активи своїх проєктів. У першій версії продукту передбачається створення краудфандингової платформи та інтеграція в неї блокчейн-складової, а також машинного навчання [2, с. 317]. Усе це дозволить швидко адаптувати систему до Інтернету нового покоління у подальших версіях.

Платформа, що розробляється, дозволить підприємцям-початківцям швидко збирати кошти для реалізації своїх ідей. При цьому будь-який інвестор зможе зручно надіслати гроші, будучи впевненим, що вони підуть саме визначеному адресату, адже інформацію про свої транзакції зможе отримати кожен користувач, який завантажить блокчейн на свій комп'ютер [1, с. 2]. Окрім того, що інвестори таким чином захищають свої інвестиції, вони також отримують частину комісії за усі нові транзакції, які їм вдалося помістити до блокчейну. Більш того, технологія блокчейну надасть змогу гарантовано отримати свої гроші назад у випадку, якщо необхідна сума не буде зібрана. Також, усі користувачі матимуть змогу отримати інформацію про вірогідність успіху проєкту, визначену штучним інтелектом, який буде навчатися на даних з вікіпедії та інших відкритих джерел. Передбачається, що на основі цієї інформації буде прийматися рішення про доцільність збору та передачі коштів.

Список використаних джерел:

1. Buzharovski S. *Introducing Blockchain with Java: Program, Implement, and Extend Blockchains with Java*. Apress Media LLC, 2022. 173 p.
2. Ko C. H., Cheng M. Y. Dynamic Prediction of Project Success Using Artificial Intelligence. *Journal of Construction Engineering and Management*. 2007. P. 316-321. DOI:10.1061/(ASCE)0733-9364(2007)133:4(316) (date of access: 02.03.2024).
3. Li Y., Reddy C. K., Rakesh V. Project Success Prediction in Crowdfunding Environments. *ResearchGate*. URL: https://www.researchgate.net/publication/310820239_Project_Success_Prediction_in_Crowdfunding_Environments (date of access: 05.03.2024).
4. Що таке Web 3.0 та чим воно важливо? *Binance Academy*. URL: <https://academy.binance.com/uk/articles/the-evolution-of-the-internet-web-3-0-explained> (date of access: 02.03.2024).

ДОДАТОК Е
Слайди презентації



Рисунок Е.1 – Слайд № 1

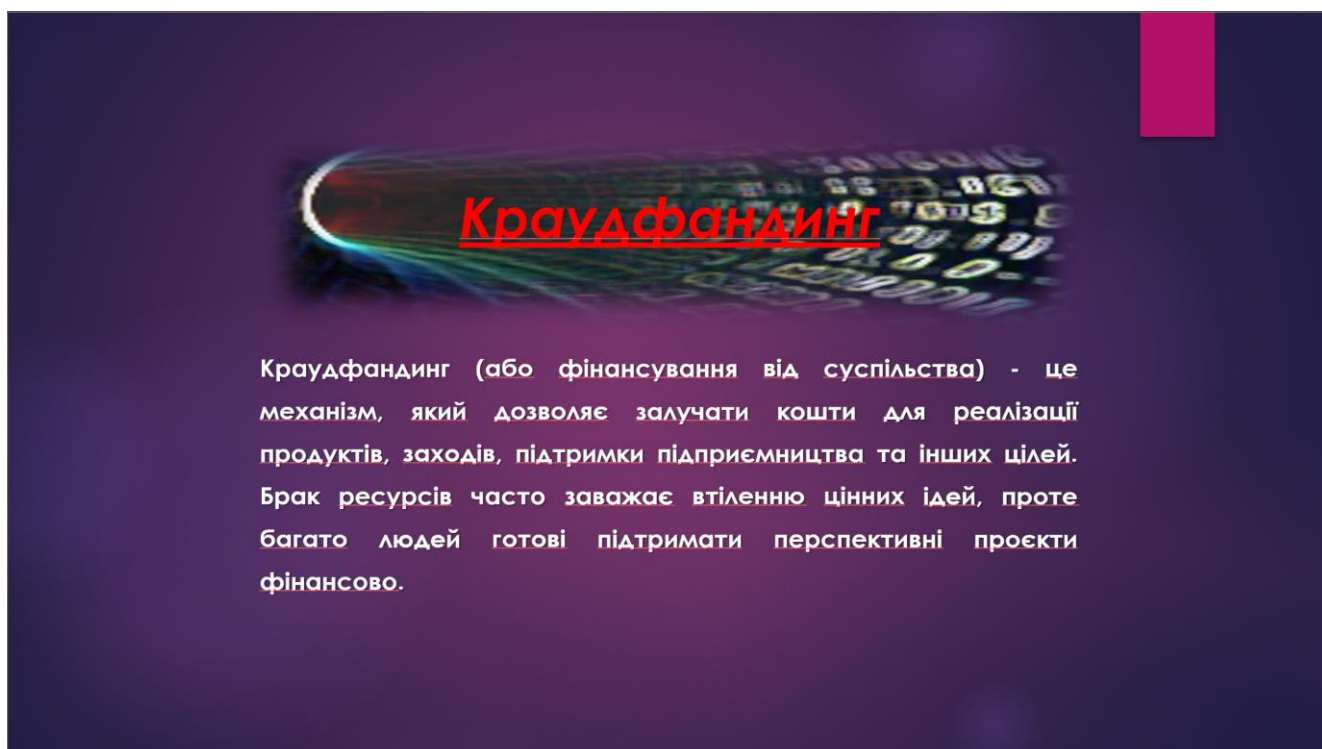


Рисунок Е.2 – Слайд № 2

Аналіз предметної галузі:

- ▶ Провідні наукові установи активно досліджують методи децентралізації у фінансових та соціальних процесах краудфандингу.
- ▶ Вчені розробляють нові моделі, які підвищують ефективність та зменшують ризики для учасників, що вказує на значний інтерес до оптимізації добровільного залучення коштів за допомогою сучасних технологій.



Рисунок Е.3 – Слайд № 3

Проблеми у галузі краудфандингу:

- недостатня прозорість та довіра до використання коштів;
- високі комісії та обмежений доступ до фінансування;
- складність прогнозування успішності проєктів.

Для вирішення цих проблем необхідний комплексний підхід із застосуванням технології Blockchain та машинного навчання для забезпечення прозорості, безпеки та передбачуваності.

Мета роботи:

- ▶ створення безпечного середовища для фінансування проєктів за допомогою криптовалюти та технології Blockchain;
- ▶ реалізація системи передбачення успішності проєктів на основі аналізу даних та використання методів машинного навчання;
- ▶ забезпечення прозорості та надійності процесу краудфандингу за допомогою технології Blockchain.

Рисунок Е.4 – Слайд № 4

Постановка задачі

Основні завдання проекту включають:

- ▶ Розробку інтерактивної краудфандингової платформи.
- ▶ Інтеграцію технології Blockchain для безпеки та прозорості транзакцій.
- ▶ Реалізацію системи прогнозування успішності проєктів на основі машинного навчання.
- ▶ Тестування та вдосконалення системи для оптимального функціонування.
- ▶ Оптимізацію використання ресурсів для швидкої та надійної роботи.
- ▶ Забезпечення масштабованості платформи для зростання користувачів і проєктів.
- ▶ Забезпечення сумісності та інтеграції з іншими сервісами.
- ▶ Врахування правових аспектів щодо зберігання та обробки даних.

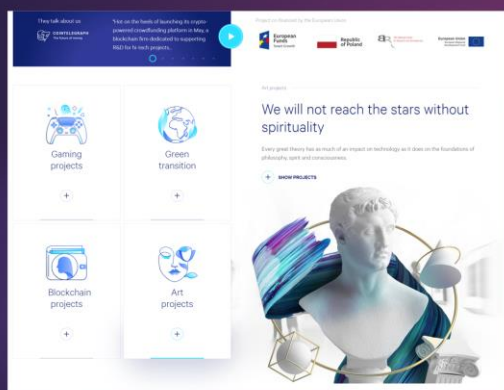
Рисунок Е.5 – Слайд № 5

Аналоги платформи

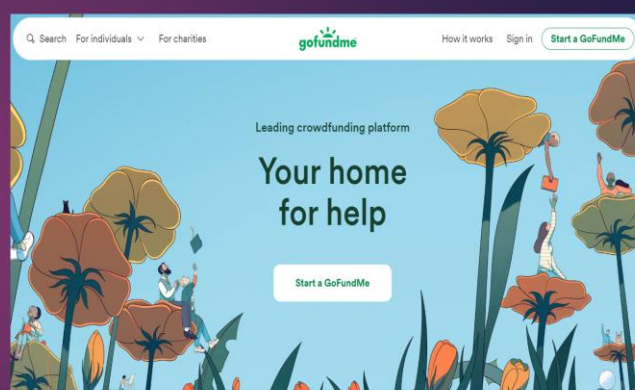
The image displays three screenshots of crowdfunding platforms. On the left is IndieGoGo, showing a search for 'Tech & Innovation' with various product listings. In the center is Fundly, displaying a 'Non-Profit and Charity Fundraisers' section with three active campaigns: 'Breath Of Heaven' (\$1,100 raised), 'Dcn. Nick's 250-Mile Swim' (\$7,254 raised), and 'Idaho Backcountry Veterans...' (\$7,788 raised). On the right is Kickstarter, showing a grid of 14 projects in the 'Food' category, including 'The Pickle Project', 'Bessarabian Cuisine', and 'SACCOONS CAVE'.

Рисунок Е.6 – Слайд № 6

Аналоги платформи



Tetra Space



GoFundMe

Рисунок Е.7 – Слайд № 7

Порівняльна таблиця:

Краудфандинговий майданчик	Прозорість	Комісії	Вигоди для інвесторів	Прогнозування успіху проєкту
Платформа, що розробляється	Доступ до даних через блокчейн	2%	Відшкодування	Є
Kickstarter	Немає	10%	Відшкодування та винагороди	Немає
Indiegogo	Немає	4-9%	Винагороди	Немає
GoFundMe	Історія кампанії	0%	Немає	Немає
Fundly	Історія кампанії	5%	Немає	Немає
Tetra Space	Доступ до даних через блокчейн	10%	Примноження інвестицій	Немає

Рисунок Е.8 – Слайд № 8

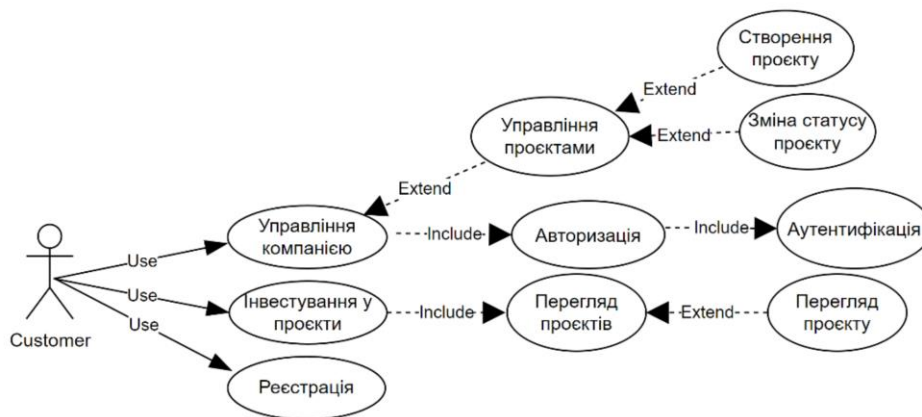


TS



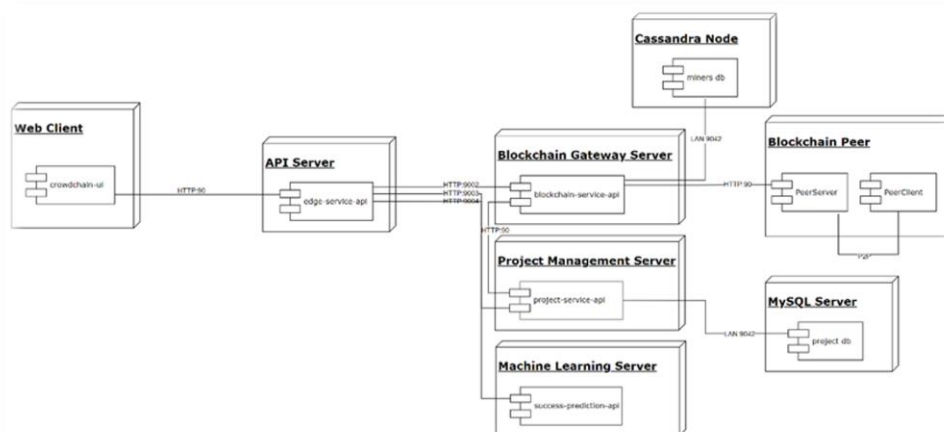
Технології

Рисунок Е.9 – Слайд № 9



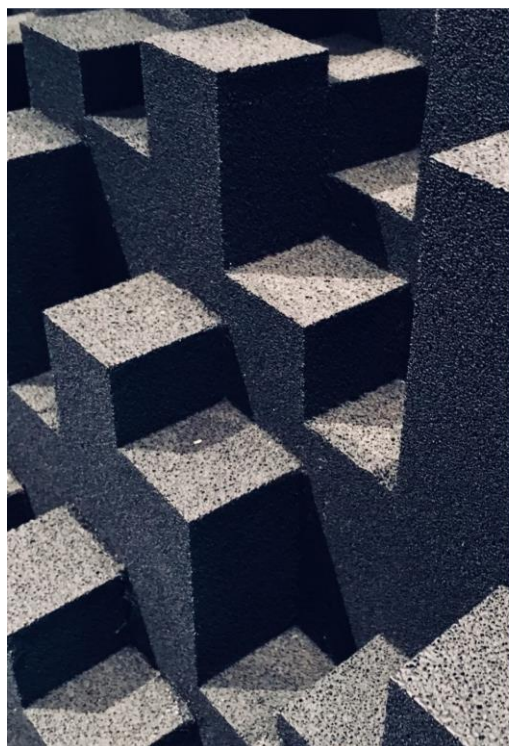
Діаграма прецедентів

Рисунок Е.10 – Слайд № 10



Архітектура

Рисунок Е.11 – Слайд № 11



КОД ВИДОБУТКУ НОВОГО БЛОКУ

```

while (!stopped.get()) {
    Block newBlock = new Block(currentBC);
    latestBlock.ifPresentOrElse(
        (block) -> newBlock.setPrevHash(block.getCurrHash()),
        () -> newBlock.setPrevHash(new byte[] {0});
    newBlock.setTransactionLedger(new
    ArrayList<>(newBlockTransactions));
    newBlock.setTimeStamp(LocalDateTime.now().toString());
    newBlock.setMinedBy(minersWallet.getPublicKey().getEncoded
    ());
    newBlock.setCurrHash(new Crypto()
        .applySHA256(newBlock.toString(),
        minersWallet.getPrivateKey()));
    if (validationService.isProved(newBlock, latestBlock,
    latestPrevBlock)) {
        currentBC.add(newBlock);
        newBlockTransactions.clear();
        return newBlock;
    }
}
  
```

Рисунок Е.12 – Слайд № 12

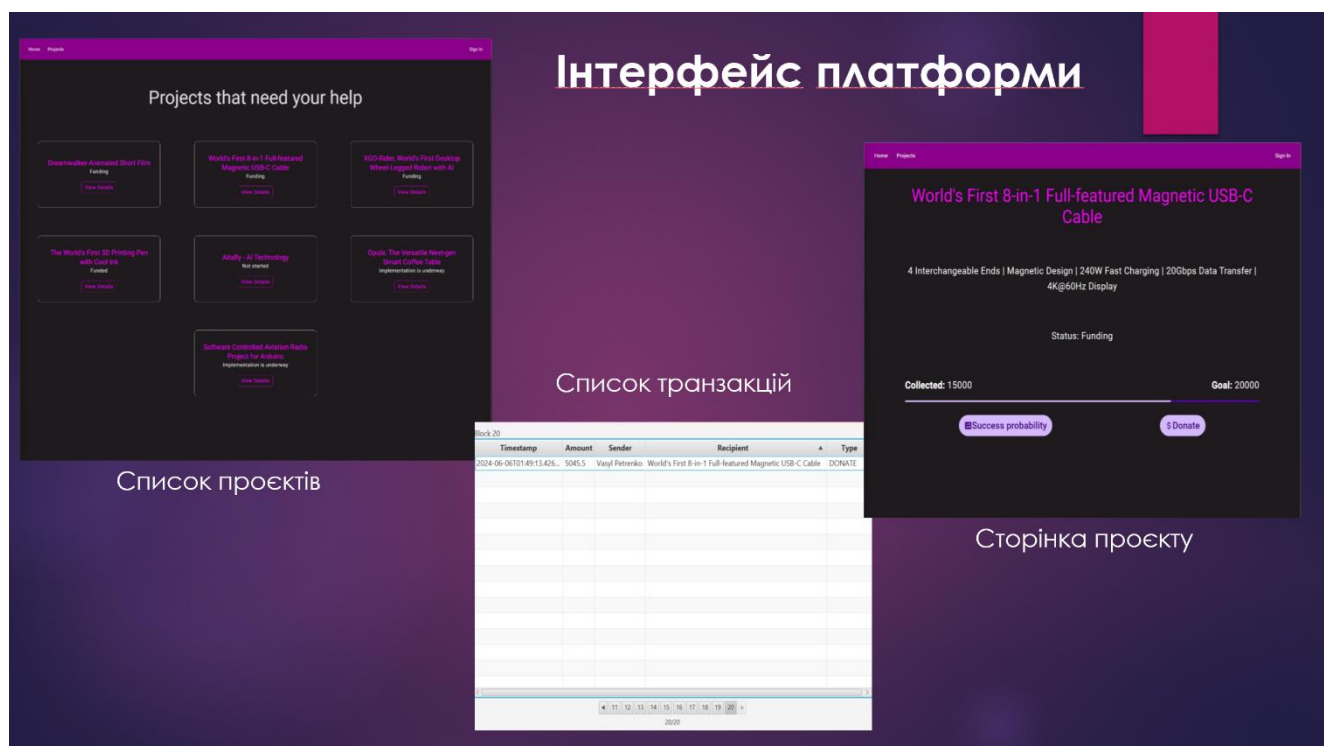
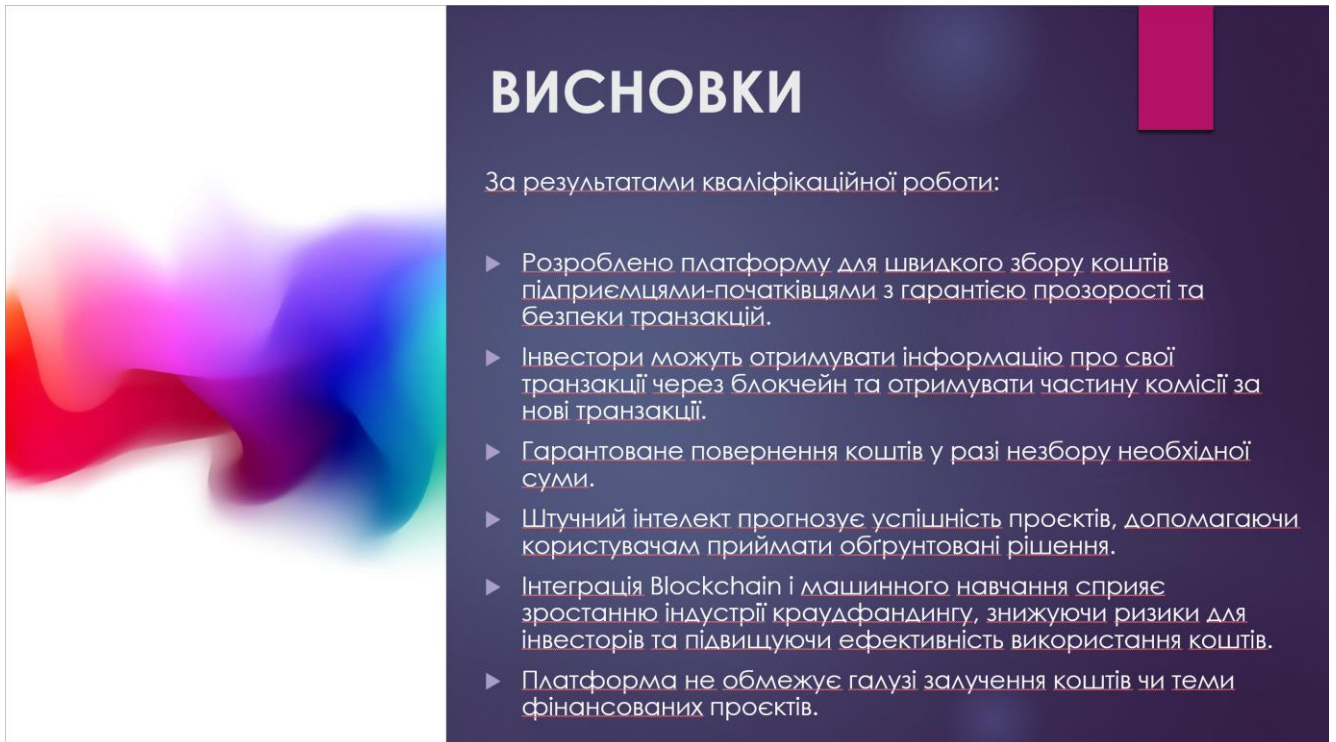


Рисунок Е.13 – Слайд № 13



Рисунок Е.14 – Слайд № 14




ВИСНОВКИ

За результатами кваліфікаційної роботи:

- ▶ Розроблено платформу для швидкого збору коштів підприємцями-початківцями з гарантією прозорості та безпеки транзакцій.
- ▶ Інвестори можуть отримувати інформацію про свої транзакції через блокчейн та отримувати частину комісії за нові транзакції.
- ▶ Гарантоване повернення коштів у разі незбору необхідної суми.
- ▶ Штучний інтелект прогнозує успішність проєктів, допомагаючи користувачам приймати обґрунтовані рішення.
- ▶ Інтеграція Blockchain і машинного навчання сприяє зростанню індустрії краудфандингу, знижуючи ризики для інвесторів та підвищуючи ефективність використання коштів.
- ▶ Платформа не обмежує галузі залучення коштів чи теми фінансованих проєктів.

Рисунок Е.15 – Слайд № 15



Дякую за увагу!

Рисунок Е.16 – Слайд № 16