

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

_____ Оксана ШОВКОПЛЯС
(підпис)

10 грудня 2024

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»

освітньо-професійної програми «Інформатика»

на тему: Інформаційна технологія створення програмного забезпечення
децентралізованої краудфандингової платформи на базі Web3

здобувача групи ІН.м-33 Щербаня Дмитра Олеговича

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело.

_____ Дмитро ЩЕРБАНЬ
(підпис)

Керівник,
старший викладач кафедри
комп'ютерних наук, к.ф.-м.н., доцент

Сергій ШАПОВАЛОВ

_____ (підпис)

Суми – 2024

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Оксана ШОВКОПЛЯС

(підпис)

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки», освітньо-професійної програми «Інформатика»
здобувача групи ІН.М-33 Щербаня Дмитра Олеговича

1. Тема роботи: Інформаційна технологія створення програмного забезпечення децентралізованої краудфандингової платформи на базі Web3
затверджую наказом по СумДУ від «03» грудня 2024 р. №1257-VI
2. Термін задачі здобувачем кваліфікаційної роботи до 06 грудня 2024 року
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблематики централізованих платформ і визначення завдань децентралізації.
2) Огляд технологій для смарт-контрактів і інтеграції Веб3. 3) Розробка функціональних компонентів платформи з використанням thirdweb. 4) Аналіз результатів роботи.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
6. Консультанти до проєкту (роботи), із значенням розділів проєкту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «14» жовтня 2024 р.

Завдання прийняв до виконання _____
(підпис)

Керівник _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблематики централізованих платформ і визначення завдань децентралізації</i>		
2	<i>Огляд технологій для смарт-контрактів і інтеграції Веб3</i>		
3	<i>Розробка функціональних компонентів платформи з використанням thirdweb</i>		
4	<i>Аналіз виконаної роботи</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти _____
(підпис)

Керівник _____
(підпис)

АНОТАЦІЯ

Записка: 61 стор., 18 рис., 1 додаток, 40 джерел.

Обґрунтування актуальності теми роботи – тема кваліфікаційної роботи є актуальною, оскільки децентралізовані краудфандингові платформи на основі технології Веб3 забезпечують прозорість, безпеку та ефективність у процесі залучення коштів. Використання блокчейн-технологій і смарт-контрактів сприяє мінімізації ризиків шахрайства, зниженню витрат і спрощенню доступу до фінансування для широкого кола користувачів, що є важливим для розвитку сучасної цифрової економіки.

Об’єкт дослідження – технічні рішення та інструменти для реалізації децентралізованих краудфандингових платформ із використанням thirdweb.

Мета роботи – розробка децентралізованої краудфандингової платформи із застосуванням технологій Веб3.

Методи дослідження – аналіз сучасних технологій та інструментів, які використовуються для створення децентралізованих додатків, моделювання та розробка з інтеграцією функціональних компонентів із використанням thirdweb.

Результати – розроблено децентралізовану платформу для краудфандингу, яка забезпечує: створення смарт-контрактів для автоматизації фінансових процесів, управління кампаніями збору коштів із прозорим відображенням даних про транзакції.

БЛОКЧЕЙН, ВЕБ3, THIRDWEB, КРАУДФАНДИНГ, DAPP

ЗМІСТ

ВСТУП	5
1. ІНФОРМАЦІЙНИЙ ОГЛЯД ТЕХНОЛОГІЙ КРАУДФАНДИНГОВИХ ПЛАТФОРМ	6
1.1 Актуальність тематики роботи.....	6
1.2 Огляд існуючих рішень у сфері краудфандингових платформ	8
1.3 Постановка задачі	12
2. ОСНОВНІ ПОЛОЖЕННЯ ДЛЯ РОЗРОБКИ ТА ФУНКЦІОНУВАННЯ ДЕЦЕНТРАЛІЗОВАНОЇ КРАУДФАНДИНГОВОЇ ПЛАТФОРМИ	14
2.1 Аналіз принципів роботи децентралізованих платформ	14
2.2 Блокчейн Ethereum для створення платформ	18
2.3 Роль децентралізації у краудфандингу	21
2.4 Технології та методи забезпечення безпеки транзакцій і збору коштів	24
2.5 thirdweb для створення та управління Web3-додатками	27
3. ПРАКТИЧНА РЕАЛІЗАЦІЯ	31
3.1 Програмна реалізація.....	31
3.2 Тестування роботи краудфандингової платформи	43
ВИСНОВОК.....	50
СПИСОК ЛІТЕРАТУРИ.....	51
ДОДАТОК А.....	55

ВСТУП

Сучасний світ перебуває в епоху стрімкої цифровізації, де інформаційні технології відіграють ключову роль у трансформації різних сфер життя. Одним із найбільш перспективних і інноваційних напрямків є впровадження блокчейн-технологій та створення децентралізованих рішень на їх основі. У цьому контексті блокчейн Ethereum посідає провідну позицію завдяки своїй гнучкості, масштабованості та потужній екосистемі смартконтрактів.

Однією з важливих галузей застосування технологій блокчейну є створення децентралізованих платформ для краудфандингу. Вони забезпечують прозорість фінансових процесів, автоматизацію збору та розподілу коштів і знижують ризики шахрайства. Технології Веб3, у свою чергу, дозволяють інтегрувати користувачів у ці платформи, надаючи їм можливість безпосередньо взаємодіяти з блокчейном через зручні інтерфейси.

У зв'язку зі зростаючим попитом на децентралізовані фінансові інструменти актуальним є дослідження та розробка програмного забезпечення для таких платформ. Це дозволяє підвищити ефективність залучення коштів, забезпечити їхню прозорість і покращити довіру користувачів до краудфандингових механізмів.

Метою даної роботи є аналіз та впровадження інформаційного й програмного забезпечення децентралізованої краудфандингової платформи, заснованої на технології Веб3. У процесі дослідження буде розглянуто основи роботи смартконтрактів, механізми інтеграції з блокчейном Ethereum та методи забезпечення безпеки і зручності використання платформи.

1. ІНФОРМАЦІЙНИЙ ОГЛЯД ТЕХНОЛОГІЙ КРАУДФАНДИНГОВИХ ПЛАТФОРМ

1.1 Актуальність тематики роботи

У сучасному світі краудфандинг став важливим інструментом для реалізації ідей та проєктів. Цей підхід дозволяє зібрати необхідні кошти від широкої аудиторії небайдужих людей, зацікавлених у підтримці новаторських рішень, соціальних ініціатив, творчих задумів чи стартапів. На відміну від традиційних методів фінансування, краудфандинг надає доступ до ресурсу широкому колу людей, які можуть внести свій внесок, навіть якщо їхній бюджет обмежений.

Основною перевагою краудфандингу є його універсальність. Цей механізм може бути використаний у найрізноманітніших сферах – від науки і технологій до мистецтва та благодійності. Він створює унікальну можливість для авторів проєктів презентувати свої ідеї та отримати підтримку без необхідності звертатися до великих інвесторів чи банків. Для багатьох новаторів краудфандинг стає першим кроком до успіху, оскільки дозволяє перевірити попит на проєкт, отримати зворотний зв'язок та знайти свою цільову аудиторію. Ще однією важливою рисою краудфандингу є його соціальна значимість. Завдяки йому люди можуть об'єднуватися навколо спільних ідей, створювати спільноти і взаємодіяти одне з одним. Учасники фінансування часто відчують, що стають частиною чогось більшого, впливаючи на розвиток цікавих для них ініціатив. Це сприяє формуванню взаємодовіри та зміцненню зв'язків між учасниками.

У квітні 2012 року президент Барак Обама підписав закон про JOBS (Jumpstart Our Business Startups), також відомий як «закон про краудфандинг» [1]. Закон про JOBS мав на меті зменшити нормативне навантаження на малий бізнес і легалізував краудфандинг акціонерного капіталу, зокрема скасував заборону на загальні збори, яка не дозволяла підприємцям оприлюднювати інформацію про те, що вони збирають гроші. На практиці закон JOBS дозволив

підприємствам та інвесторам налагодити зв'язок на ранніх стадіях циклу розробки, дозволяючи приватним інвесторам отримати переваги, які раніше були зарезервовані лише для заможних інвесторів та венчурних капіталістів. Початковий рахунок передбачав максимум 1 070 000 доларів на зібрану суму. Indiegogo та Kickstarter стали двома найпопулярнішими платформами краудфандингу [2], де підприємці, стартапи та люди з творчими проектами могли залучати капітал для фінансування розвитку. Це впливало на певний сектор розвитку бізнесу, але залишило поза увагою багато більших, більш прибуткових якісних підприємств з більшими потребами в капіталі.

Спираючись на початковий успіх краудфандингу та розширюючи доступний ринок, який може отримати вигоду від цього відносно нового методу залучення капіталу, 2 листопада 2020 року Комісія з цінних паперів і бірж оголосила про нові правила для акціонерного краудфандингу [3]. Ці нові правила набули чинності 15 березня 2021 року. Основні зміни були щодо спрощення збору коштів за допомогою краудфандингу, а також збільшення річного ліміту з ~1 мільйона доларів до 5 мільйонів доларів.

За останні роки обсяг ринку краудфандингу стрімко зріс із 15,42 мільярда доларів США у 2023 році до 17,87 мільярда доларів США у 2024 році при середньорічному темпі зростання 15,9% [4]. Зростання в історичний період можна пояснити потребами у фінансуванні стартапів і малого бізнесу, доступом до глобального капіталу, а також підтримкою творчих проектів, для соціальних цілей.

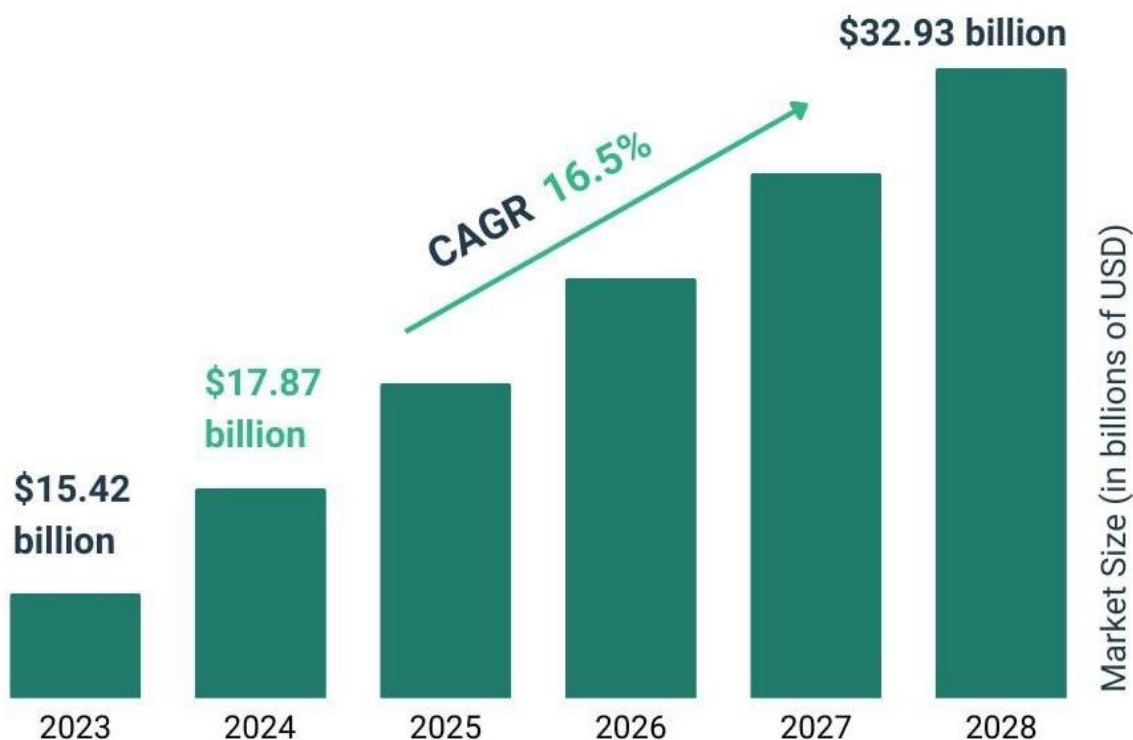


Рисунок 1.1 – Прогноз зростання ринку краудфандингу

На рисунку 1.1 зображена очікувана зміна розміру ринку краудфандингу. Очікується продовження зростання в наступні кілька років. І в 2028 році він зросте до 32,93 мільярда доларів США при середньорічному темпі зростання 16,5%. Зростання в прогнозований період можна пояснити диверсифікацією моделей краудфандингу, зростанням краудфандингу нерухомості, впровадженням відомими компаніями, технологічною інтеграцією та платформами, глобальним розширенням екосистеми краудфандингу. Основні тенденції в прогнозованому періоді включають транскордонні краудфандингові кампанії, впровадження невзаємозамінних токенів у краудфандингу, а також використання смарт-контрактів для краудфандингу[5].

1.2 Огляд існуючих рішень у сфері краудфандингових платформ

На сьогоднішній день існує безліч краудфандингових платформ, кожна з яких має свої переваги та обмеження. Більшість із них орієнтовані на

підтримку різних категорій проєктів – від творчих ініціатив до технологічних стартапів. Вони надають зручний інтерфейс для користувачів, але часто мають обмеження щодо географії, правил використання або комісій. Серед найбільш популярних платформ можна виділити Indiegogo, Kickstarter та Patreon. Після детального аналізу кожної з них можна відзначити особливості кожної з них.

1. Indiegogo, створена за рік до Kickstarter, є краудфандинговою платформою, яка об'єднує користувачів і дозволяє їм збирати кошти на свої кампанії та сприяти іншим. У кампаніях Indiegogo запускаються нові та революційні продукти ще до того, як вони потраплять на основні ринки. Завдяки платформі спонсори мають можливість підтримувати підприємців і нові технології з самого раннього етапу розвитку. Вони можуть переглядати кампанії, читати історії від самих підприємців, оцінювати етапи розвитку та будь-які потенційні виробничі ризики, а потім фінансувати проєкти, успіх яких вони хочуть допомогти. Indiegogo відома своєю універсальністю та гнучкістю. Вона дозволяє авторам проєктів збирати кошти як для комерційних, так і для соціальних ініціатив. Однією з ключових особливостей Indiegogo є можливість вибору між фіксованим і гнучким фінансуванням: у першому випадку автор отримує кошти лише за умови досягнення цільової суми, у другому – отримує всі зібрані кошти незалежно від результату. Такий підхід дозволяє адаптувати платформу під різні потреби користувачів, але в свою чергу має ряд недоліків. Порівняно з Kickstarter, Indiegogo не такий популярний, загальний рівень трафіку на Kickstarter приблизно в 4 рази вищий. Це не означає, що стартап не зможе отримати вигоду від краудфандингової кампанії. Це просто означає, що їм потрібно трохи більше попрацювати, щоб охопити цільову аудиторію. Навіть якщо стартап не отримує повного фінансування кампанії, він все одно повинен надати винагороду, за підтримку кампанії. Це може поставити стартап у слабке фінансове становище, якщо він не отримав бюджетної інфраструктури, на яку очікував. Зрештою в той час як Kickstarter стягує фіксовану комісію в 5% лише за успішні кампанії, Indiegogo

стягуватиме комісію в розмірі 9% від загальної зібраної суми, якщо стартап вирішить взяти пожертвовані гроші, але не досяг цілей зі збору коштів [6].

2. Kickstarter є найвідомішою платформою краудфандингу на даний момент, і їй приписують понад 400 000 успішно профінансованих кампаній, отримавши понад 4 мільярди доларів у вигляді внесків від 17,2 мільйонів спонсорів. На даний момент краудфандингова платформа має понад 13 категорій і 36 підкатегорій, починаючи від творчого мистецтва та харчової промисловості до видавничої та технологічної індустрії з моменту її запуску в 2009 році [7]. Платформа поділена на творців і спонсорів. Творці – це ті, хто представляє свої ідеї проектів і створює кампанії, а спонсори – це ті, хто їх фінансує. Щоб створити успішний проект на Kickstarter, творцям потрібно налаштувати сторінку, щоб відображати всі деталі свого проекту за допомогою тексту, відео та фотографій, для ознайомлення глядачів зі своїм проектом. Потім творці встановлюють ціль фінансування та кінцевий термін, включаючи різні винагороди, які спонсори можуть отримати, вносячи певну суму. Як правило, різні рівні підтримки забезпечать різні винагороди. Не зважаючи на це все, платформа все ще має свої недоліки. Щоб отримати кошти від краудфандингу через кампанію на Kickstarter, стартап має досягти своїх цілей фінансування. Якщо цього не зробити, стартап не отримає того, що міг зібрати. Хоча початкових витрат немає, якщо кампанія стартапу буде успішною, Kickstarter візьме 5% усіх пожертвованих грошей як комісію. Крім того, до 3% комісії за обробку платежів можуть зменшити кошти стартапу. Враховуючи, що середня тривалість краудфандингової кампанії становить до 90 днів, ранні прихильники можуть забути, що вони взяли участь у кампанії. Як наслідок, під час стягнення плати за методом онлайн-платежів ці спонсори можуть оскаржити їх. Через це деяким стартапам важко отримати доступ до всієї готівки.

3. Patreon має іншу модель роботи, яка фокусується на регулярній підтримці від прихильників. Ця краудфандингова платформа, спеціально створена для художників, онлайн-знаменитостей і впливових людей, які

працюють у сфері створення контенту. За останні кілька років популярність платформи значно зросла. Його підхід до краудфандингу на основі винагород є унікальним, і це те, що відрізняє його від Kickstarter та Indiegogo. Як правило, на двох інших платформах краудфандингу спонсори підтримують творців одноразовими внесками в обмін на винагороду. Однак модель Patreon заохочує патронів підписуватися на вміст творців і здійснювати регулярні платежі на постійній основі в обмін на доступ до їх вмісту. Коли стартапи створюють обліковий запис на Patreon, їх просять класифікувати свої проекти та працювати в 14 різних категоріях платформи. Patreon ідеально підходить для підприємців і творчих команд, які постійно створюють контент [8]. Стартапи з іншими видами проектів, які потребують краудфандингу, як-от бізнес-проекти та збір коштів для екстреної медичної допомоги, повинні замість цього перевірити Kickstarter або Indiegogo. Незважаючи на те, що більшість оглядів насамперед є додатковими, розділ «Довідка» Patreon не такий широкий, як хотілося б. З цієї причини для стартапів може бути незрозуміло, як працює фінансування на платформі та яка частина їхніх коштів йде на комісію за обробку. На відміну від Kickstarter і Indiegogo, Patreon не має вбудованих інструментів просування, за винятком можливості для стартапів додавати посилання на соціальні мережі до своєї кампанії. Крім того, у 2018 році виникла серйозна проблема, коли численні творці та стартапи отримали значно нижчі виплати, ніж отримували раніше, через відмову Patreon приймати платежі їхніх підписників. Крім того, багато творців і стартапів виявили, що не можуть отримати доступ до своїх коштів, що залишилися на цій платформі. Patreon назвав причиною цього обмеження «підозрілу поведінку» [9]. Через обмежений розділ довідки платформи творці, які стикалися з цими проблемами, часто не отримували відгуків, роз'яснень чи допомоги від платформи.

1.3 Постановка задачі

1. Літературний огляд:

- Провести детальний огляд літератури за темою, зосереджуючись на основних публікаціях і наукових дослідженнях у галузі краудфандингових платформ, блокчейн-технологій та технології Веб3.

- Визначити актуальні проблеми й виклики, що пов'язані зі створенням та функціонуванням децентралізованих краудфандингових платформ.

2. Аналіз методик роботи краудфандингових платформ:

- Дослідити існуючі методи організації краудфандингу на традиційних та блокчейн-платформах.

- Проаналізувати механізми прозорості, довіри та безпеки, що застосовуються у децентралізованих системах збору коштів.

3. Розробка архітектури платформи:

- Розробити інформаційну модель краудфандингової платформи, включаючи опис основних компонентів: смартконтрактів, користувацького інтерфейсу, та інтеграції з блокчейном Ethereum.

- Дослідити можливості автоматизації процесів збору, розподілу та контролю коштів через смартконтракти.

4. Розробка програмного забезпечення:

- Реалізувати смартконтракт для краудфандингової платформи з використанням технології Веб3, забезпечуючи ключові функції: збір коштів, визначення цільової суми, повернення внесків у разі невдачі кампанії.

- Створити користувацький інтерфейс для взаємодії з платформою через Web3.js чи інші інструменти інтеграції.

5. Тестування та валідація:

Провести перевірку ефективності роботи платформи, включаючи коректність транзакцій, прозорість процесів та безпеку.

б. Висновки та рекомендації:

- На основі результатів дослідження сформулювати висновки щодо ефективності та перспектив розробленої краудфіндингової платформи.
- Надати рекомендації для подальшого розвитку платформи, зокрема покращення її функціональності та підвищення рівня безпеки й зручності для користувачів.

2. ОСНОВНІ ПОЛОЖЕННЯ ДЛЯ РОЗРОБКИ ТА ФУНКЦІОНУВАННЯ ДЕЦЕНТРАЛІЗОВАНОЇ КРАУДФАНДИНГОВОЇ ПЛАТФОРМИ

2.1 Аналіз принципів роботи децентралізованих платформ

Децентралізована програма (dApp) – це програма, яка працює в мережі блокчейн і є:

- з відкритим вихідним кодом – вихідний код є загальнодоступним, що означає, що будь-хто може перевіряти, використовувати, копіювати та змінювати його;
- децентралізованою і криптографічно захищеною – уся інформація в dApp є криптографічно захищеною та зберігається в загальнодоступному децентралізованому блокчейні, який підтримується кількома користувачами;
- токенованою системою – до dApp можна отримати доступ за допомогою криптографічного токена. Вони можуть приймати криптовалюту або генерувати власний токен за допомогою консенсусного алгоритму, наприклад Proof-of-Work або Proof-of-Stake.

Поява та широке поширення dApps пов'язане з розвитком технологій Web3. Цей термін був введений Гевіном Вудом і стосується наступного покоління Інтернету, побудованого на різних децентралізованих цифрових технологіях, таких як криптовалюти, незамінні токени, децентралізовані автономні організації, децентралізовані фінанси, тощо [10].

Блокчейн є базовою технологією, яка керує dApps. Основні елементи екосистеми блокчейну забезпечують безпеку, продуктивність і виконання функцій dApps. Це смарт-контракти, алгоритми консенсусу та стандарти токенів, які визначають набір правил, яким має відповідати токен у межах блокчейну. Токени можуть представляти різноманітні активи та є важливими для dApps, особливо тих, які включають роботу з криптовалютами або цифровими активами.

Усі операції в dApp мають детермінований характер. Це означає, що dApps виконують свої функції однаково, незалежно від середовища

виконання. Іншими словами, за умови однакових вхідних даних, dApp завжди генеруватиме ідентичний результат, незалежно від часу чи місця виконання. Така детермінованість є ключовим фактором для забезпечення узгодженості та надійності роботи в децентралізованій мережі. Оскільки dApps працюють у одноранговій мережі з багатьма вузлами, вузли повинні узгодити результат кожної операції [11].

Ключовою відмінністю між децентралізованими та традиційними веб-додатками є їх базова архітектура. Веб-програма розміщена на централізованих серверах. На відміну від цього, децентралізована програма працює в децентралізованій мережі контроль і прийняття рішень розподілені між декількома автономними органами, а не централізовані в одному.

Більшість dApps працюють завдяки взаємодії трьох компонентів: смарт-контрактів, блокчейнів і токенів. В основі кожного dApp лежить один або кілька смарт-контрактів. dApp використовує технологію blockchain для збереження своєї децентралізованої природи. Для взаємодії з dApp потрібен «газ», який оплачується у рідному токени блокчейну.

Додатки dApps набули значної популярності, насамперед завдяки безлічі переваг, які вони надають перед традиційними централізованими додатками:

1. Безпека, завдяки своїй децентралізованій природі dApps менш вразливі до злому та витоку даних, оскільки немає єдиної точки збою.
2. Прозорість, усі транзакції та зміни реєструються в блокчейні, що забезпечує прозору та незмінну історію операцій програми.
3. Стійкість до цензури, dApps не керуються жодним центральним органом, що робить їх стійкими до цензури та втручання з боку урядів чи корпорацій.
4. Контроль користувача, користувачі мають повний контроль над своїми даними та активами, що знижує ризики, пов'язані з централізованим зберіганням даних [12].

Оскільки можливості застосування децентралізованих програм продовжують розширюватися та розвиватися, з'явилося кілька різних

категорій dApps, кожна з яких служить унікальним цілям і пропонує різні функціональні можливості. Ці категорії відображають різні способи використання технології блокчейн для децентралізації різних секторів і аспектів нашого життя, від фінансів і бірж до ігор і управління ланцюгом поставок. Децентралізовані програми можна класифікувати за різними критеріями, зокрема за функціональністю, призначенням, галузевим застосуванням та платформою розробки. Такий підхід дозволяє краще зрозуміти різноманітність та можливості dApps у сучасному цифровому середовищі.

Децентралізовані програми поділяються на три основні типи залежно від рівня їхньої взаємодії з блокчейном [13]:

- Type I dApps: працюють безпосередньо на власному блокчейні. Прикладом є Ethereum, який надає платформу для розгортання смартконтрактів і інших dApps.
- Type II dApps: використовують блокчейн програм типу I та вимагають спеціальних токенів для функціонування. Наприклад, мережа OMG, яка працює на Ethereum.
- Type III dApps: створюються на базі програм типу II, формуючи багаторівневу ієрархічну структуру. Наприклад, децентралізована програма для голосування, побудована на протоколі Ethereum.

dApps розробляються для різних сфер застосування, серед яких можна виділити наступні основні типи [14]:

- DeFi (децентралізовані фінанси): орієнтовані на фінансові операції без посередників. Приклади: Compound, Aave, MakerDAO.
- Ринки та біржі: децентралізовані платформи для обміну активами. Приклади: OpenSea, Uniswap.
- Децентралізовані автономні організації (DAO): організації, які керуються правилами, закодованими в смартконтрактах. Приклад: MolochDAO.

- Соціальні мережі та комунікація: забезпечують децентралізовані соціальні взаємодії. Приклади: Peepeth, Status.
- Ігри та розваги: створення децентралізованих платформ для ігор і розваг. Приклади: CryptoKitties, Decentraland.
- Ідентифікація та репутація: рішення для перевірки особи та управління репутацією. Приклади: uPort, Civic.
- Ланцюг постачання та Інтернет речей: покращують прозорість ланцюга постачання та інтеграцію IoT-пристроїв. Приклади: OriginTrail, Chainlink.
- Децентралізоване зберігання даних: надають альтернативу традиційним хмарним сховищам. Приклади: Filecoin, Storj.
- Створення та монетизація контенту: дозволяють творцям напряму заробляти на споживачах. Приклади: Audius, LBRY.

Різні мережі блокчейну надають інструменти для розробки dApps, кожна з яких має свої особливості [15]:

- Ethereum: найбільш популярна платформа з широкою підтримкою спільноти.
- Binance Smart Chain: приваблює своїми низькими комісіями та швидкими транзакціями.
- TRON: забезпечує масштабованість і популярна у секторі розваг.
- EOS: відома безкоштовними транзакціями та масштабованістю.
- Інші: платформи, такі як Tezos, Cardano, Polkadot, також пропонують інноваційні рішення для створення dApps.

Ця класифікація відображає різноманіття можливостей, які відкриваються завдяки використанню децентралізованих програм. Вибір платформи, сектора чи рівня залежить від завдань і цілей конкретного проєкту.

2.2 Блокчейн Ethereum для створення платформ

Очікується, що до 2026 року ринок технології блокчейн досягне 67,4 млрд доларів. Враховуючи те, як швидко з'являються послуги розробки програмного забезпечення Web3, галузь, безсумнівно, запровадить кілька інноваційних підходів [16].

Оскільки на ринку так багато різноманітних мереж, потрібно витратити час, щоб зважити свої альтернативи та обрати найбільш підходящу мережу. Це надзвичайно важливо, враховуючи різноманітність функцій, які можуть пропонувати блокчейн-мережі. Різні архітектури блокчейну можуть зробити продукти кращими чи гіршими, а отже вибір блокчейн-мережі є важливим етапом у процесі розробки децентралізованих рішень. Різноманітність доступних платформ вимагає детального аналізу їх характеристик, переваг та недоліків. Для визначення найбільш підходящої технології слід враховувати низку ключових критеріїв: публічний чи приватний, рівень використання, конфіденційність та безпека, масштабованість, вартість транзакцій, екосистема.

Публічні блокчейни є відкритими для всіх користувачів. Вони часто використовуються для створення децентралізованих бірж або інших платформ, які не потребують високого рівня контролю даних. Такі блокчейни пропонують прозорість, доступність та децентралізацію. Крім того, вони зазвичай мають власний токен для виконання операцій.

Натомість приватні блокчейни забезпечують обмежений доступ для учасників. Вони характеризуються підвищеною безпекою, високою швидкістю обробки транзакцій та нижчим рівнем енергоспоживання. Завдяки цим особливостям, приватні блокчейни ідеально підходять для розробки корпоративних програмних рішень, таких як SaaS-додатки, або спеціалізованих NFT-маркетплейсів [17].

Ще одним важливим критерієм є популярність та рівень поширення блокчейн-технології. Чим вищий рівень використання мережі, тим більше

шансів, що вона буде активно підтримуватися та вдосконалюватися в майбутньому. Мережі з високим рівнем adoption rate також мають більшу кількість інструментів і документації для розробників, що значно спрощує процес створення продукту. Наприклад, мережі Ethereum і Binance Smart Chain широко використовуються, що гарантує їх подальший розвиток.

Безпека є критично важливим аспектом для будь-якої блокчейн-мережі. Перед вибором варто проаналізувати історію атак на платформу, оцінити нанесені збитки та виявити причини вразливостей. Надійна платформа повинна мати високий рівень захисту даних користувачів та механізми запобігання несанкціонованому доступу [18].

Масштабованість визначає здатність блокчейн-мережі обробляти великий обсяг транзакцій за одиницю часу. Це особливо важливо для проєктів, що мають потенціал для стрімкого зростання. Наприклад, платформи з підтримкою високої пропускну здатності, такі як Solana, забезпечують швидкі транзакції навіть при значному збільшенні кількості користувачів.

Вартість транзакцій безпосередньо впливає на бюджет проєкту та зручність користування платформою для кінцевих клієнтів. Мережі на основі алгоритму консенсусу Proof of Work, зазвичай мають високі комісії, тоді як платформи з алгоритмом Proof of Stake, пропонують меншу комісію за транзакцію [19]. Ефективний розподіл бюджету враховує як комісії за транзакції, так і загальну взаємодію користувачів із платформою.

Широка екосистема платформи та доступність інструментів для розробки є додатковим фактором, який варто врахувати. Платформи, такі як Ethereum, мають численні бібліотеки, готові шаблони смартконтрактів і активну спільноту розробників, що сприяє швидкому впровадженню інновацій.

Ethereum є лідером серед блокчейнів завдяки своїй зрілості, активній спільноті розробників, широкому вибору інструментів та безпеці. Прогресивна дорожня карта та гнучкість у програмуванні роблять Ethereum оптимальним вибором для створення децентралізованого додатку

краудфандингової платформи, що відповідатиме сучасним технологічним викликам.

Ethereum має найбільшу екосистему активних розробників серед усіх блокчейн-платформ. Його нативна криптовалюта, Ether (ETH), є другою за ринковою капіталізацією у світі. Ця платформа стала основою для створення безлічі креативних рішень на основі блокчейну [20]. Завдяки відкритості спільноти Ethereum, розробники мають доступ до широкого спектра інструментів і фреймворків, що значно спрощує процес розробки. Наприклад, інструменти для написання смарт-контрактів часто вже доступні у вигляді готових рішень з відкритим кодом. Чим більше проєктів створюється в екосистемі Ethereum, тим вища її загальна цінність, що стимулює подальший розвиток платформи.

Ethereum функціонує з 2015 року та за цей час продемонстрував високу стабільність. Платформа була протестована в численних сценаріях використання, включаючи фінансові послуги, NFT, DeFi та багато інших сфер. На даний момент на платформі працює понад 300 проєктів, а кількість активних смарт-контрактів перевищує 50 мільйонів [21]. Протягом багатьох років Ethereum зазнавав різних протоколів оновлення, що свідчить про здатність децентралізованої спільноти вирішувати значні технічні виклики.

Не зважаючи на здобуті досягнення мережа продовжує вдосконалюватися, впроваджуючи інноваційні оновлення, які спрямовані на підвищення продуктивності, зниження вартості транзакцій та покращення досвіду користувачів.

Одним із ключових етапів розвитку стало оновлення Merge, коли платформа перейшла з механізму консенсусу Proof of Work на Proof of Stake. Це дозволило значно знизити енергоспоживання, підвищити безпеку та оптимізувати транзакції. Майбутні плани розвитку включають адаптацію Ethereum до викликів, пов'язаних із квантовими обчисленнями, що підвищить стійкість мережі до нових загроз. Дорожня карта Ethereum також дозволяє

розробникам заздалегідь планувати впровадження оновлень для своїх додатків[22].

Ethereum Virtual Machine (EVM) підтримує кілька мов програмування, таких як Solidity та Vyper, які ідеально підходять для початківців. Для досвідчених розробників доступні проміжні мови, як-от Yul, Huff та Cairo, що дозволяють створювати оптимізовані смарт-контракти.

Ethereum пропонує широкий вибір інструментів для розробки та тестування dApps. Серед найбільш популярних інструментів:

- Truffle – середовище розробки та тестування для смарт-контрактів.
- Hardhat – інтегроване середовище розробки для створення та розгортання додатків на Ethereum.
- OpenZeppelin SDK – набір інструментів для роботи зі смарт-контрактами.

Ці інструменти дозволяють швидко запускати локальні екземпляри мережі, компілювати смарт-контракти та тестувати їх перед розгортанням[23].

Ethereum має одну з найбільш розвинених систем безпеки серед блокчейнів. Протягом багатьох років екосистема стикалася з різними атаками, що сприяло розробці ефективних заходів захисту. Сьогодні компанії, які займаються аудитом смарт-контрактів, пропонують широкий спектр послуг для забезпечення безпеки додатків. Новостворені блокчейни не мають такого ж рівня підтримки, що робить Ethereum більш привабливим вибором для розробників, які цінують надійність своїх проєктів.

2.3 Роль децентралізації у краудфіндингу

Традиційні краудфіндингові платформи мають низку обмежень, серед яких високі комісії, необхідність у посередниках, суворі регуляторні вимоги та ризики шахрайства або невиконання зобов'язань. Децентралізований краудфіндинг, що базується на технології блокчейн і смарт-контрактах, є альтернативою, яка сприяє більш прозорому, ефективному та демократичному

фінансуванню проєктів. Завдяки своїм можливостям, децентралізований краудфандинг має потенціал значного впливу на розвиток бізнесу та інновацій. Основними перевагами даної концепції є те, що:

1. Зникає необхідність у посередниках, таких як банки, платіжні процесори чи традиційні платформи, які зазвичай стягують комісії та накладають обмеження на використання коштів і доступ до фінансування. Завдяки цьому знижується вартість участі для обох сторін: як для ініціаторів проєктів, так і для спонсорів. Цей підхід також сприяє створенню більш різноманітних ініціатив, які раніше могли бути недоступними через високу вартість входу чи регуляторні обмеження.

2. Смарт-контракти, що лежать в основі децентралізованих краудфандингових платформ, гарантують автоматичне виконання умов кампанії, таких як цільова сума фінансування, дедлайни, обіцянки винагород та умови повернення коштів. Це дозволяє забезпечити прозорість процесів, оскільки кошти будуть випущені лише після виконання усіх умов. Спонсори можуть відстежувати прогрес і результати проєкту, що значно знижує ризики шахрайства та підвищує відповідальність ініціаторів [24].

3. Надає ініціаторам можливість залучати колективний інтелект та креативність спільнот, формувати тісний зв'язок зі спонсорами і створювати проєкти разом із ними. Окрім фінансової підтримки, учасники кампаній можуть брати участь у спільному прийнятті рішень, голосуванні чи інвестуванні через токенізацію. Це створює відчуття спільної відповідальності та сприяє появі інноваційних рішень.

4. Сприяє розвитку інновацій та соціального впливу у сферах мистецтва, культури, науки, технологій та інших галузей. Завдяки прозорості та відповідальності такого підходу, спонсори можуть відстежувати прогрес проєктів, оцінювати їхню легітимність і якість.

5. Завдяки відсутності централізованих платформ чи інвесторів, які часто диктують умови, ініціатори проєктів зберігають більше контролю над

своїми ідеями та творчою свободою. Вони також отримують доступ до глобальної аудиторії, що дозволяє залучити більш різноманітних учасників.

Децентралізований краудфандинг є новаторським підходом до фінансування проєктів, що зменшує витрати, підвищує прозорість і створює умови для тіснішої взаємодії між ініціаторами та спільнотами. Завдяки використанню блокчейну та смарт-контрактів, ця концепція відкриває нові можливості для розвитку бізнесу, стимулювання інновацій та соціальних змін.

Також такі платформи можна використовувати для розвитку криптовалютного краудфандингу завдяки забезпеченню безпеки, прозорості та ефективності у процесі збору коштів. Усі транзакції записуються у незмінний публічний реєстр, що дозволяє інвесторам бачити, як використовуються їхні внески. Це підвищує довіру до проєкту та допомагає ініціаторам демонструвати свою відповідальність [25].

Існує кілька типів криптовалютного краудфандингу, кожен із яких має свої унікальні особливості та переваги.

1. Initial Exchange Offering (IEO) – проводиться на криптовалютних біржах, які виступають посередниками між проєктами та інвесторами. Біржа забезпечує продаж токенів, створюючи додаткову довіру до кампанії.

2. Initial Coin Offering (ICO) – дозволяє компаніям створювати та продавати свої токени для залучення фінансування. Інвестори можуть купувати токени за криптовалюту або фіатні кошти, а згодом торгувати ними на біржах.

3. Security Token Offering (STO) – схожі на ICO, але регулюються законами про цінні папери. Токени зазвичай підтримуються реальними активами або джерелами доходів, надаючи інвесторам можливість отримати прибуток або право голосу.

4. Initial DEX Offering (IDO) – є новим децентралізованим способом краудфандингу, що дозволяє запускати продаж токенів через децентралізовані біржі. Це забезпечує швидший процес залучення коштів і відкриває нові можливості для стартапів [26].

Блокчейн-краудфандинг є інноваційним підходом до фінансування проєктів, що усуває бар'єри традиційних платформ і відкриває нові можливості для підприємців та інвесторів. Завдяки своїй прозорості, безпеці та глобальності, цей метод стає важливим інструментом для залучення коштів і розвитку інноваційних ідей.

2.4 Технології та методи забезпечення безпеки транзакцій і збору коштів

Як вже згадувалося раніше, традиційні технології краудфандингу мають низку технічних недоліків, пов'язаних із безпекою даних, прозорістю та відстежуваністю. Такі проблеми характерні не лише для краудфандингових платформ, а й для багатьох інших централізованих систем. Основна архітектурна особливість традиційних технологій – централізоване управління. Це означає, що для роботи платформи потрібен посередник або довірена особа, яка контролює виконання операцій.

Централізований підхід створює низку вразливостей. Зокрема, система стає більш схильною до шахрайських дій, що підриває довіру інвесторів. Контроль над процесом належить користувачам або адміністраторам платформи, а не самій системі, що створює потенційні ризики недобросовісної поведінки. Крім того, такий підхід може призводити до зниження прозорості операцій, створюючи лазівки для приховування дій або маніпуляцій даними[27].

Ще одним недоліком традиційного краудфандингу є залежність від третіх сторін, таких як банки або державні органи. Ця необхідність значно ускладнює та уповільнює процес, оскільки кожна операція має бути підтверджена зовнішніми організаціями. Внаслідок цього зловмисники можуть використовувати слабкі місця системи для витоку конфіденційної інформації або інших шкідливих дій.

Значною проблемою є ще відсутність ефективних механізмів відстеження даних у централізованих системах. Це ускладнює моніторинг і контроль діяльності користувачів, що додатково знижує ефективність та безпеку системи.

Смарт-контракт – це комп’ютерна програма, яка керує передачею цифрових грошей або активів між двома сторонами за незаперечними правилами. Розумний контракт визначає правила та штрафні санкції, пов’язані з угодою, так само, як і традиційний контракт, але він може забезпечити виконання зобов’язань, передбачених угодою автоматично.

Система створена за допомогою React.js як інтерфейс користувача [28], а смарт-контракт на Solidity – як бекенд. Solidity – це об’єктно-орієнтована мова програмування, яка використовується в розумних контрактах. Використовуючи Metamask, користувач зможе отримати доступ до системи. Web3.js працює для з’єднання між користувачами та смарт-контрактами. Коли клієнт створює новий проект, смарт-контракт також автоматично генерує та вносить дані про це в Ethereum [29]. Поєднання цих компонентів створює базовий будівельний блок системи краудфандингу. Розумний контракт забезпечує прозорість, щоб заохотити спонсорів вкладати свої ресурси в проект.

Для впровадження технології блокчейн необхідно вирішити низку важливих питань, які безпосередньо впливають на досягнення необхідного рівня її продуктивності. Хоча блокчейн пропонує інноваційні рішення в галузі безпеки, прозорості та децентралізації, ефективність його застосування залежить від подолання технічних та організаційних викликів.

Прозорість транзакцій є одним із ключових переваг блокчейн-технології, однак водночас вона створює ризик розкриття конфіденційності користувачів. Деякі види атак, такі як selfish mining та Sybil-атаки, прямо суперечать принципам блокчейн-систем. У консенсусному алгоритмі Proof-of-Work selfish mining може призводити до марної витрати ресурсів чесних

майнерів. Sybil-атака виникає, коли користувач створює кілька ідентичностей у різних частинах мережі, що дозволяє маніпулювати системою [30].

Окрім цих загроз, автономність смарт-контрактів може стати причиною появи нових типів атак. Три основні напрями вразливостей включають:

- уразливості безпосередньо в блокчейні;
- помилки у коді смарт-контрактів;
- проблеми у віртуальній машині, яка виконує код.

Створення контракту є центральним елементом смарт-контрактів. Слід розробляти цифрові контракти та адаптувати їхні умови до різних платформ, таких як Ethereum, Hyperledger чи NEM [31].

Однак одна з основних характеристик смарт-контрактів – їхня незмінність після впровадження – робить верифікацію перед запуском надзвичайно важливою. Недоліки у вихідному коді можуть призводити до непередбачуваних наслідків. Крім того, більш ніж 77% смарт-контрактів не мають публічного вихідного коду, що робить їх непрозорими для офіційних аудиторів. Це ускладнює аудит та підвищує ризики помилок [32].

Послідовність виконання транзакцій є ще однією проблемою, яка може виникати у блокчейн-системах. Коли в одному блоці виконується кілька транзакцій, які залежать одна від одної, це може спричинити проблему конкурентності транзакцій. У таких випадках неправильна або відкладена послідовність може дозволити зловмисникам запустити атаку на систему.

У смарт-контрактах Ethereum майнери можуть встановлювати довільні позначки часу з певним рівнем толерантності при створенні нового блоку. Ця функція може використовуватися для проведення атак, коли зловмисники змінюють позначки часу в рамках дозволеного діапазону, що впливає на результати роботи системи.

Смарт-контракти можуть викликати себе самостійно або за допомогою функцій. Стек викликів, який використовується для зворотного відстеження послідовності викликів, має обмеження у 1024 записи. Зі збільшенням складності блокчейн-мережі та кількості функцій смарт-контрактів це

обмеження може бути перевищене, що призводить до виняткових ситуацій та помилок у виконанні.

У традиційних підходах до розробки смарт-контрактів відсутній механізм зупинки транзакцій до їхнього завершення. У разі помилки транзакція все одно записується у блокчейн, що є незворотною дією через природу блокчейна як системи з незмінним зберіганням даних [33]. Це може вплинути на продуктивність системи та ускладнити її використання.

Час підтвердження транзакцій у блокчейн-системах іноді недостатній для бізнес-цілей. Наприклад, підтвердження транзакції в блокчейні може займати 10 хвилин або більше. Використання системи спрощеної перевірки платежів, Simplified Payment Verification [34], може допомогти скоротити цей час, але ця стратегія не перевіряє всі транзакції вузла, що може мати свої недоліки.

Розглянуті проблеми демонструють, що, попри значний потенціал блокчейн-технологій, їхнє впровадження вимагає ретельного врахування можливих викликів. Надійне вирішення цих проблем сприятиме подальшому розвитку блокчейн-екосистеми та розширенню її застосування в різних галузях.

2.5 thirdweb для створення та управління Web3-додатками

Сучасний розвиток технологій Web3 відкриває нові можливості для створення децентралізованих додатків, однак розробка таких рішень часто потребує глибоких знань у програмуванні смарт-контрактів, інтеграції блокчейн-функціоналу та роботи з децентралізованими мережами. Платформа thirdweb пропонує інноваційний підхід до створення Web3-додатків, спрощуючи процеси розгортання смарт-контрактів, керування токенами та інтеграції блокчейн-функцій. Використовуючи thirdweb, розробники можуть зосередитися на бізнес-логіці та дизайні своїх проєктів, не витрачаючи значних ресурсів на вирішення технічних складнощів. Цей розділ присвячений

аналізу можливостей thirdweb, її ролі у створенні децентралізованих рішень та переваг, які платформа надає Web3-розробникам.

Заснована Фурканом Ридханом і Стівеном Бартлеттом у 2021 році, thirdweb є провідною платформою для розробників web3. Розробники можуть використовувати службу thirdweb для створення, керування та аналізу своїх програм web3. Компанія пропонує ряд пропозицій послуг, таких як аналітичні платформи, протоколи автентифікації та параметри децентралізованого зберігання. На сьогодні понад 55 000 розробників користуються продуктами thirdweb і нещодавно його оцінили в 160 мільйонів доларів [35].

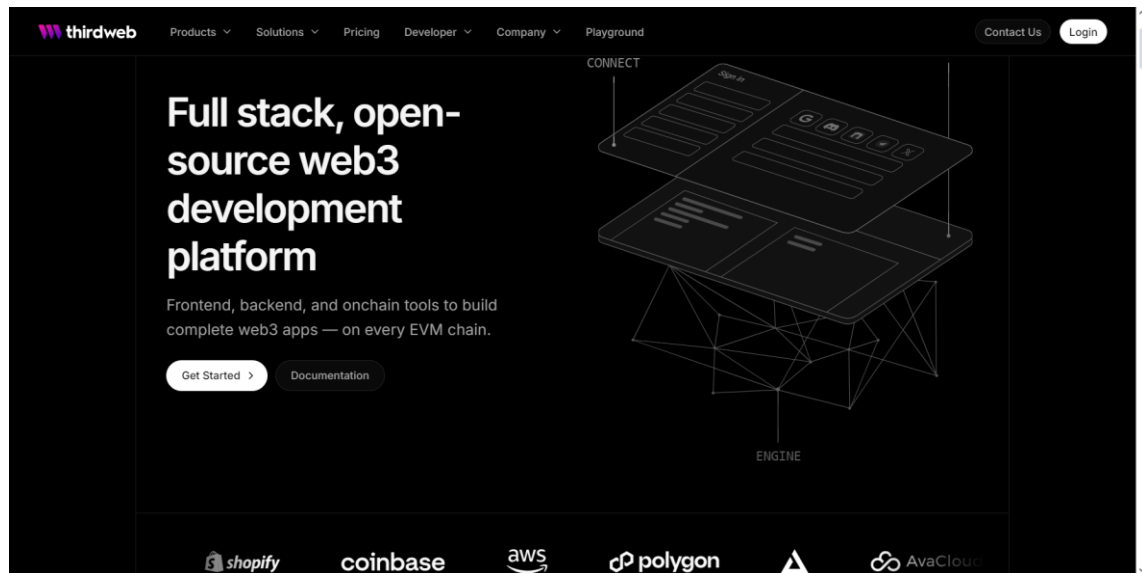


Рисунок 2.1 – Головна сторінка thirdweb

На рисунку 2.1 зображена головна сторінка thirdweb. thirdweb – це проста у використанні веб-панель для створення програм Web3. thirdweb спрощує створення та розгортання таких програм, як NFT collection drops. Зазвичай для отримання NFT потрібно кодувати смарт-контракт та інші сценарії для карбування NFT. thirdweb виконує цей процес за вас. thirdweb можна використовувати з об'єктами, що зберігаються в IPFS, тому об'єкти, що зберігаються у сегменті Filebase IPFS [36], можна безперешкодно завантажувати для використання з програмою thirdweb.

Thirdweb – це не просто ще одна назва у величезному морі платформ Web3, це вказівник як для розробників. Пропонуючи набір інструментів і

послуг, спеціально розроблених для екосистеми Web3, Thirdweb здобув собі нішу. Ключові особливості:

- Комплексна основа розвитку.
- Повна інтеграція з децентралізованими мережами.
- Зручний інтерфейс, який підходить як новачкам, так і експертам.

Платформа thirdweb відкриває широкі можливості для розробників, допомагаючи їм створювати інноваційні рішення в межах екосистеми Web3. Її інструменти дозволяють реалізовувати проекти з використанням NFT, децентралізованих автономних організацій, публічних маркетплейсів, ігор з цифровими економіками та інших новітніх концепцій.

Один із найпопулярніших сценаріїв застосування thirdweb – це створення NFT для надання членства, доступу до ексклюзивних послуг або контенту. Ці NFT можуть виступати у ролі токенів доступу до:

- приватних спільнот.
- закритого контенту.
- майбутніх продуктів чи сервісів.
- цифрових чи фізичних подій.

Яскравим прикладом є співпраця thirdweb з кіберспортивною організацією 100 Thieves [37]. Команда thirdweb розробила платформу, яка дозволила користувачам отримувати NFT без необхідності попередньо створювати криптовалютний гаманець. Це залучило понад 700 000 користувачів, багато з яких раніше не стикалися з криптовалютами.

Платформа thirdweb також допомагає інтегрувати можливості Web3 у традиційні Web2-додатки. Це стає все популярнішим серед компаній, які прагнуть модернізувати свої бізнес-моделі.

Наприклад, SaaS-компанія може використовувати NFT-дропа для надання раннього доступу до платформи [38]. Користувачі отримують NFT, який надає їм доступ, а після використання токена можуть перепродати його іншим користувачам. Водночас компанія отримує роялті з кожного

перепродажу, створюючи одночасно цікаву взаємодію для користувачів і нове джерело доходу.

Децентралізовані автономні організації є інструментом для груп людей, які мають спільні інтереси, з можливістю організувати діяльність за допомогою блокчейну. Основна особливість DAO – це децентралізоване управління, коли рішення ухвалюються всіма учасниками, а не обмеженим колом лідерів. DAO часто використовують NFT для продажу членства [39]. Платформа thirdweb спрощує запуск таких організацій, забезпечуючи безпеку, прозорість і гнучкість управління через смартконтракти.

thirdweb сприяє створенню інноваційних рішень у сфері маркетплейсів і цифрових економік. Наприклад, один із клієнтів платформи розробляє маркетплейс для NFT із доповненою реальністю (AR), що стане першим у своєму роді [40]. На такій платформі користувачі зможуть продавати та обмінювати NFT, які застосовуються у віртуальних іграх або VR-платформах.

Крім цього, thirdweb підтримує запуск масштабних NFT-колекцій та ігор, що мають власні цифрові економіки. Ці проєкти дозволяють інтегрувати токенизовані активи та віртуальні валюти в ігрові екосистеми, забезпечуючи новий рівень взаємодії між гравцями та розробниками.

Використання thirdweb дозволяє створювати різноманітні проєкти, що інтегрують можливості блокчейну у звичні та нові сфери. Платформа знижує технічні бар'єри для розробників і стимулює впровадження інновацій, які змінюють уявлення про бізнес-моделі, ігрові світи та користувацький досвід.

3. ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Програмна реалізація

Створення програмного забезпечення для децентралізованої краудфандингової платформи на основі технології Веб3 вимагає інтеграції сучасних інструментів та підходів для розробки. Основною метою цього розділу є детальний опис реалізації ключових компонентів системи, таких як смарт-контракти та інтеграція з блокчейн-мережею. Розробка охоплює вибір відповідних інструментів, налаштування середовища розробки, написання коду смарт-контракту, використання thirdweb для спрощення інтеграції функцій. У цьому розділі буде продемонстровано етапи програмної реалізації, які забезпечують створення безпечної, прозорої та зручної у використанні платформи.

Створення смарт-контракту починається з визначення базової структури, яка виступає фундаментом для подальшої розробки. Ця структура дозволяє описати всі ключові характеристики краудфандингового проєкту. Визначення цієї структури є важливим етапом, оскільки воно формує основу для організації даних і забезпечення зручного управління ними в рамках смарт-контракту.

```
struct Project {  
    address creator;  
    string name;  
    string details;  
    uint256 goal;  
    uint256 dueDate;  
    uint256 totalFunds;  
    string banner;  
    address[] contributors;  
    uint256[] ontributions;  
}
```

Рисунок 3.1 – Код структури Project

Цей код оголошує структуру Project, яка описує модель даних для проєкту на краудфандинговій платформі. Структура використовується для збереження інформації про окремий проєкт і його пов'язані характеристики. Ось детальний опис кожного поля:

- `address creator` - це поле зберігає адресу ініціатора проєкту, тобто гаманця, що створив проєкт. Воно використовується для ідентифікації того, хто є власником проєкту та має право отримувати зібрані кошти.
- `string name` - це поле дозволяє відобразити заголовок, який описує проєкт.
- `string details` - це поле використовується для детального пояснення цілей проєкту, мотивації та інших важливих аспектів, які можуть зацікавити потенційних донорів.
- `uint256 goal` – поле зберігає цільову суму в одиницях криптовалюти, яку необхідно зібрати для виконання проєкту. Визначає фінансову мету краудфандингової кампанії.
- `uint256 dueDate` - містить кінцеву дату збору коштів, представлену в UNIX-часі (кількість секунд, що минули з 1 січня 1970 року), що обмежує тривалість кампанії.
- `uint256 totalFunds` - зберігає суму, яку вже було зібрано на момент звернення до даних про проєкт. Поле дозволяє відстежувати поточний прогрес кампанії.
- `string banner` - поле містить посилання або шлях до зображення банера проєкту. Використовується для графічного представлення або реклами проєкту.
- `address[] contributors` - масив, що зберігає адреси всіх користувачів, які внесли пожертви до цього проєкту. Дозволяє відстежувати, хто брав участь у фінансуванні.
- `uint256[] contributions` - масив, що зберігає суми, які відповідні користувачі з поля `contributors` пожертвували. Поле дозволяє деталізувати інформацію про кожну транзакцію та зв'язати її з відповідним донором.

Ця структура моделює основні аспекти краудфіндингового проєкту, дозволяючи легко зберігати та маніпулювати пов'язаними з ним даними в смарт-контракті. Вона використовується для управління процесом збору коштів, забезпечення прозорості, відстеження внесків і взаємодії з користувачами на децентралізованій платформі.

```
mapping(uint256 => Project) public projects;  
uint256 public projectCount = 0;
```

Рисунок 3.2 – Змінні унікального ідентифікатора та кількості проєктів

Далі було створено змінну для збереження всіх створених проєктів у смарт-контракті, забезпечуючи доступ до них за їх унікальним номером. Завдяки ключу `uint256` кожен проєкт легко знайти та опрацювати, що є важливим для роботи з множиною проєктів у системі. А також змінна, яка зберігає кількість створених проєктів. Вона використовується як лічильник для генерування унікальних ідентифікаторів проєктів, починаючи з нуля. Кожного разу, коли створюється новий проєкт, ця змінна збільшується, забезпечуючи унікальність ідентифікатора для кожного нового запису у `projects`.

Це потрібно для правильної роботи проєктів: `mapping` дозволяє ефективно організувати дані про всі краудфіндингові проєкти, що зберігаються в смарт-контракті; `projectCount` гарантує, що кожен проєкт матиме свій унікальний номер, за яким його можна буде ідентифікувати; Оскільки змінна `projects` є публічною, будь-хто може переглядати інформацію про проєкти безпосередньо через блокчейн.

Наступним етапом реалізації смарт-контракту стала функція `startProject`, яка відповідає за створення нових проєктів у системі. Ця функція дозволяє ініціювати новий краудфіндинговий проєкт із заданими параметрами, зберігаючи інформацію про нього в структурі `Project` та додаючи її до колекції

всіх проєктів. Ключовим завданням функції є забезпечення коректності введених даних і створення унікального запису для кожного проєкту.

```
function startProject(address _creator, string memory _name, string
memory _details, uint256 _goal, uint256 _dueDate, string memory _banner)
public returns (uint256) {
    Project storage project = projects[projectCount];

    require(project.dueDate > block.timestamp, "The dueDate should be a
date in the future.");

    project.creator = _creator;
    project.name = _name;
    project.details = _details;
    project.goal = _goal;
    project.dueDate = _dueDate;
    project.totalFunds = 0;
    project.banner = _banner;

    projectCount++;

    return projectCount - 1;
}
```

Рисунок 3.3 – Код функції startProject

`Project storage project = projects[projectCount];` - створює змінну `project`, яка посилається на новий запис у відображенні `projects`. Ідентифікатор нового проєкту визначається через лічильник `projectCount`.

`require(project.dueDate > block.timestamp, "The dueDate should be a date in the future.");` - гарантує, що вказана дата завершення (`_dueDate`) знаходиться в майбутньому відносно поточного часу. Якщо це не так, функція викликає помилку, забезпечуючи коректність параметрів проєкту.

Далі всі передані функції параметри (`_creator`, `_name`, `_details`, `_goal`, `_dueDate`, `_banner`) зберігаються у відповідних полях структури `Project`. Це забезпечує повноту та доступність інформації про проєкт. Полю `totalFunds` задається початкове значення `0`, оскільки проєкт ще не отримав жодних внесків.

Після чого лічильник `projectCount` збільшується на одиницю, створюючи новий унікальний ідентифікатор для майбутніх проєктів, але функція повертає ідентифікатор щойно створеного проєкту. Це дозволяє викликачеві функції отримати унікальний номер проєкту для подальшої роботи з ним.

Функція `contributeToProject` відіграє важливу роль у функціонуванні краудфандингової платформи, оскільки вона забезпечує можливість користувачам здійснювати внески в проєкти. Основне завдання функції полягає в обробці отриманих коштів, їх передачі ініціатору проєкту та фіксації інформації про транзакцію в блокчейні.

```
function contributeToProject(uint256 _id) public payable {
    uint256 amount = msg.value;

    Project storage project = projects[_id];

    project.contributors.push(msg.sender);
    project.ontributions.push(amount);

    (bool sent,) = payable(project.creator).call{value: amount}("");

    if(sent) {
        project.totalFunds = project.totalFunds + amount;
    }
}
```

Рисунок 3.4 – Код функції `contributeToProject`

Функція починається із прийому параметрів: ідентифікатора проєкту `_id` та автоматично зчитує суму, яку користувач надсилає на адресу смарт-контракту через модифікатор `payable`. Вказаний ідентифікатор `_id` використовується для доступу до відповідного запису проєкту в структурі `projects`.

Рядок `uint256 amount = msg.value;` ініціалізує змінну `amount`, яка зберігає кількість ефірів, надісланих користувачем разом із викликом функції. Далі за ідентифікатором `_id` отримується запис проєкту з мапи `projects`, що дозволяє отримати доступ до відповідної структури `Project`. У полі `contributors`

зберігається адреса користувача, який зробив внесок, а у полі `contributions` фіксується сума цього внеску. Це дозволяє зберігати повну історію транзакцій і забезпечувати прозорість системи.

`(bool sent,) = payable(project.creator).call{value: amount}("");` виконує передачу отриманих коштів на адресу ініціатора проєкту, використовуючи метод `call`. Параметр `sent` використовується для перевірки, чи була транзакція успішною. Після чого у разі успішної передачі коштів поле `totalFunds` оновлюється шляхом додавання отриманого внеску до вже зібраної суми.

Функція `getAllProjects` забезпечує можливість отримання інформації про всі створені проєкти. Це дозволяє користувачам платформи ознайомитися з доступними кампаніями для аналізу та вибору проєктів для підтримки. Основна задача функції полягає у формуванні масиву всіх проєктів, які були створені, та його поверненні.

```
function getAllProjects() public view returns (Project[] memory) {
    Project[] memory allProjects = new Project[](projectCount);

    for(uint i = 0; i < projectCount; i++) {
        Project storage item = projects[i];

        allProjects[i] = item;
    }

    return allProjects;
}
```

Рисунок 3.5 – Код функції `getAllProjects`

Логіка роботи функції полягає в тому, що спочатку створюється новий тимчасовий масив `allProjects`, який має фіксований розмір, що дорівнює кількості проєктів, створених на платформі (вимірюється змінною `projectCount`). Далі у функції реалізований цикл `for`, який проходить через всі індекси збережених у мапі `projects` записів. На кожній ітерації зчитується

інформація про відповідний проєкт із мапи `projects` за допомогою оператора доступу `projects[i]` і зберігається у змінній `item`.

Змінна `item`, яка містить дані поточного проєкту, додається до масиву `allProjects` на позицію, відповідну індексу ітерації. Таким чином, масив `allProjects` поступово заповнюється інформацією про всі доступні проєкти. Після завершення циклу функція повертає цей масив, що містить повний список проєктів.

Ця функція є функцією лише для читання, що визначено модифікатором `view`, що означає, що вона не змінює стан блокчейну і є оптимізованою для отримання даних. Це важливо для збереження ефективності системи, оскільки виклики таких функцій не вимагають оплати комісій за газ.

Функція `getContributors` призначена для отримання інформації про учасників, які зробили внески в конкретний проєкт, а також сум цих внесків. Її основна мета – забезпечити доступ до даних про підтримку проєкту, що сприяє прозорості платформи та дозволяє аналізувати активність користувачів.

```
function getContributors(uint256 _id) view public returns (address[]
memory, uint256[] memory) {
return (projects[_id].contributors, projects[_id].contributions);
```

Рисунок 3.6 – Код функції `getContributors`

Функція приймає один параметр – ідентифікатор проєкту `_id`, за яким здійснюється доступ до запису у мапі `projects`. Використовуючи цей ідентифікатор, функція звертається до властивостей `contributors` і `ontributions` конкретного проєкту.

Властивість `contributors` є масивом адрес користувачів, які зробили внески до цього проєкту, а `ontributions` – це масив сум відповідних внесків. Таким чином, ці два масиви зберігають історію підтримки проєкту: хто з учасників надавав фінансування і в якому обсязі.

Функція визначена як така, що лише читає дані, модифікатор `view`, тому вона не змінює стан блокчейну. Це дозволяє користувачам запитувати дані без додаткових витрат на виконання операції. Результатом виконання функції є повернення двох масивів – адрес користувачів і відповідних їм сум внесків.

Завдяки функції `getContributors` система забезпечує легкий доступ до інформації про підтримку кожного проєкту. Це сприяє підвищенню прозорості процесу краудфандингу, оскільки дозволяє відстежувати, хто саме бере участь у фінансуванні кампаній і в якому обсязі. У контексті децентралізованих платформ така функціональність є важливим елементом для створення довіри між учасниками.

У процесі розробки клієнтської частини децентралізованої краудфандингової платформи було створено низку компонентів, які забезпечують структуровану організацію інтерфейсу користувача. Всі компоненти розміщені в папці `src/components` і виконують окремі функції для досягнення загальної мети. Нижче наведено детальний опис кожного з компонентів та їхньої ролі в реалізації проєкту.

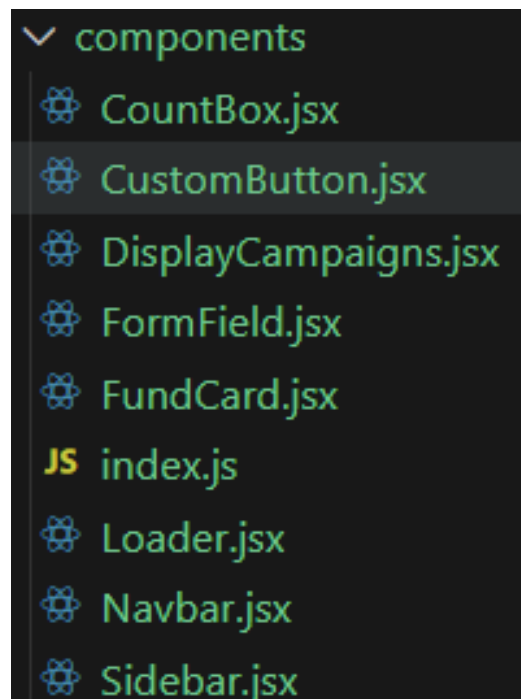


Рисунок 3.7 – Структура файлів компонентів клієнтської частини

- `CountBox.jsx` – компонент, який використовується для відображення числових значень, таких як кількість днів до завершення кампанії, сума зібраних коштів чи кількість учасників. Він забезпечує зручний спосіб подання ключових метрик у зрозумілому вигляді та може включати стилізовані елементи для покращення сприйняття інформації.
- `CustomButton.jsx` – цей компонент реалізує кнопки з кастомним дизайном, які використовуються в різних частинах інтерфейсу платформи. Він підтримує налаштування тексту, стилів і функцій, які викликаються при натисканні, забезпечуючи гнучкість та уніфікований вигляд кнопок у всьому додатку.
- `DisplayCampaigns.jsx` – відповідає за відображення списку активних кампаній. Компонент отримує дані про кампанії та створює візуальний список із використанням інших підкомпонентів, таких як `FundCard`. Це дозволяє зручно представити користувачам інформацію про всі доступні краудфандингові проєкти.
- `FormField.jsx` – компонент для створення полів введення у формах, який підтримує різні типи введення, такі як текст, числа чи електронна пошта. Його функціонал включає валідацію введених даних, що підвищує точність і зручність користувача під час взаємодії з формами.
- `FundCard.jsx` – цей компонент використовується для відображення детальної інформації про окрему кампанію. Він включає такі дані, як назва проєкту, мета збору коштів, зображення банера та зібрана сума. `FundCard` є ключовим елементом візуалізації кампаній, забезпечуючи естетично привабливе та інформативне подання даних.
- `Loader.jsx` – компонент для відображення анімації завантаження або індикатора процесу. Його основна мета – покращити взаємодію користувача з платформою, надаючи зворотний зв'язок під час очікування завершення асинхронних операцій, таких як отримання даних із блокчейну чи сервера.

- `Navbar.jsx` – відповідає за створення верхньої навігаційної панелі платформи. Цей компонент містить логотип, посилання на основні сторінки та кнопку для підключення криптовалютного гаманця. Він забезпечує інтуїтивно зрозумілу навігацію для користувачів.

- `Sidebar.jsx` – реалізує бічне меню, яке використовується для швидкого доступу до основних функцій платформи чи переходу між розділами. Компонент може містити адаптивний функціонал для відкриття або закриття меню залежно від розміру екрана.

- `index.js` – файл, який забезпечує централізований експорт усіх компонентів з папки `components`. Це дозволяє спростити імпорт компонентів у інших частинах проєкту, покращуючи читабельність та підтримку коду.

У структурі проєкту папка `src/pages` містить файли, які відповідають за різні сторінки інтерфейсу користувача децентралізованої платформи. Кожен файл реалізує окремий екран, надаючи доступ до різних функцій та розділів системи.

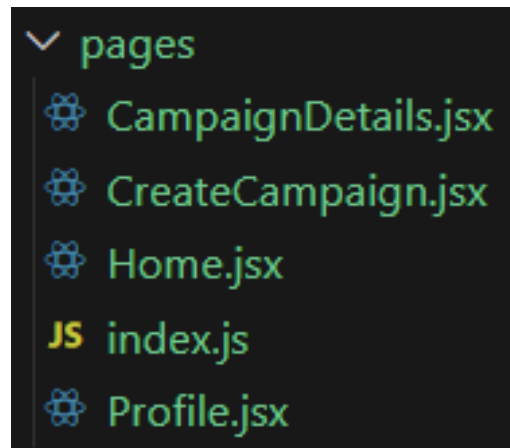


Рисунок 3.8- Структура сторінок інтерфейсу

- `CampaignDetails.jsx` – цей компонент реалізує сторінку з детальною інформацією про конкретну краудфандингову кампанію. Він дозволяє користувачам переглядати ключові характеристики кампанії, такі як її мета, зібрані кошти, залишковий час до завершення та список учасників.

Крім того, сторінка може включати форму для здійснення внеску, забезпечуючи прямий зв'язок з функцією донату.

- `CreateCampaign.jsx` – ця сторінка надає інтерфейс для створення нової краудфандингової кампанії. Компонент містить форму, у якій користувач може вказати деталі кампанії, такі як назва, опис, цільова сума, кінцева дата збору коштів та банерне зображення. Валідація введених даних допомагає забезпечити правильність інформації перед відправкою на сервер або в блокчейн.

- `Home.jsx` – головна сторінка платформи, яка відіграє роль централізованого хабу для відображення активних кампаній. Вона використовує такі компоненти, як `DisplayCampaigns`, для відображення списку доступних кампаній, і може включати фільтри або сортування для зручного пошуку проєктів.

- `Profile.jsx` – компонент, що відповідає за відображення профілю користувача. Сторінка надає інформацію про створені користувачем кампанії, зібрані кошти. Вона також може містити додаткові функції, такі як редагування особистих даних або перегляд історії транзакцій.

- `index.js` – файл, який використовується для централізованого експорту всіх сторінок, розташованих у папці `pages`. Це спрощує імпорт компонентів у інших частинах додатку та робить структуру проєкту більш організованою.

Структура папки `src/pages` забезпечує чіткий поділ між основними розділами платформи, дозволяючи створювати адаптивний і зручний інтерфейс. Кожна сторінка виконує свою функцію, що сприяє модульності, легкості розширення та підтримки проєкту.

`main.jsx` є основним файлом для ініціалізації та рендерингу клієнтського застосунку, який використовує бібліотеку `React` для побудови інтерфейсу користувача.

```

import React from "react";
import ReactDOM from "react-dom/client";
import { BrowserRouter as Router } from "react-router-dom";
import { ChainId, ThirdwebProvider } from "@thirdweb-dev/react";

import { StateContextProvider } from "./context";
import App from "./App";
import "./index.css";

const root = ReactDOM.createRoot(document.getElementById("root"));

root.render(
  <ThirdwebProvider activeChain="sepolia"
  clientId="2bc23b8ae3900f8afa77ffed8427dd91">
    <Router>
      <StateContextProvider>
        <App />
      </StateContextProvider>
    </Router>
  </ThirdwebProvider>
);

```

Рисунок 3.9 – Код файлу main.jsx

Більш детально про інформацію з файлу:

- React та ReactDOM імпортуються для використання основних функціональностей React, таких як створення елементів інтерфейсу та рендеринг в DOM.
- BrowserRouter as Router імпортується з бібліотеки react-router-dom, що дозволяє реалізувати маршрутизацію в додатку, тобто керувати переходами між різними сторінками.
- ChainId та ThirdwebProvider імпортуються з бібліотеки @thirdweb-dev/react, що дає можливість інтеграції з Web3, зокрема для підключення до блокчейну та взаємодії з контрактами. У даному випадку використовується мережа Sepolia.
- StateContextProvider імпортується для забезпечення контексту стану додатку, що дозволяє доступ до глобального стану в будь-якому компоненті проекту.

- App імпортується як головний компонент, який містить всю логіку відображення та структуру застосунку.
- `./index.css` імпортується для застосування стилів до всіх компонентів.
- `const root = ReactDOM.createRoot(document.getElementById("root"));` створює корінь застосунку, прив'язуючи його до DOM-елемента з `id "root"`, що визначає місце, де буде рендеритись React-компонент.
- Метод `render`, фікористовується для рендерингу кореневого компонента `App`, який є точкою входу до інтерфейсу застосунку.
- `ThirdwebProvider` забезпечує доступ до Web3 можливостей та налаштовує взаємодію з блокчейном.
- `Router` керує навігацією по сторінках додатку.
- `StateContextProvider` дозволяє глобальне управління станом додатку, що важливо для масштабованості та спільного доступу до даних між різними компонентами.

Цей файл є базовою ініціалізацією для React-додатку, що працює з блокчейном та використовує додаткові бібліотеки для керування станом і маршрутизацією. Він створює структуру, яка дозволяє розширювати додаток і інтегрувати різні Web3 функції без необхідності змінювати основну логіку рендерингу та взаємодії між компонентами.

3.2 Тестування роботи краудфіндингової платформи

Для подальшої роботи з платформою необхідно провести тестування її функціональності, щоб перевірити всі етапи взаємодії з користувачем, а також забезпечити коректність роботи смарт-контракту. Це включає в себе перевірку створення і підтримки кампаній, здійснення пожертвувань, взаємодію з блокчейном. Тестування також дозволить виявити можливі баги та помилки в

інтерфейсі, що забезпечить користувачам комфортне та безпечне використання платформи.

Після запуску сайту користувачів зустрічає головна сторінка, на якій наразі не відображаються активні проєкти. Це може свідчити про те, що кампанії ще не були створені. На цій сторінці буде зібрана вся необхідна інформація про доступні проєкти, з можливістю переглядати деталі, а також підтримувати взаємодію з платформою, що дасть користувачам можливість долучатися до нових ініціатив.

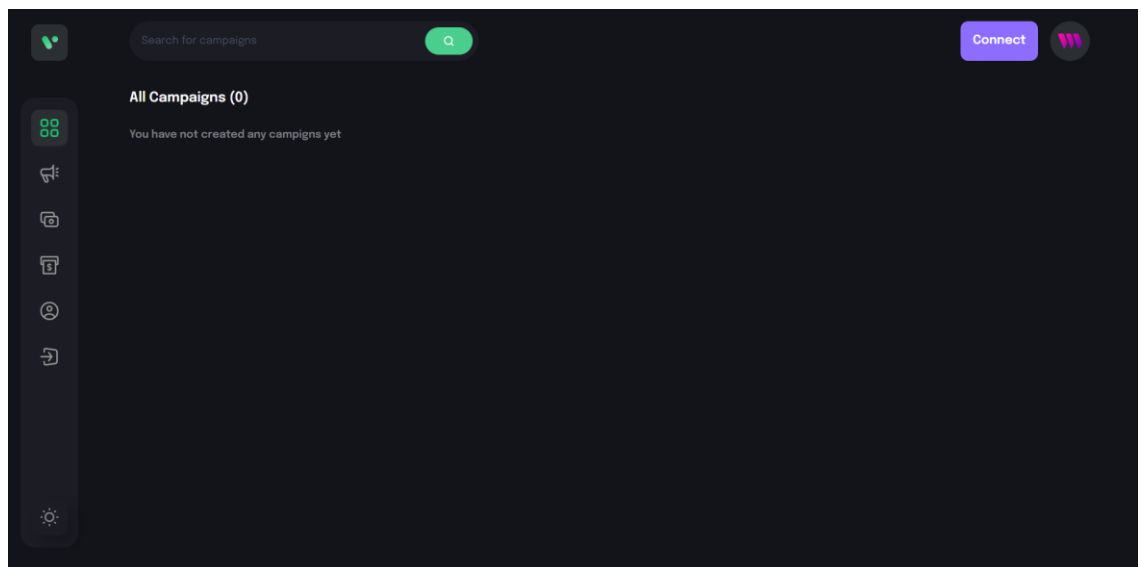


Рисунок 3.10 – Головна сторінка платформи

Для початку підключимось до платформи за допомогою гаманця Metamask. Після чого кнопка Connect змінюється на зелену з написом «Create a project», після натискання на неї відкривається сторінка створення проєкту.

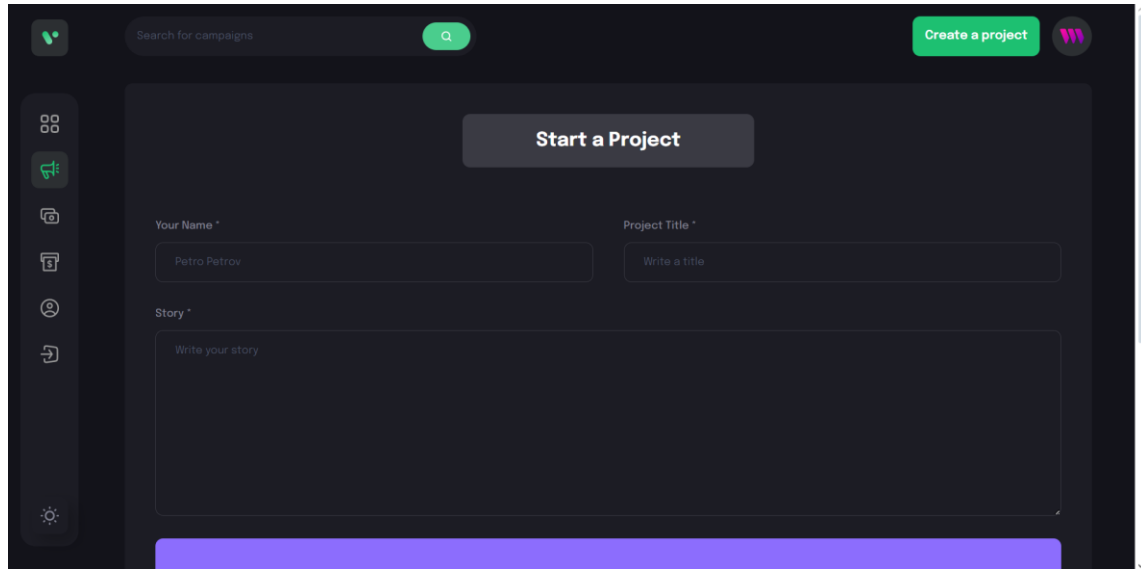


Рисунок 3.11 – Вигляд сторінки створення нового проєкту

Тестування платформи передбачає ретельну перевірку усіх її функцій, зокрема процесу заповнення даних форми для створення нових кампаній. Це важливий етап, оскільки форма є основним інструментом для користувачів, які хочуть ініціювати кампанії на платформі. Під час тестування необхідно перевірити правильність введення даних у кожне поле, зокрема назву, опис, цільову суму, термін закінчення та банер кампанії. Тестування повинно підтвердити, що після успішного заповнення форми кампанія коректно зберігається в блокчейні і відображається на головній сторінці та в списку активних кампаній, забезпечуючи правильне відображення введених даних користувачем.

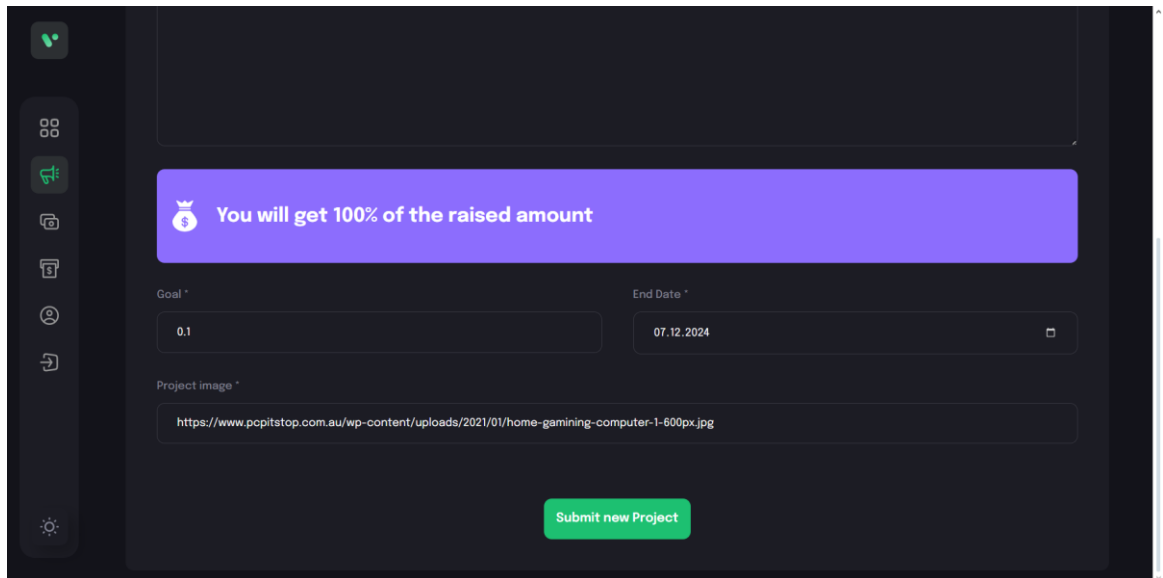


Рисунок 3.11 – Заповнення форми

Тому після заповнення форми інформацією, натискання на кнопку «Submit new Project» створює транзакцію, після успішного виконання транзакції дані повинні відобразитись на головній сторінці, та сторінці користувача.

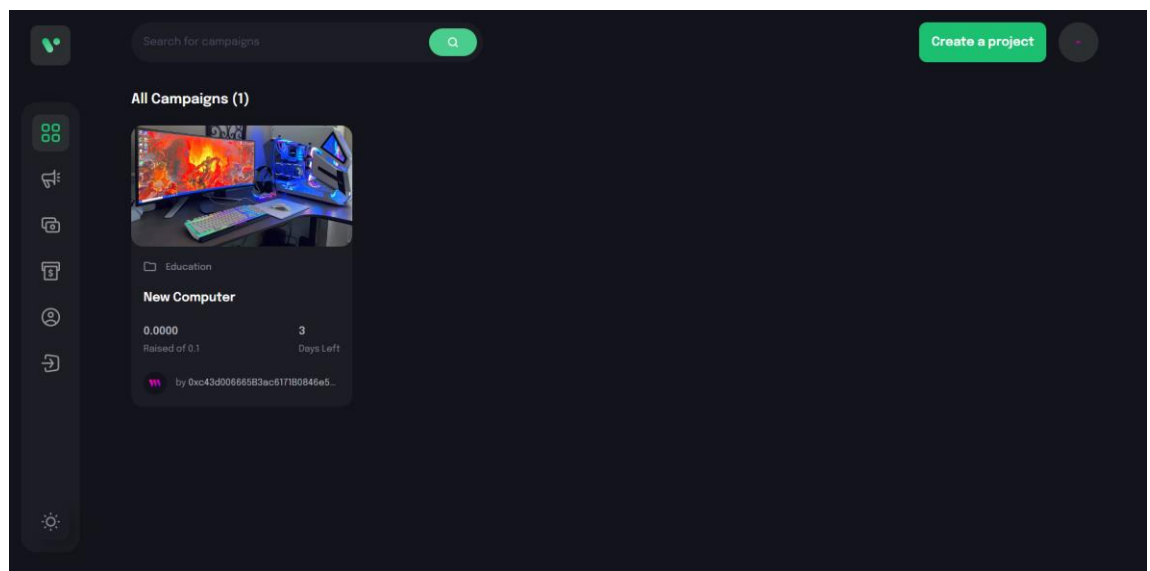


Рисунок 3.12 – Відображення створеного проекту

На головній сторінці платформи всі елементи відображаються коректно, що підтверджує правильну роботу інтеграції між фронтендом і бекендом. Крім того, відображення банера та загальної суми зібраних коштів для кожної

кампанії підтверджує правильну роботу функцій взаємодії з блокчейн-системою. Таким чином, головна сторінка платформи функціонує стабільно, надаючи користувачам усю необхідну інформацію про кампанії та дозволяючи переходити до деталей кожної з них.

Після натискання проєкт, відкривається сторінка проєкту, на якій можна дізнатись більш детальну інформацію, щодо проєкту, а також пожертвувати кошти на проєкт.

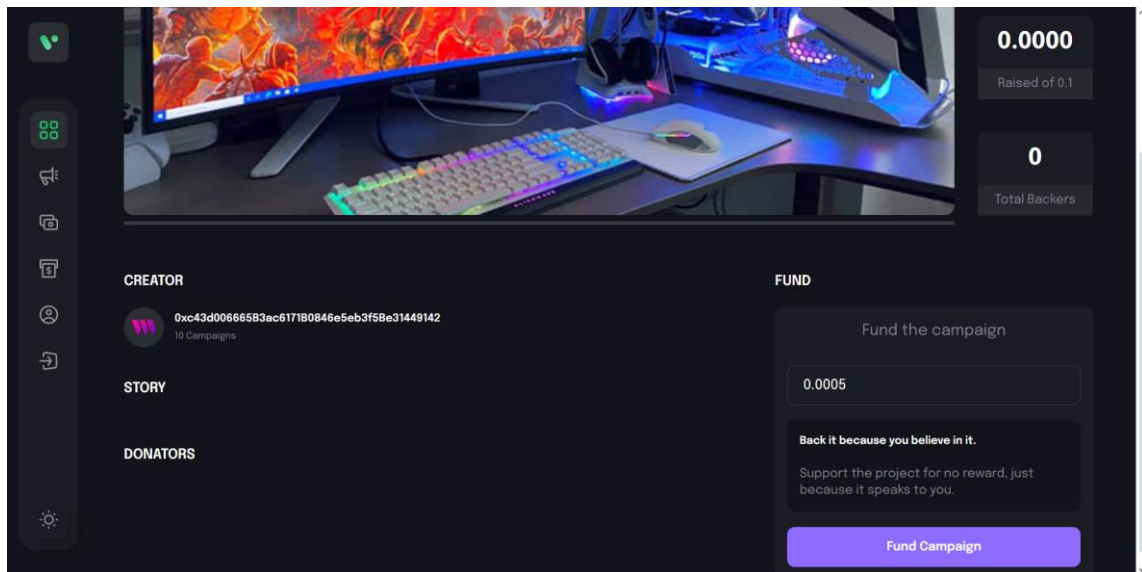


Рисунок 3.13 – Відображення сторінки проєкту

Для тестування функціональності пожертвувань на платформі ми спробуємо здійснити внесок коштів з іншого аккаунту. Це дозволить перевірити правильність обробки транзакцій та забезпечити коректне відображення зібраних коштів на сторінці кампанії, а також перевірити, чи правильно відображаються донатори та їхні внески.

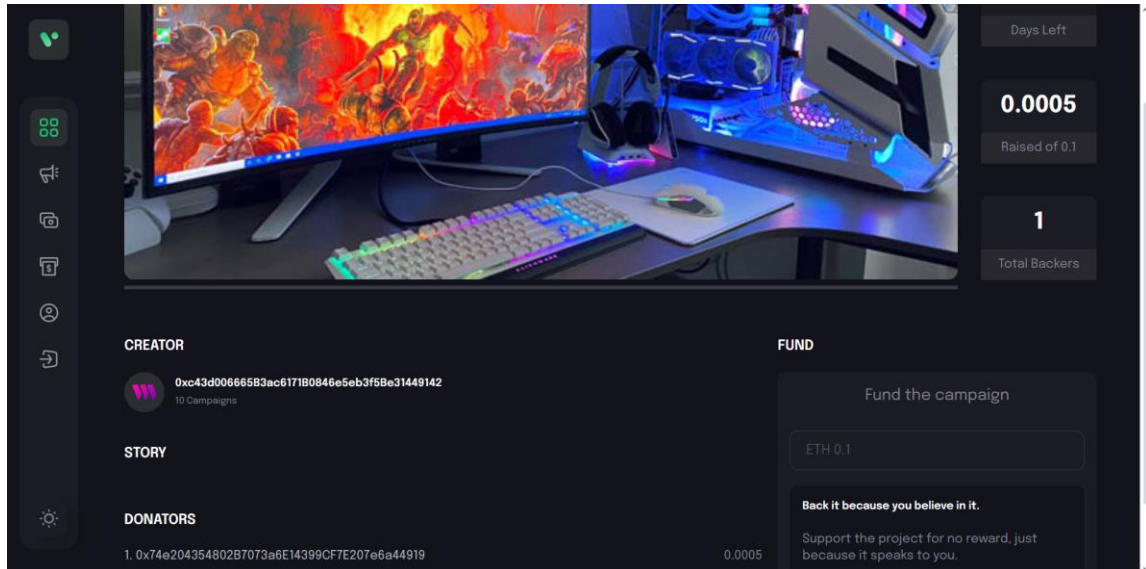


Рисунок 3.14 – Оновлена інформація про проєкт

Після поповнення кампанії коштами відбулося оновлення даних, і на сторінці кампанії з'явилася інформація про донаторів. Зокрема, відобразилась адреса користувача, який здійснив пожертву, а також відповідна сума внеску, що підтверджує правильність обробки транзакцій і коректність відображення зібраних коштів.

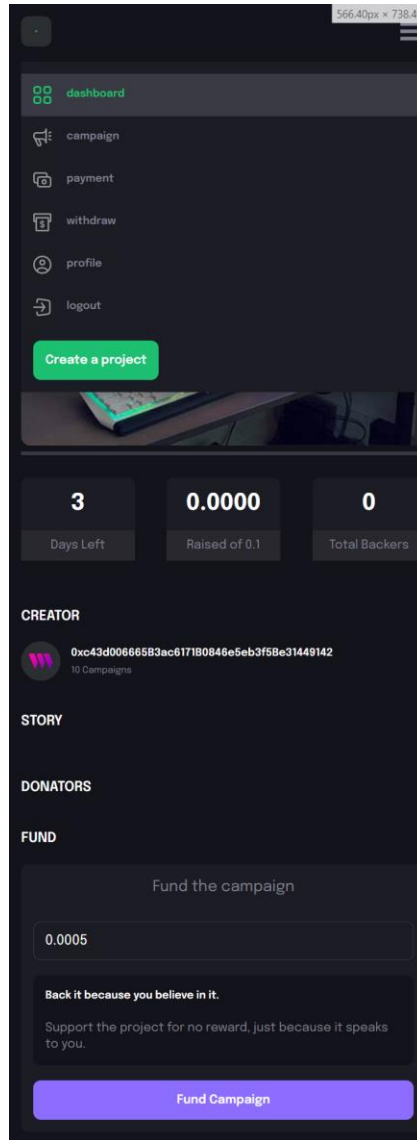


Рисунок 3.15 – Відображення на телефоні

Якщо відкрити сайт з мобільного телефону, він також коректно відобразатиметься завдяки адаптивному дизайну, що автоматично підлаштовується під різні розміри екранів. Всі елементи інтерфейсу залишаються зручними та доступними, навіть на маленьких дисплеях. Крім того, в правому верхньому куті екрану з'являється випадаюче меню, яке дозволяє користувачеві швидко отримати доступ до основних функцій платформи, що значно покращує зручність використання сайту на мобільних пристроях.

ВИСНОВОК

Отже, під час виконання дипломної роботи було досягнуто значних результатів, які стали основою для успішного завершення проєкту. Основна увага була зосереджена на вивченні технологій блокчейну, їхніх можливостей та викликів, а також на аналізі сучасних платформ для реалізації децентралізованих рішень у сфері краудфандингу.

У процесі дослідження було виявлено основні технічні проблеми традиційних платформ краудфандингу, такі як відсутність прозорості, залежність від посередників та тривала обробка транзакцій. Встановлено, що впровадження технології блокчейн здатне усунути ці недоліки завдяки децентралізованій природі, використанню смартконтрактів та прозорій архітектурі. Також були детально проаналізовані потенційні ризики, пов'язані з блокчейн-технологіями, такі як логічні помилки у смартконтрактах, загрози атак Sybil і проблеми із затримками транзакцій, а також розглянуті методи їхнього уникнення.

Окрему увагу було приділено платформі thirdweb, яка надає потужні інструменти для створення Web3-проєктів. У межах роботи були досліджені її можливості для створення NFT-колекцій, децентралізованих автономних організацій, публічних маркетплейсів та блокчейн-ігор. Зроблено висновок, що використання thirdweb значно спрощує процеси розробки та впровадження інноваційних рішень у блокчейн-середовищі.

Таким чином, виконана робота стала важливим етапом у реалізації дипломного проєкту, і практична частина була успішно завершена. Усі заплановані завдання були виконані, що дозволило досягти поставлених цілей і підготувати повністю функціонуючу платформу для децентралізованого краудфандингу на базі технології Web3.

СПИСОК ЛІТЕРАТУРИ

1. The United States Securities and Exchange Commission. Jumpstart Our Business Startups (JOBS) Act. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.sec.gov/rules-regulations/jumpstart-our-business-startups-jobs-act>.
2. CrowdCrux. Kickstarter vs. Indiegogo in 2024: Which Is Best? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.crowdcrux.com/kickstarter-vs-indiegogo/>.
3. Wyrick Robbins. SEC Passes Final Rule to Harmonize and Improve its Exempt Offering Framework. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.wyrick.com/news-insights/sec-passes-final-rule-to-harmonize-and-improve-its-exempt-offering-framework>.
4. GlobeNewswire. Crowdfunding Market to Soar to USD 5.4 Billion by 2032. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.globenewswire.com/news-release/2024/11/18/2982860/0/en/Crowdfunding-Market-to-Soar-to-USD-5-4-Billion-by-2032-The-Rise-of-Alternative-Funding-Solutions-Powers-Market-Growth-Research-by-SNS-Insider.html>.
5. GII Research. Crowdfunding Global Market Report 2024. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.giiresearch.com/report/tbrc1436173-crowdfunding-global-market-report.html>.
6. Indiegogo. Start a Campaign. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.indiegogo.com/start-a-campaign>.
7. Kickstarter. Stats. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.kickstarter.com/help/stats>.
8. Software Advice. Patreon Software Reviews, Pros and Cons. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.softwareadvice.com/membership-management/patreon-profile/reviews/>.

9. The Verge. Patreon creators scramble as payments are mistakenly flagged as fraud. [Электронный ресурс] – Режим доступа до ресурсу: <https://www.theverge.com/2018/8/2/17644404/patreon-payment-declined-subscriber-flagged-fraud>.
10. Popchev I., Radeva I. Decentralized Application (dApp) Development and Implementation. Cybernetics and Information Technologies, 2024. pp.122-123.
11. Next Big Thing AG. Building A Decentralized Future With dApps. [Электронный ресурс] – Режим доступа до ресурсу: <https://nextbigthing.ag/blog/building-a-decentralized-future-with-dapps/>.
12. Zhaobin C., Jingrong H., Chenyang F., Xinyi J. Research on smart contract and front-end technology integration in Dapp development. Academic Journal of Computing & Information Science, 2024. p.57.
13. Use Cases and Comparative Analysis of Blockchain Networks and Layers for DApp Development / Jelena Gjorgjev et al. Mediterranean Conference on Embedded Computing (MECO), 2024.
14. Castell Ferreres G. Development of a DAPP. UPCommons, 2023
15. Yang D., Long C., Xu H., Peng S. A review on scalability of blockchain. International Conference on Blockchain Technology, 2020.
16. Li Y. Application Analysis of Blockchain in Business. International Conference on Economic Development and Business Culture, 2022. pp.1522-1527.
17. Public and private blockchain in construction business process and information integration / Yang R et al. Automation in Construction, 2020.
18. Singh S., Hosen A., Yoon B. Blockchain security attacks, challenges, and solutions for the future distributed iot network. Ieee Access, 2021. pp.13938-13959.
19. Nair P., Dorai D. Evaluation of performance and security of proof of work and proof of stake using blockchain. Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, 2021.
20. PEXX. Blockchains With The Highest Developers Count In 2023. [Электронный ресурс] – Режим доступа до ресурсу:

<https://pexx.com/chaindebrief/blockchains-with-the-highest-developers-count-in-2023/>.

21. Metcalfe W. Ethereum, smart contracts, DApps. Blockchain and Cryptocurrency, 2020. pp.77-93.
22. Kapengut E., Mizrach B. An event study of the ethereum transition to proof-of-stake. Commodities, 2023.
23. Wöhrer M., Zdun U. Devops for ethereum blockchain smart contracts. IEEE International Conference on Blockchain and Cryptocurrency, 2021.
24. Yadav N.; Sarasvathi V. Venturing Crowdfunding using Smart Contracts in Blockchain. International Conference on Smart Systems and Inventive Technolog, 2020.
25. Jadye S., Chattopadhyay S., Khodankar Y., Pati N. Decentralized Crowdfunding Platform Using Ethereum Blockchain Technology. International Research Journal of Engineering and Technology, 2021.
26. Boreiko D. The rise and evolution of Initial Coin Offerings: From genesis to now Understanding Initial Coin Offerings, 2024.
27. Lorenz G. Regulating Decentralized Financial Technology: A Qualitative Study on the Challenges of Regulating DeFi with a Focus on Embedded Supervision. Stanford Journal of Blockchain Law & Policy, 2024.
28. Lazuardy M, Anggraini D. Modern Front End Web Architectures with React.Js and Next.Js. International Research Journal of Advanced Engineering and Science, 2022.
29. Benila S., Crowdfunding using Blockchain. Journals-Global Research and Development Journal For Engineering, 2019.
30. Zhang J., Zha C., Zhang Q. Denial-of-Service Attack Based on Selfish Mining and Sybil Attack in Blockchain Systems. IEEE Access, 2024.
31. Sharma P., Jindal R., Borah M. A review of smart contract-based platforms, applications, and challenges. Cluster Computing, 2023.
32. Andrés J., Lorca P. On the impact of smart contracts on auditing. The International Journal of Digital Accounting Research, 2021. pp. 155-181.

33. Blockchain Support Center. Can my transaction be canceled or reversed? [Электронный ресурс] – Режим доступа до ресурсу: <https://support.blockchain.com/hc/en-us/articles/211162263-Can-my-transaction-be-canceled-or-reversed>.

34. GeeksforGeeks. Full vs Simple Payment Verification in Blockchain. [Электронный ресурс] – Режим доступа до ресурсу: <https://www.geeksforgeeks.org/full-vs-simple-payment-verification-in-blockchain/>.

35. Thirdweb. Mission. [Электронный ресурс] – Режим доступа до ресурсу: <https://thirdweb.com/mission>.

36. Legault M. A practitioner's view on distributed storage systems: Overview, challenges and potential solutions. Technology Innovation Management Review, 2021.

37. Web3Auth. How Web3Auth enabled 300K 100Thieves fans to claim Diamond Chain collectible. [Электронный ресурс] – Режим доступа до ресурсу: <https://blog.web3auth.io/how-web3auth-enabled-300k-100thieves-fans-to-claim-diamond-chain-collectible/>.

38. Almajed R., Abualkishik A., Ibrahim A., Mourad N. Forecasting NFT Prices on Web3 Blockchain Using Machine Learning to Provide SAAS NFT Collectors. Fusion: Practice and Applications, 2023. pp. 55-68.

39. Ciantar M. NFTs, DAOs, and Smart Contracts, Where is the Decentralisation? Int'l JL Changing World, 2023.

40. Thirdweb. The Next Billion Dollar Opportunity in Web3. [Электронный ресурс] – Режим доступа до ресурсу: <https://newsletter.thirdweb.com/p/disruptive-web3-company-ideas>.

ДОДАТОК А

FundraisingPlatform.sol

```
// SPDX-License-Identifier: UNLICENSED
```

```
pragma solidity ^0.8.9;
```

```
contract FundraisingPlatform {
```

```
    struct Project {
```

```
        address creator;
```

```
        string name;
```

```
        string details;
```

```
        uint256 goal;
```

```
        uint256 dueDate;
```

```
        uint256 totalFunds;
```

```
        string banner;
```

```
        address[] contributors;
```

```
        uint256[] contributions;
```

```
    }
```

```
    mapping(uint256 => Project) public projects;
```

```
    uint256 public projectCount = 0;
```

```
    function startProject(address _creator, string memory _name, string  
memory _details, uint256 _goal, uint256 _dueDate, string memory _banner) public  
returns (uint256) {
```

```
        Project storage project = projects[projectCount];
```

```
        require(project.dueDate > block.timestamp, "The dueDate should be a  
date in the future.");
```

```

project.creator = _creator;
project.name = _name;
project.details = _details;
project.goal = _goal;
project.dueDate = _dueDate;
project.totalFunds = 0;
project.banner = _banner;

projectCount++;

return projectCount - 1;
}

function contributeToProject(uint256 _id) public payable {
    uint256 amount = msg.value;

    Project storage project = projects[_id];

    project.contributors.push(msg.sender);
    project.contributions.push(amount);

    (bool sent,) = payable(project.creator).call{value: amount}("");

    if(sent) {
        project.totalFunds = project.totalFunds + amount;
    }
}

function getAllProjects() public view returns (Project[] memory) {
    Project[] memory allProjects = new Project[](projectCount);

```



```

for(uint i = 0; i < projectCount; i++) {
    Project storage item = projects[i];

    allProjects[i] = item;
}

return allProjects;
}

```

```

function getContributors(uint256 _id) view public returns (address[]
memory, uint256[] memory) {
    return (projects[_id].contributors, projects[_id].contributions);
}
}

```

main.jsx

```

import React from "react";
import ReactDOM from "react-dom/client";
import { BrowserRouter as Router } from "react-router-dom";
import { ChainId, ThirdwebProvider } from "@thirdweb-dev/react";

```

```

import { StateContextProvider } from "./context";
import App from "./App";
import "./index.css";

```

```

const root = ReactDOM.createRoot(document.getElementById("root"));

```

```

root.render(

```

```

    <ThirdwebProvider
      activeChain="sepolia"
      clientId="2bc23b8ae3900f8afa77ffed8427dd91">
      <Router>
        <StateContextProvider>
          <App />
        </StateContextProvider>
      </Router>
    </ThirdwebProvider>
  );

```

index.jsx

```

import React, { useContext, createContext } from 'react';

import { useAddress, useContract, useMetamask, useContractWrite } from
 '@thirdweb-dev/react';

import { ethers } from 'ethers';

import { EditionMetadataWithOwnerOutputSchema } from '@thirdweb-
dev/sdk';

const StateContext = createContext();

export const StateContextProvider = ({ children }) => {
  const { contract } =
    useContract('0xB718176901C0173551294698Eb86Fc061f2f171e');
  const { mutateAsync: startProject } = useContractWrite(contract,
    'startProject');

  const address = useAddress();
  const connect = useMetamask();

```

```
const publishProject = async (form) => {
  try {
    const data = await startProject(
      { args: [
        address, // owner
        form.title, // title
        form.description, // description
        form.target,
        new Date(form.deadline).getTime(), // deadline,
        form.image
      ]
    }
  )

  console.log("contract call success", data)
} catch (error) {
  console.log("contract call failure", error)
}
}

const getAllProjects = async () => {
  const campaigns = await contract.call('getAllProjects');

  const parsedCampaigns = campaigns.map((campaign, i) => ({
    owner: campaign.owner,
    title: campaign.title,
    description: campaign.description,
    target: ethers.utils.formatEther(campaign.target.toString()),
    deadline: campaign.deadline.toNumber(),
```

```
ethers.utils.formatEther(campaign.amountCollected.toString()),
    image: campaign.image,
    pId: i
  }));

return parsedCampaigns;
}

const getUserProjects = async () => {
  const allCampaigns = await getAllProjects();

  const filteredCampaigns = allCampaigns.filter((campaign) =>
campaign.owner === address);

  return filteredCampaigns;
}

const donate = async (pId, amount) => {
  const data = await contract.call('donateToCampaign', [pId], { value:
ethers.utils.parseEther(amount)});

  return data;
}

const getContributors = async (pId) => {
  const donations = await contract.call('getDonators', [pId]);
  const numberOfDonations = donations[0].length;

  const parsedDonations = [];
```

```

for(let i = 0; i < numberOfDonations; i++) {
  parsedDonations.push({
    donator: donations[0][i],
    donation: ethers.utils.formatEther(donations[1][i].toString())
  })
}

return parsedDonations;
}

return (
  <StateContext.Provider
    value={{
      address,
      contract,
      connect,
      startProject: publishProject,
      getAllProjects,
      getUserProjects,
      donate,
      getContributors
    }}
  >
    {children}
  </StateContext.Provider>
)
}

export const useStateContext = () => useContext(StateContext);

```