Міністерство освіти і науки України
Харківський національний університет імені В.Н. Каразіна
Навчально-науковий інститут комп'ютерних наук та штучного інтелекту

Рекомендовано до захисту:
протокол засідання кафедри
№ ___ від ____.____.2025 р.
Завідувач кафедри ___ІПСіТ___
_____ _Олег Олешко_
(підпис) (прізвище та ініціали)

Пояснювальна записка

до кваліфікаційної роботи бакалавра

на тему__Розробка децентралізованої системи зберігання та перевірки___
автентичності документів про освіту на основі технології блокчейн

Захищено на засіданні ЕК № _____
протокол № ____ від ____.____.2025 р.
Оцінка _____ / _____
Голова ЕК
_____ ____Фесенко Г.В___
(підпис) (прізвище та ініціали)

Виконав:
студент 4 курсу, групи КС- 42
Спеціальності:
_____122 Комп'ютерні науки____
(шифр і назва спеціальності підготовки)
__Дідич Артем Русланович___
(прізвище, ім'я та по батькові, підпис)
Керівник PhD Родінко М.Ю
(ступень, звання, прізвище та ініціали, підпис)
Рецензент ____д.т.н., професор___
Олійников Р.В._____
(ступень, звання, прізвище та ініціали, підпис)

Харків – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет імені В.Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра інтелектуальних програмних систем і технологій
Рівень вищої освіти (освітньо-кваліфікаційний рівень) бакалавр
Спеціальність F3 Комп'ютерні науки

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри ІПСіТ

_____ Олег ОЛЕШКО
підпис ім'я, прізвище

"____" _____2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дідича Артема Руслановича
(прізвище, ім'я, по батькові студента)

1. Тема роботи: «Розробка децентралізованої системи зберігання та перевірки автентичності документів про освіту на основі технології блокчейн»

керівник роботи PhD, доцент Родінко М.Ю.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від " 16 " квітня 2025 року № 4101-5/962

2. Строк подання студентом роботи 7 червня 2025 року

3. Перелік питань, які потрібно розробити:

Ознайомитись з особливостями технології блокчейн у контексті управління освітніми даними, провести огляд існуючих методів зберігання та верифікації документів про освіту, познайомитися з алгоритмом роботи і методиками розробки смарт-контрактів для автоматизації процесів видачі та перевірки цифрових сертифікатів, створити модель децентралізованої системи управління освітніми сертифікатами, провести програмну реалізацію та експериментальне дослідження розробленої системи.

№ з/п	Назви етапів роботи
1	Формулювання мети та повного переліку завдань для виконання кваліфікаційної роботи (КР).
2	Пошук та аналіз інформаційних джерел за темою КР.
3	Визначення особливостей застосування блокчейн-технологій для управління освітніми даними.
4	Розробка моделей децентралізованої системи видачі та верифікації сертифікатів.
5	Програмна реалізація прототипу системи на основі Ethereum.
6	Проведення тестування та аналізу функціональності розробленої децентралізованої системи.
7	Підсумок отриманих результатів та оформлення пакету документів для процедури захисту КР в екзаменаційній комісії ННІ КН та ШІ.

5. Дата видачі завдання _15.01.2025_

Студент _____ (підпис) А.Р. Дідич (ініціали, прізвище)

Керівник роботи _____ (підпис) М. Ю. Родінко (ініціали, прізвище)

АНОТАЦІЯ

Кваліфікаційна робота містить: 78 сторінок, 17 рисунків, 1 таблицю, 26 джерел.

Метою дослідження є розробка програмного рішення для децентралізованого зберігання та верифікації документів про освіту з урахуванням українських стандартів.

Об'єктом дослідження є процеси створення, зберігання та автентифікації документів про освіту в рамках інтегрованої системи управління освітніми даними.

Предметом дослідження виступає децентралізована система на базі технології блокчейн, що розглядається як інструмент для забезпечення незмінності та прозорості даних при роботі з освітніми сертифікатами.

У роботі розглянуто та описано принципи децентралізованого управління освітніми сертифікатами на базі блокчейну Ethereum, виконано аналіз існуючих централізованих систем і нормативно-правових вимог України. Розроблено програмну реалізацію трирівневої архітектури, що складається зі смарт-контракту на Solidity із середовищем Hardhat, бекенду на Node.js і Express із базою даних MongoDB та фронтенду на React із інтеграцією MetaMask та Ethers.js; реалізовано функції видачі, верифікації й відкликання сертифікатів із модульним логуванням та обробкою помилок; проведено тестування в тестовій мережі Sepolia, виміряно час відгуку REST-API і витрати газу для ключових операцій.

БЛОКЧЕЙН, ETHEREUM, СМАРТ-КОНТРАКТИ, СЕРТИФІКАТИ, ДЕЦЕНТРАЛІЗАЦІЯ

ABSTRACT

The qualification paper contains: 78 pages, 17 figures, 1 table, 26 sources.

The purpose of the research is to develop a software solution for decentralised storage and verification of educational documents based on Ukrainian standards.

The research object is the processes of creating, storing and authenticating educational documents within the framework of an integrated educational data management system

The research subject is a decentralised system based on blockchain technology, which is considered as a tool for ensuring the immutability and transparency of data when working with educational certificates.

The paper considers and describes the principles of decentralised management of educational certificates based on the Ethereum blockchain, analyses the existing centralised systems and regulatory requirements of Ukraine. A software implementation of a three-tier architecture consisting of a smart contract on Solidity with the Hardhat environment, a backend on Node.js and Express with the MongoDB database, and a frontend on React with the integration of MetaMask and Ethers.js was developed; the functions of issuing, verifying and revoking certificates with modular logging and error handling were implemented; testing was carried out in the Sepolia test network, REST-API response time and gas consumption for key operations were measured.

BLOCKCHAIN, ETHEREUM, SMART CONTRACTS, CERTIFICATES, DECENTRALISATION

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ 7
ВСТУП 8
1. ОГЛЯД ТА АНАЛІЗ ПУБЛІКАЦІЙ 11
1.1. Проблеми у сфері управління освітніми даними 11
1.2. Існуючі рішення в освітній сфері 14
1.3. Технологія блокчейн 17
1.4. Децентралізовані рішення на базі блокчейну 20
1.5. Українські стандарти для документів про освіту 27
1.6. Висновки до розділу 29
2. РОЗРОБКА ДЕЦЕНТРАЛІЗОВАНОЇ СИСТЕМИ 31
2.1. Аналіз розв’язуваної задачі 31
2.2. Архітектура системи 34
2.3. Модель смарт-контракту 37
2.4. Бекенд і API 41
2.5. Фронтенд і UI 45
2.6. Відповідність українським стандартам 48
2.7. Висновки до розділу 51
3. ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ 54
3.1. Опис отриманих результатів 54
3.2. Висновки до розділу 70
ВИСНОВКИ 73
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 75
ДОДАТОК А. ПРИКЛАД ВИХІДНОГО КОДУ СМАРТ-КОНТРАКТУ 79

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

БД – база даних;

ЗВО – заклад вищої освіти;

ПІБ – прізвище, ім’я, по батькові;

ABI – двійковий інтерфейс прикладного програмування;

API – прикладний програмний інтерфейс;

ARK – блокчейн-платформа Ark;

CA – центр сертифікації;

CSS – каскадні таблиці стилів;

CORS – механізм доступу до ресурсів між різними доменами;

EBSI – Європейська інфраструктура блокчейн-послуг;

ECDSA – алгоритм цифрового підпису на еліптичних кривих;

ECTS – Європейська система трансферу та накопичення кредитів;

EDCI – Європейська інфраструктура цифрових освітніх сертифікатів;

ENIC – Європейська мережа інформаційних центрів;

ETH – ефір (криптовалюта мережі Ethereum);

gas – умовна одиниця обчислювальних ресурсів у мережі Ethereum;

gwei – одиниця вимірювання газу (1 gwei = 10^{-9} ETH);

JSON – текстовий формат обміну даними;

NFT – невзаємозамінний токен (унікальний цифровий актив);

QR – двовимірний штрихкод;

REST – архітектурний стиль передачі даних у вебі;

RSA – алгоритм шифрування з відкритим ключем;

SSI – модель цифрової ідентифікації на основі самоврядування;

UI – користувацький інтерфейс;

Web3 – децентралізована версія інтернету на основі блокчейн-технологій;

XML – розширювана мова розмітки;

ВСТУП

Актуальність роботи. Сучасний етап розвитку інформаційних технологій характеризується активним впровадженням цифрових рішень у різні сфери суспільного життя, зокрема в освіту. Одним із ключових викликів у цій галузі є забезпечення автентичності документів про освіту, таких як дипломи, сертифікати та атестати. Централізовані системи зберігання даних, що традиційно застосовуються в освітніх установах, мають низку проблем: вразливість до кібератак, можливість підробки документів, а також складність і тривалість процесу верифікації. Ці недоліки стають особливо відчутними в умовах глобалізації, коли зростає потреба в швидкій і надійній перевірці документів на міжнародному рівні. Усе це підкреслює необхідність пошуку нових підходів до управління освітніми даними.

На сьогодні у світі активно досліджуються та впроваджуються децентралізовані технології, серед яких особливе місце посідає блокчейн. Це рішення, що базується на розподіленому реєстрі, забезпечує незмінність, прозорість і безпеку даних завдяки криптографічним методам і децентралізованій архітектурі. Блокчейн вже успішно застосовується в таких галузях, як фінанси, логістика та охорона здоров’я, та має потенціал у сфері освіти. Зарубіжний досвід демонструє приклади інтеграції блокчейну в освітні процеси, але в Україні подібні ініціативи перебувають на початковому етапі, що зумовлює актуальність дослідження та розробки адаптованих рішень, які враховують національні особливості та нормативні вимоги.

Актуальність роботи пояснюється нагальною потребою в ефективних і безпечних системах управління освітніми даними в умовах цифрової трансформації. Проблема підробки документів про освіту залишається серйозним викликом для України, особливо з огляду на інтеграцію в європейський освітній простір. Застосування традиційних методів захисту, таких як водяні знаки чи голограми, не забезпечує достатнього рівня безпеки в цифрову епоху. Натомість блокчейн пропонує принципово новий підхід,

який унеможливлює несанкціоновані зміни даних і спрощує процес верифікації для всіх зацікавлених сторін — від навчальних закладів до роботодавців. Це особливо важливо в контексті міжнародної мобільності студентів і фахівців, коли швидка та достовірна перевірка документів може суттєво вплинути на ефективність освітніх і трудових процесів.

Об’єктом дослідження є процеси створення, зберігання та автентифікації документів про освіту в рамках інтегрованої системи управління освітніми даними

Предметом дослідження виступає децентралізована система на базі технології блокчейн, що розглядається як інструмент для забезпечення незмінності та прозорості даних при роботі з освітніми сертифікатами.

Метою дослідження є розробка програмного рішення для децентралізованого зберігання та верифікації документів про освіту з урахуванням українських стандартів. Для досягнення цієї мети необхідно виконати такі завдання:

1) провести огляд опублікованих джерел за тематикою дослідження;
2) провести аналіз сучасних централізованих та блокчейн-технологій і їхнього застосування в управлінні освітніми даними;
3) дослідити переваги та недоліки існуючих децентралізованих систем;
4) зробити огляд нормативні вимог до документів про освіту в Україні;
5) розробити модель децентралізованої системи видачі та верифікації сертифікатів на базі Ethereum;
6) реалізувати програмне забезпечення та провести його тестування та аналіз функціональності розробленої децентралізованої системи.

Новизна дослідження полягає в розробці децентралізованої системи, адаптованої до специфіки українських документів про освіту. На відміну від зарубіжних аналогів, запропоноване рішення враховує вимоги національного

законодавства, зокрема щодо захисту персональних даних і формату дипломів, а також в перспективі передбачає інтеграцію з існуючими системами, такими як Єдина державна електронна база з питань освіти (ЄДЕБО). Крім того, у роботі реалізовано двомовний інтерфейс, що підвищує доступність системи в міжнародному контексті.

Основні ідеї роботи полягають у створенні програмного рішення з тришаровою архітектурою: смарт-контракти на платформі Ethereum для зберігання даних, серверна частина для обробки запитів і клієнтська частина для зручної взаємодії користувачів. Система забезпечує повний цикл роботи з дипломами — від їхньої видачі до перевірки та відкликання.

Практичне значення дослідження полягає в можливості впровадження розробленого рішення в освітню інфраструктуру українських університетів, що сприятиме підвищенню довіри до документів про освіту, спрощенню їхньої верифікації та зміцненню позицій вищих навчальних закладів країни в глобальному освітньому просторі.

1. ОГЛЯД ТА АНАЛІЗ ПУБЛІКАЦІЙ

1.1. Проблеми у сфері управління освітніми даними

В умовах цифровізації у сучасних освітніх системах накопичуються величезні обсяги даних про навчальні досягнення, дипломи та сертифікати. В таких умовах виникає необхідність надійного зберігання, видачі та верифікації цих документів. Водночас у галузі освіти існують серйозні виклики, пов’язані із фальсифікацією документів, зловживанням даними й неефективними процесами. Підробка академічних записів, несанкціоноване їхнє використання та зміни даних супроводжуються повільними процедурами верифікації, труднощами з контролем доступу та проблемами відповідності систем управління даними потребам користувачів. Ці фактори створюють високі ризики для безпеки інформації й довіри до освітніх документів. Зокрема, через централізовану природу багатьох існуючих реєстрів освітня інформація стає вразливою до шахрайства та неавторизованих змін[1].

Однією з найгостріших проблем залишається поширення підроблених дипломів і сертифікатів. Фальсифікація освітніх документів у глобальному контексті перетворилася на серйозну загрозу: традиційні методи перевірки документів часто не забезпечують належного рівня захисту та вимагають значних зусиль від роботодавців і навчальних закладів[2]. Підробка дипломів разом із пов’язаними з нею шахрайськими схемами перетворилась на поширену проблему. Виходячі з цього традиційні способи видачі документації, які спираються на паперові чи ізольовані електронні реєстри, мають обмежену надійність так як, у таких системах непомітні зміни даних або втручання можуть залишатися непоміченими[3], що ускладнює виявлення підробок і руйнує цілісність реєстрів.

За даними ЮНЕСКО, до 30% роботодавців у деяких країнах стикалися з підробленими дипломами[4]. У 2022 році в США виявлено масштабну аферу з видачею фіктивних дипломів медсестрам — понад 7 600 осіб отримали

недійсні документи, з яких 2 800 були працевлаштовані в медичних установах[5]. В Україні неодноразово фіксувалися факти фальсифікації та покупки дипломів без проходження навчання[6]. Дані випадки підтверджують масштабність проблеми фальсифікації дипломів.

Не менш важливою є складність процедури верифікації дипломів – вона залишається надмірно повільною і бюрократичною. Перевірка автентичності освітнього документа часто передбачає звернення до державних архівів та освітніх реєстрів різних організацій і навіть країн. Як зазначено в аналізі традиційних систем сертифікації, ці процеси вимагають багато часу та ресурсів, особливо коли мова йде про транснаціональні взаємодії. Затримки виникають через потребу підтвердження кваліфікацій через різні бюрократичні процедури: наприклад, легалізацію документів, перевірку статусу закладу освіти чи оцінку відповідності дипломів кваліфікації. Дослідження підтверджують, що тривалі процедури верифікації знижують ефективність системи та можуть створювати корупційні ризики[7].

Сучасні системи обліку освітніх даних зазвичай спираються на централізовані сховища інформації. Хоча цифровізація полегшує накопичення освітньої статистики, вона породжує нові загрози. Центральні бази даних становлять єдиний вузький елемент, який уразливий до зламів, внутрішніх зловживань або технічних збоїв. Відсутність належного розподілу контролю збільшує ризик компрометації даних. При цьому погана сумісність форматів і протоколів між різними навчальними закладами чи регіонами створює інформаційні силоси: дані залишаються фрагментованими, а спільних стандартів для їх інтеграції немає. Все це означає, що освітня інформація про того ж студента може зберігатися одночасно в кількох «закритих» реєстрах, причому обмін та синхронізація цих даних може бути дуже громіздкою. За таких умов навіть просте перенесення даних між установами (наприклад, при вступі на навчання в інший регіон чи країну) є складним через низьку інтероперабельність систем. Важливо відзначити також питання захисту

персональних даних: великі централізовані сховища підвищують ризики витоку чутливих даних студентів і випускників [4].

Перераховані проблеми обґрунтовують потребу в застосуванні нових технологічних і організаційних рішень, які забезпечать вищий рівень прозорості, довіри та безпеки управління освітніми даними. У світовій практиці відзначають перехід від закритих централізованих баз даних до систем, орієнтованих на безпечний розподілений обмін інформацією. Зокрема, ЮНЕСКО підкреслює, що сучасні цифрові технології (наприклад, блокчейн) можуть слугувати «спільним, децентралізованим і захищеним» реєстром для зберігання цифрових ідентичностей, у тому числі освітніх документів. У таких системах кожна зміна фіксується у відкритому журналі трансакцій, що практично виключає можливість непомітної підробки. Також експерти ЄС зазначають, що спільна технічна інфраструктура для цифрових освітніх документів може створити «надійну та стійку до підробки систему», що гарантує конфіденційність даних і дозволяє будь-кому швидко перевірити дійсність документа. Зокрема, планований Європейської комісією проект European Digital Credentials Infrastructure (EDCI) спрямований на автоматичну верифікацію та визнання кваліфікацій в усіх країнах-членах, підвищуючи прозорість і знижуючи адміністративні витрати[8].

Публічні приклади таких підходів вже з’являються у різних країнах. Наприклад, ЮНЕСКО констатує, що пілотні проєкти на базі блокчейну для видачі сертифікатів підтверджують їхню життєздатність у формальній та неформальній освіті[8]. У Європі запровадження єдиної діджитал-інфраструктури зумовлене необхідністю спрощення обміну даними та забезпечення прозорості процедур визнання дипломів. В Україні, наприклад, Міністерство освіти та науки запустило платформу StudyPass, яка використовує технологію блокчейн для зберігання і перевірки автентичності документів про вступ іноземних студентів[9]. Цей та подібні проєкти демонструють розуміння української та світової спільноти необхідності переходу до більш відкритих і безпечних систем зберігання освітніх даних.

1.2. Існуючі рішення в освітній сфері

У традиційних системах документи про освіту видаються на паперовому носії – дипломи, свідоцтва, додатки до дипломів тощо. Такі документи зберігаються в архівах шкіл, коледжів та університетів або у сертифікаційних центрах. Фізичні носії вразливі до пошкодження, втрати та фальсифікацій; до того ж їх архівування потребує багато ресурсів та простору. Втрата диплома внаслідок пожежі, стихійного лиха або бойових дій робить неможливим його перевипуск без значних зусиль. Перевірка достовірності паперових документів зазвичай вимагає звернень до освітнього закладу або державного реєстру, що займає багато часу і ресурсів. Крім того, паперові дипломи відносно легко підробити: відсутність захисних електронних механізмів шифрування чи цифрового підпису ускладнює гарантування їх автентичності. Водночас витрати на фізичне зберігання великого обсягу документів (зокрема в університетських архівах) постійно зростають. Тому вже у середині 1990-х років в Україні були започатковані спроби автоматизувати верифікацію освітніх даних (інформаційна система «Освіта»), пізніше інтегровані до Єдиної державної електронної бази з питань освіти (ЄДЕБО)[10]. Зазначені заходи покликані полегшити пошук та перевірку даних про освіту, проте обмежені централізованою архітектурою.

Нові цифрові рішення використовують електронні формати документів і стандарти для верифікації. У міжнародній практиці поширено впровадження стандарту Open Badges від 1EdTech (IMS Global), який задає структурований формат метаданих для цифрових сертифікатів і значків з використанням криптографічних підписів і сумісного з моделлю «верифікованих атрибутів» W3C[11]. Ряд глобальних платформ (Pearson Credly, Accredible та ін.) пропонують послуги видачі цифрових сертифікатів і бейджів. Ці сервіси зазвичай автоматизують процес випуску та надання сертифікатів: користувач отримує повідомлення на e-mail і може створити обліковий запис на платформі, де його цифрові документи зберігатимуться у вигляді профілю або «цифрового гаманця». Видавач сертифіката прикріплює до нього унікальний

QR-код або посилання, що дозволяє миттєво перевірити автентичність у мережі. Наприклад, у Credly випускник отримує цифрову «бейдж» і зберігає його у своєму профілі, звідки може поділитися ним з роботодавцями чи освітніми платформами [12].

Незважаючи на переваги, деякі аспекти таких платформ мають обмеження. Зокрема, комерційні рішення, як Credly чи Accredible, розроблені переважно для курсів та сертифікацій професійного рівня і не інтегровані з офіційними державними реєстрами дипломів. Дослідження показують, що без застосування додаткових заходів перевірка їхніх цифрових сертифікатів може бути складною: згідно з одним порівнянням, верифікація і безпека в Credly/Accredible ускладнюються тим, що лише деякі користувачі можуть скористатися сучасними методами, такими як блокчейн, для додаткового захисту, а процес підтвердження справжності документів часто довгий і неінтуїтивний[12].

У ряді країн створюють свої централізовані реєстри для перевірки дипломів через API. В Україні ключову роль у цифровізації освітніх документів відіграють ЄДЕБО та мобільний застосунок «Дія». ЄДЕБО збирає дані про освітні заклади і видані дипломи з 2000 року, що забезпечило основу для централізованої перевірки. В Україні функції верифікації реалізовані саме так: інші державні структури чи роботодавці звертаються до ЄДЕБО за допомогою веб-сервісів, щоб підтвердити достовірність освітніх документів. Хоч це й прискорює перевірку в порівнянні з паперовими копіями, проте такий підхід накладає сувору залежність від єдиної системи.

З 22 березня 2024 року українські дипломи масово формуються у цифровому вигляді і синхронізуються із застосунком «Дія» [10]. Новий електронний документ про освіту створюється на підставі записів ЄДЕБО, отримує унікальний ідентифікатор (QR-код чи штрихкод) і доступний як у додатку, так і на порталі «Дія»[13]. Наприклад, повна верифікація диплома в «Дії» здійснюється миттєво за допомогою сканування QR-коду – система автоматично порівнює метадані диплома із записами ЄДЕБО. Завдяки цим

рішенням український уряд надає цифровим освітнім документам ту ж юридичну силу, що й паперовим екземплярам[10].

Основними перевагами сучасних електронних систем є зручність і швидкість верифікації, що суттєво скорочують час підтвердження кваліфікацій. Замість багатоденного звернення до архіву паперових документів, достатньо передати QR-код або реєстраційний номер, за яким автоматично витягується перевірена інформація. Цифрові документи легко зберігати й відновлювати у разі фізичного ушкодження, що особливо актуально для воєнного часу та надзвичайних ситуацій [13]. Окрім того, у багатьох системах активується подвійний захист: наприклад, мобільний застосунок «Дія» вимагає авторизації через мобільний банк чи електронний підпис, щоб заборонити доступ до документів стороннім особам.

Незважаючи на безперечні переваги, цифрові рішення також мають обмеження. По-перше, оскільки всі цифрові документи централізовано зберігаються у державній системі, відсутність доступу до інтернету або будь-які технічні збої в ЄДЕБО/«Дії» роблять верифікацію недоступною. Таким чином, хоча хмарні рішення прискорюють і спрощують перевірку освітніх документів, вони вимагають централізованої інфраструктури, а централізована аутентифікація означає, що всі запити проходять через один контролюючий орган, що створює єдину точку відмови і є вразливими до потенційних збоїв і атак. По-друге, централізація систем зберігання вимагає високого рівня захисту персональних даних та надійного резервного копіювання: як зазначено в порівняльному дослідженні, централізовані рішення створюють «єдину точку відмови» й уразливість до атак та технічних збоїв [14]. По-третє, стандарти інтеграції різних платформ ще далекі від єдиних: наприклад, освітні заклади та третя сторона потребують API або погоджених форматів (JSON, XML, стандарт Open Badges тощо) для взаємодії. Хоча багато платформ дотримуються відкритих стандартів (зокрема Open Badges для сертифікатів), відсутність єдиного всіма прийнятого формату може ускладнювати сумісність. По-четверте, цифрові сервіси, як правило,

надаються комерційними чи державними структурами і можуть бути платними або обмежені регіонально. Наприклад, українська система «Дія» поки доступна лише громадянам України, а її міжнародне визнання потребуватиме синхронізації з базами інших країн. Нарешті, залежність від електронних платформ вимагає певного рівня цифрової грамотності в освітян та працедавців.

Отже, можна прийти до висновку, що цифрові освітні платформи відіграють ключову роль у трансформації процесів управління освітніми документами, проте їх подальший розвиток вимагає підвищення рівня безпеки, стандартизації та переходу до більш надійних моделей зберігання й верифікації.

1.3. Технологія блокчейн

Традиційні централізовані системи зберігання академічних документів уразливі: за наявності посередника вони часто повільні, дублюють дані та схильні до фальсифікацій або кібератак (розділи 1.1 та 1.2)[10]. У відповідь на ці виклики технологія блокчейн пропонує новий підхід. Це розподілений реєстр транзакцій, що зберігає інформацію в послідовно пов’язаних блоках на основі криптографічних алгоритмів. Кожен блок містить геш попереднього, тому будь-яка зміна у вже збереженому записі призведе до невідповідності та виявлення спроби підробки. Крім того, кожна транзакція підписується цифровим підписом відправника (з використанням пари відкритого і приватного ключів), що підтверджує авторство і справжність запису. Такі криптографічні методи роблять записи блокчейну стійкими до підробки і несанкціонованих змін [12]. Таким чином, дані в блокчейні фактично незмінні і прозорі: всі учасники мережі мають доступ до історії транзакцій, але жоден з них не може скасувати або стерти раніше записане [13]. Усе це забезпечує довіру до мережі без централізованого управління.

У блокчейн-системі учасники узгоджують оновлення реєстру за допомогою консенсусу. Тобто більшість вузлів мережі мають погодитися з тим, що нова транзакція дійсна, перед тим як вона потрапить до наступного блоку. Для реалізації консенсусу використовують різні алгоритми. Найпоширеніші – Proof-of-Work (PoW) та Proof-of-Stake (PoS) [12]. У PoW-мережах, як-от у першому блокчейні Bitcoin, вузли («майнери») розв'язують складні криптографічні задачі; той, хто швидше знайде рішення, додає новий блок і отримує нагороду. У PoS-мережах, до яких відноситься нові версії Etherium, етап додавання блоку визначається розміром і тривалістю володіння криптовалютою (stake) – чим більше монет і чим довше вони заблоковані в мережі, тим вищі шанси стати валідатором блоку. Такий підхід значно знижує енергоспоживання системи порівняно з PoW. Незалежно від механізму, консенсус гарантує остаточність транзакцій: після додавання транзакції до блоку її неможливо змінити чи видалити без внесення нової виправної транзакції, що також буде зафіксовано у реєстрі [11].

Смарт-контракти – це програмні контракти з попередньо заданими умовами, що автоматично виконуються у мережі блокчейн. Вони дозволяють реалізувати бізнес-логіку без участі сторонніх посередників: при настанні визначених подій смарт-контракт самостійно генерує транзакції, змінюючи стан реєстру за заздалегідь прописаними правилами [14]. У контексті освіти смарт-контракти можуть автоматизувати адміністративні процеси – наприклад, реєстрацію здобуття студентами певних модулів або видачу сертифікатів, – забезпечуючи при цьому довіру і прозорість. Як зазначають дослідники, впровадження смарт-контрактів упорядковує документообіг і підвищує безпеку: учасники довіряють автоматизованим правилам, а записи про досягнення студентів фіксуються послідовно та незмінно [15].

Усі ці властивості роблять блокчейн особливо привабливим для зберігання і верифікації освітніх даних. По-перше, незмінність записів гарантує, що диплом, атестат чи сертифікат про навчання буде неможливо фальсифікувати після їх внесення в мережу. Навіть якщо навчальний заклад

припинить існування або станеться збій у локальних базах даних, інформація про видані дипломи й надалі буде верифікована за записами у блокчейні [11].

По-друге, децентралізація усуває єдину точку відмови: мережа з багатьма вузлами гарантує, що дані не опиняться «в одних руках» і з ними не зможуть маніпулювати. Співпраця вузів і інших установ може відбуватися напряму через блокчейн без залучення посередників [13].

По-третє, прозорість блокчейну створює довіру до академічних документів. Освітні записи у відкритому або дозволеному блокчейні доступні для перевірки будь-яким зацікавленим сторонам (наприклад, роботодавцям або університетам-контрагентам) без додаткових довідок від видавця. Це означає, що після видачі диплома установа більше не мусить відповідати на кожен запит щодо його достовірності – кожен може самостійно перевірити диплом у ланцюзі блоків [15].

По-четверте, блокчейн забезпечує високий рівень безпеки даних. Криптографічна захищеність гарантує, що видавати дипломи можуть тільки уповноважені особи – приватний ключ вишу підписує документ, а кожен може перевірити цей підпис за відкритим ключем установи. Самі транзакції записуються у мережу так, що їх майже неможливо змінити «заднім числом» [12]. Одночасно блокчейн може впорядкувати ідентифікацію учасників освітнього процесу: наприклад, студент із власним приватним ключем сам контролює доступ до своїх персональних даних, а сторонні вузи обмежують свої запити до перевірки підписаних записів [14].

Загалом технологія блокчейн надає універсальний набір механізмів – децентралізований розподілений реєстр, криптографічний захист, алгоритми консенсусу та смарт-контракти – що суттєво підвищують довіру, прозорість та безпеку освітніх даних. Наприклад, концепція системи цифрових дипломів на блокчейні передбачає токенізацію дипломів (часто у формі NFT) з автоматизованою перевіркою їх достовірності. У результаті сформована модель видачі дипломів стає «невтомним бухгалтером», який зберігає

вірогідні записи про досягнення студентів, доступні у будь-який час і в будь-якому місці без потреби у посередниках [15].

1.4. Децентралізовані рішення на базі блокчейну

Блокчейн – це розподілена база даних з архівом транзакцій у вигляді послідовних блоків, які неможливо підробити після підтвердження. Існують три основні типи мереж блокчейну: публічні, приватні та консорціумні. У публічних блокчейнах будь-хто може приєднатися, читати й записувати дані в мережу, що забезпечує максимальну прозорість і децентралізацію. Прикладом є Bitcoin або Ethereum – криптовалюти з відкритим кодом і глобальними вузлами. Натомість у приватних блокчейнах мережу контролює одна організація чи обмежена група, учасники мають права лише при запрошенні, а оператор може змінювати або видаляти записи при необхідності. Вони не настільки децентралізовані (часто нагадують розподілену базу даних), але можуть забезпечити високу швидкість і конфіденційність даних при жорсткому контролі доступу[16]. Консорціумні блокчейни формують комбінацію публічних і приватних: їх утримують кілька відомих організацій (наприклад, декілька університетів або державних установ), що спільно беруть участь у консенсусі. Вони забезпечують баланс між відкритістю та контролем – присутність кількох власників усуває єдиного централізованого оператора, а обмежена кількість вузлів спрощує механізм підтвердження транзакцій і знижує комісію[17].

Для освітньої сфери особливо важливо обирати тип мережі залежно від сценарію. Публічний блокчейн гарантовано незмінність дипломів чи сертифікатів після занесення їх хешів в ланцюг, а будь-хто (наприклад, роботодавці чи інші університети в світі) може самостійно перевірити справжність документу за публічним блокчейном. Однак такі мереж мають високі транзакційні витрати (наприклад, плата за газ у Ethereum). Приватний блокчейн, навпаки, дає змогу зберігати академічні записи за високою

швидкістю і з обмеженим колом довірених вузлів (наприклад, в межах одного університету чи освітнього департаменту), але це рішення вимагає довіри до оператора мережі та несе ризик централізації (адміністратор може змінювати дані)[16]. Консорціумний блокчейн з'єднує переваги обох підходів – кілька університетів чи державних установ ведуть спільну мережу, у якій жодна сторона не має монопольного контролю. Це дуже доцільно для обміну академічними даними між закладами: мережа виконує підтвердження швидше, ніж публічний блокчейн, при цьому зберігаючи високу безпеку завдяки обмеженій кількості валідаторів і наперед погодженим правилам. Однак організація консорціуму складніша: всі учасники повинні домовитися про протоколи взаємодії та оновлення мережі[17]. У сфері освіти консорціумні мережі особливо корисні при необхідності міжуніверситетської або міждержавної верифікації дипломів – наприклад, коли університети різних країн створюють спільну мережу для автоматичного підтвердження курсів і заліків студентів.

Технічно впровадження кожного типу блокчейну різниться. Публічні мережі (Bitcoin, Ethereum) використовують децентралізований консенсус (раніше – Proof-of-Work, в сучасних версіях Ethereum – Proof-of-Stake), високий ступінь реплікації (кожен вузол зберігає весь ланцюг блоків) і відкриту книгу транзакцій. Дані у публічному блокчейні є прозорими: наприклад, якщо оприлюднити хеш диплому, будь-хто може виконати команду з перевірки цього хешу в блоках. Однак через високі комісії та час підтвердження такі мережі часто використовують лише для фіксації короткого цифрового «відбитку» документу, а не повного вмісту. Приватні блокчейни (наприклад, побудовані на Hyperledger Fabric) використовують централізовані або напівцентралізовані механізми (Raft, PBFT) і дозволяють контролювати доступ за сертифікатами (MSP – Membership Service Providers). У такому ланцюгу транзакції записуються у блоки та одночасно оновлюються в локальній базі стану (World State). Доступ до реєстру можуть мати лише вузли організацій, що беруть участь, а дані можуть зберігатися у вигляді відкритих

текстів або шифрованих записів, залежно від правил. Консорціумні блокчейни використовують змішаний підхід: наприклад, у моделі Delegated Proof-of-Stake (DPoS) учасники мережі (скажімо, декани університетів) обирають довірених делегатів для валідації блоків, а усі кандидати на приєднання проходять перевірку (довідки про існування в реальних реєстрах). Прикладом є платформа EduCTX, де на основі блокчейну Ark (DPoS) було побудовано мережу університетів: кожному студенту виділяється унікальна адреса з токенами ECTX за курси, і тільки за допомогою мультипідпису (двохз підписів – студент і університет) ці токени можуть переміщуватись[18]. Такий підхід гарантує анонімність студента (на записах лише блокчейн-адреса) і водночас запобігає несанкціонованому використанню «кредитів» без підтвердження університету.

Розглянемо конкретні блокчейн-платформи та їх роль у децентралізованому зберіганні освітніх документів. Ethereum – публічна смарт-контрактова платформа. Вона підтримує загальнодоступний ланцюг блоків та віртуальну машину (EVM) для виконання кодів контрактів. Будь-який вузол може зчитувати і виконувати контракти, що створює високий рівень прозорості. Децентралізовану верифікацію диплому на Ethereum реалізують зазвичай шляхом збереження хешу сертифікату або повідомлення про його видачу у транзакції. Прикладом є OpenCerts у Сінгапурі: освітні заклади формують відкритий JSON-документ диплому, обчислюють з нього криптографічну хеш-функцію і заносять цю «цифрову відбитку» у публічний ланцюг Ethereum. Будь-хто може потім через веб-портал або мобільний додаток подати цей документ на перевірку: сервіс обчислює хеш і порівнює з тим, що записано у блоках. Оскільки майже неможливо підробити вміст диплома так, щоб збігся хеш, це гарантує цілісність документу. Публічний характер Ethereum означає, що для внесення інформації потрібна комісія («gas»), але не потрібно тримати власні вузли – перевірка відбувається публічно, без централізованих посередників[19][20]. Водночас у відкритому

етері дані блокчейну доступні всім, тому на ланцюг записують тільки неперсоніфіковані дані (лише хеш, без особистих полів).

Hyperledger – це проєкт Linux Foundation, що надає кілька модулів для власних блокчейн-мереж. Найпоширеніший – Hyperledger Fabric. Fabric розрахований на дозвільні (permissioned) рішення. Тут кожна організація отримує сертифікат від мережевого CA і може запускати вузли Peer, які зберігають свою копію ланцюга і «світовий стан» (World State). Смарт-контракти (chaincode) виконуються у Docker-контейнерах і зміни даних реєстру затверджуються відповідно до політики (наприклад, вузли декількох університетів повинні підписати результат). Fabric дозволяє сегментувати мережу на канали або приватні зони – кожен канал матиме свій розподілений ланцюг, тож зберігання записів може бути роздільним для різних груп. Такий підхід добре підходить, коли дипломи чи транскрипти потрібно зберігати у внутрішній мережі консорціуму – наприклад, кілька університетів можуть створити спільний блокчейн, в якому лише вузли-партнери бачать та підтверджують випуск документів. Децентралізована верифікація у Fabric може бути реалізована через запити до smart-контрактів: наприклад, програма-верифікатор запитує запис про студента (по його унікальному ідентифікатору чи ключу) і отримує підтвердження з блокчейну, підписане відповідними вузлами. На відміну від публічних платформ, Hyperledger не має відкритого загальнодоступного перегляду – це більш контрольована модель з високою продуктивністю та приватністю.

Як було розглянуто раніше, EduCTX – це академічна платформа на базі блокчейну для управління кредитами і заліками (ECTS) у вищій освіті. Вона реалізована на основі форку ARK Blockchain (механізм DPoS), але адаптована до консорціумного режиму. Архітектура EduCTX передбачає, що кожен університет стає вузлом-учасником мережі (директори програми чи декани виконують роль делегатів у консенсусі). Кожному студенту генерується анонімна блокчейн-адреса, на яку виписуються токени ECTX за успішне завершення курсів. Для захисту від зловживань застосовано протокол

мультипідписів: щоб студент не міг вивести свої кредити без дозволу університету, адреса студента прив’язана до 2 із 2 підписів (студент + його університет). Таким чином, навіть якщо студент володіє закритим ключем, провести транзакцію кредитів можна лише за участі університетського вузла. Усі записи в мережі публічні і незмінні, а будь-хто може перевірити запис про заліки конкретного студента (через блок-експлорер EduCTX) – це й реалізує децентралізовану верифікацію результатів навчання. Також код платформи відкритий і доступний на GitHub, а приєднання нових учасників мережі вимагає попередньої домовленості та підтвердження ідентичності (наприклад, підпису декана).[18] Таким чином, EduCTX ілюструє приклад технічної реалізації консорціумного блокчейну для освіти, де університети утримують мережу, а студенти контролюють свої кредитні записи.

Щодо реалізації верифікації дипломів на практиці, існує кілька сучасних прикладів. Blockcerts – це відкрита специфікація для створення, видачі й перевірки цифрових освітніх сертифікатів на базі блокчейну[21]. Основна ідея полягає в тому, що офіційні документи (дипломи, сертифікати, резюме навичок тощо) формуються у стандартизованому форматі (JSON), який цифрово підписується закритим ключем закладу-видавця, а потім хеш цього документа записується на блокчейн. Мережа блокчейну забезпечує неможливість змінити запис без відкриття відповідних блоків. Стек Blockcerts складається з кількох модулів: генератор сертифікатів (Issuer), який створює і підписує документ; мобільний додаток–гаманець (Wallet), де власник збирає свої сертифікати; універсальний верифікатор (Verifier), що підтверджує достовірність документів; а також фронтенд-сервіси для демонстрації сертифікатів[22]. Всі компоненти – open source, що дає можливість розробникам будувати індивідуальні рішення на основі цього стандарту. Блокчейном-платформою може бути будь-яка публічна мережа: Blockcerts спочатку підтримував Bitcoin, а потім Ethereum, прагнучи бути «агностичним» щодо конкретного ланцюга[21]. На практиці для дипломів часто

використовують Ethereum або біткойн, оскільки вони найстабільніші і найбільше захищені.

Криптографічна складова Blockcerts передбачає цифровий підпис сертифікату за допомогою приватного ключа видавця (часто RSA або ECDSA), а на блокчейн записується коротка хеш-сумма документа. Це робить дані невідтворними: перевіряючи диплом, зовнішній сервіс обчислює хеш отриманого документа та порівнює з записом у блоках. Якщо навіть один символ оригінального документу змінити, хеш не співпаде – і веріфікатор сповістить про невідповідність. Блокчейн при цьому виконує роль «чекового документу»: всі транзакції публічні і незмінні, тому жоден учасник не може приховати факт видачі чи змінити виданий сертифікат. У Blockcerts передбачені також механізми анулювання (revocation) помилково виданих сертифікатів або випуску нової версії пакета документів[21].

На прикладі MIT видно, як Blockcerts застосовується на практиці: Массачусетський технологічний інститут випустив близько 100 перших цифрових дипломів через мобільний додаток Blockcerts Wallet – і для цього використав блокчейн Bitcoin (через такий вибір надавали пріоритет безпеці мережі). Цей приклад демонструє, що провідні університети вважають за доцільне використати публічні блокчейни для надійної перевірки дипломів: власник диплома отримує зашифрований цифровий документ, який може у будь-який момент самостійно поділитися з роботодавцем чи іншим закладом, а останній – через вбудовану кнопку перевірки — зможе переконатися у справжності, звернувшись до блокчейну. Саме тому стандарти Blockcerts підкреслюють самовладну ідентичність (SSI): отримувач має повний контроль над своїм цифровим дипломом і над тим, кому його надати[22].

Ще одним сучасним прикладом робочої системи є згаданий вище OpenCerts – державний проєкт Сінгапуру, що використовує Ethereum для блокчейн-верифікації академічних документів. У межах OpenCerts випускники сінгапурських вузів отримують криптографічно захищені цифрові сертифікати, а копія звіту надходить в блокчейн у вигляді хешу. Реєстрація

хешу сертифіката у ланцюзі дає змогу будь-кому перевірити актуальність диплому без необхідності через офіційні канали. Керівники проекту зазначають, що завдяки Ethereum (де багато вузлів і сильна підтримка розробників) верифікація може відбуватися миттєво і без вартості сторонньої послуги. Сам диплом формально «належить» випускнику, який зберігає його файл або адресу у гаманці; при показі документу перевіряючий бачить статус «Revoked» тільки у разі, коли видавець офіційно скасував сертифікат. OpenCerts вже впроваджено в багатьох установах Сінгапуру – за даними на 2020 рік, більше 18 інститутів видавали дипломи через цей блокчейн-сервіс[19][20]. Уряди інших країн також зацікавлені у таких рішеннях, оскільки подібні системи помітно спрощують боротьбу з фальсифікацією документів.

Україна теж приєдналася до світових ініціатив у цьому напрямі: у 2022 році вона стала не-ЄС спостерігачем у Європейському блокчейн-партнерстві (EBP), яке курує створення Європейської інфраструктури блокчейнових сервісів (EBSI)[23]. Це відповідає загальноєвропейській тенденції: EBSI реалізує проєкти з цифрових дипломів (Formal Accreditation and Recognition) та інших публічних сервісів, об’єднуючи університети різних країн для спільного верифікування ступенів і сертифікатів. У рамках цих ініціатив стандарти, наприклад, пропонують виражати дипломи у форматі Verifiable Credentials і зберігати їх в ланцюгах Ethereum із використанням європейського фреймворку W3C-VC. Фактичне включення України у такі пілоти ще перебуває на стадії домовленостей, але вищезгадані кроки демонструють готовність держави до впровадження подібних технологій.

Таким чином, децентралізовані блокчейн-рішення відкривають нові можливості для зберігання та перевірки освітніх документів. Кожен тип мережі має свої переваги та обмеження: публічні мережі забезпечують абсолютну незмінність і вседоступність записів, але мають уразливі місця у масштабованості та приватності; приватні блокчейни гарантують контроль і конфіденційність у межах організації, але потребують довіри до оператора;

консорціумні ланцюги поєднують швидкість і приватність із спільним управлінням (скажімо, між університетами), проте їх впровадження ускладнене узгодженням учасників. При цьому концепція децентралізованої верифікації документів здатна значно скоротити бюрократію: розробники використовують смарт-контракти, мультипідписи, криптографічні хеші та відкриті стандарти (як у Blockcerts) для створення універсальних рішень. Ефективне застосування цих технологій вже доведено на прикладі Ethereum (OpenCerts), біткойну/ethereum (Blockcerts) та спеціалізованих проєктів (EduCTX), які спрощують переклад академічних досягнень у надійний цифровий формат. За наявності державної підтримки (як, наприклад, через EBSI) та співпраці університетів, такі системи мають потенціал зробити освітні документи безпечними, прозорими й доступними для миттєвої перевірки у будь-якій точці світу.

1.5. Українські стандарти для документів про освіту

Документи про здобуття вищої освіти (диплом молодшого бакалавра, бакалавра, магістра, доктора філософії/мистецтва) видаються за встановленими законодавством формами та містять конкретний набір обов’язкових реквізитів. Згідно з Законом України «Про вищу освіту», у дипломі зазначаються найменування закладу освіти й освітньої програми, ступінь вищої освіти та здобута кваліфікація (спеціальність, галузь знань, у разі потреби – спеціалізація чи професійна кваліфікація)[24]. Також обов’язково вказуються прізвище та ім’я випускника (від 2021 р. без по батькові), рік закінчення навчання та найменування ЗВО, серія (буква та цифри) і реєстраційний номер диплома, який присвоюється при реєстрації у Єдиній державній електронній базі з питань освіти (ЄДЕБО). Крім того, диплом повинен містити посаду та підпис керівної особи (зазвичай ректора чи уповноваженої особи) з вказанням її прізвища та ініціалів, печатку ЗВО і дату видачі[25]. Усі ці відомості дублюються українською й англійською мовами.

У дипломі доктора філософії/мистецтва, окрім того, зазначаються ступінь, галузь знань та спеціальність (для міждисциплінарних робіт – декілька галузей і спеціальностей), а також назва закладу та наукової установи, де відбувалася підготовка й захист (наукові здобутки)[24].

Не менш важливо, що невід'ємною частиною дипломів молодшого бакалавра, бакалавра, магістра та доктора філософії/мистецтва є додаток європейського зразка – диплом Supplement. У додатку структуровано наводяться дані про результати навчання, освітні компоненти, отримані оцінки, здобуті кредити ECTS та інформація про національну систему вищої освіти України[24]. Нині усі дипломи державного зразка реєструються в ЄДЕБО[25]. З 2020 року дипломи видаються лише за акредитованими програмами – у самому дипломі зазначається назва органу акредитації, а в додатку – реквізити акредитаційних сертифікатів.

Форми та зовнішній вигляд бланків затверджуються Міністерством освіти (Наказ МОН №102 від 25.01.2021) і мають державний характер[26]. ЗВО можуть відрізнятися дизайном (колір, матеріал, поліграфія), однак зміст диплома стандартизований. Так, на офіційних сайтах (ENIC Ukraine) наводяться зразки заповнення дипломів та вказується, що дизайн бланка і захисні елементи затверджує кожен заклад самостійно. Мета реєструвати дипломи в ЄДЕБО та друкувати їх на державних бланках – унеможливити видачу недійсних документів. Для прикладу, ENIC Ukraine роз'яснює, що державний диплом повинен містити серію й номер (з ЄДЕБО), дані про ЗВО і кваліфікацію, підпис та печатку (в обох мовах)[25].

У цифровій системі ці стандарти можна відтворити за допомогою смарт-контрактів і структурованих метаданих. Наприклад, кожен виданий диплом може представлятися в блокчейні записом, що містить зашифровані чи хешовані поля – ПІБ випускника, ступінь, спеціальність, номер диплома, рік та назву ЗВО, а також унікальний ідентифікатор транзакції. Смарт-контракт може автоматично перевіряти коректність введених реквізитів (формат номера, наявність підписів, строків акредитації тощо) та фіксувати випуск

документа у блокчейні, забезпечуючи послідовність транзакцій і прозорість записів. При цьому кожна транзакція містить криптографічний хеш даних документу, що робить записи незмінними: будь-яка спроба підробки чи зміни інформації легко виявиться, оскільки дані не можна «перезаписати» в ланцюзі. Використання блокчейну гарантує децентралізовану верифікацію: будь-який роботодавець чи навчальний заклад зможе перевірити автентичність диплому, порівнявши опубліковану хеш-інформацію з поданим документом. Таким чином структура блокчейну (децентралізація, транзакції з цифровими підписами, незмінність блоків) природно реалізує вимоги щодо захисту автентичності і цілісності даних про освіту[3].

Отже сучасне українське законодавство чітко регламентує, які саме реквізити мають міститися у дипломах (ступінь, спеціальність, ПІБ, серії/номери, підписи, печатка тощо). Ці вимоги легко формалізуються у блокчейн-рішенні: у вигляді смарт-контрактів зі структурованими метаданими, де кожен запис (диплом) є незмінним та криптографічно захищеним.

1.6. Висновки до розділу

Проведений огляд літературних джерел, нормативних актів та аналіз існуючих технічних рішень у сфері зберігання і верифікації документів про освіту засвідчив наявність глибоких проблем у централізованих системах управління освітніми даними. Традиційні підходи, зокрема паперові дипломи або централізовані реєстри, демонструють вразливість до підробок, обмежену доступність для перевірки та низьку інтероперабельність у глобальному освітньому середовищі. Навіть попри впровадження цифрових сервісів, як-от ЄДЕБО чи мобільного додатку «Дія», питання прозорості, захисту від шахрайства та довготривалої незмінності даних залишається відкритим.

Блокчейн, як децентралізована технологія із вбудованою підтримкою незмінності, прозорості та автоматизації через смарт-контракти, відкриває

нові можливості для цифровізації документів про освіту. Огляд наукових публікацій та прикладів реальних проєктів (Blockcerts, OpenCerts, EduCTX) підтверджує практичну життєздатність блокчейн-рішень у сфері освіти. Ці системи дозволяють автоматизувати процеси видачі, перевірки та анулювання дипломів, при цьому зберігаючи їхню достовірність і забезпечуючи контроль доступу до персональних даних. Вони не лише підвищують ефективність взаємодії між університетами, студентами та роботодавцями, але й зменшують адміністративне навантаження на освітні установи.

Втім, ключовою прогалиною залишається відсутність платформи, що враховує специфіку українських нормативних вимог до документів про освіту. Існуючі закордонні рішення не адаптовані до національних стандартів, таких як форма диплома, обов'язковий склад реквізитів, особливості додатків європейського зразка та вимоги до реєстрації в ЄДЕБО. Більше того, більшість систем створено у рамках пілотних ініціатив окремих університетів чи компаній і не передбачають масштабованої інтеграції на державному рівні. Відсутність єдиної державної децентралізованої платформи для верифікації освітніх документів, адаптованої до українських стандартів, створює унікальне вікно можливостей для реалізації інноваційного рішення, що поєднує переваги блокчейну з вимогами національного освітнього середовища.

Таким чином, проведене дослідження підтвердило актуальність і перспективність використання блокчейн-технологій для вирішення проблем автентифікації освітніх документів. Встановлені нормативні обмеження можуть бути технічно реалізовані засобами смарт-контрактів і захищених транзакцій. Водночас ідентифіковані обмеження та невирішені питання у наявних платформах створюють логічне підґрунтя для подальшої розробки децентралізованої системи, здатної забезпечити довіру, безпеку і відповідність українським освітнім стандартам.

2. РОЗРОБКА ДЕЦЕНТРАЛІЗОВАНОЇ СИСТЕМИ

2.1. Аналіз розв’язуваної задачі

Перед практичною реалізацією системи доцільно здіснити детальний аналіз розв’язуваної задачі, що включає оцінку обмежень, опис загальних особливостей системи та декомпозицію задачі на основні складові, необхідні для її реалізації. Такий підхід дозволить сформувати чітке розуміння викликів і вимог, які необхідно врахувати на етапі проектування, та створити основу для подальшого викладу в наступних частинах роботи.

Розробка децентралізованої системи на базі технології блокчейн для управління освітніми сертифікатами є складним завданням, яке потребує врахування як технічних особливостей обраної платформи, так і функціональних вимог до системи. Оскільки основою реалізації обрано публічний блокчейн Ethereum, аналіз задачі має враховувати його специфіку. Водночас система повинна відповідати нормативним стандартам, зокрема українському законодавству щодо документів про освіту, що додає додатковий рівень складності. Таким чином, аналіз розв’язуваної задачі є необхідним етапом для виявлення ключових проблем, визначення сильних сторін технології та структурування подальшої розробки.

Одним із основних аспектів, які необхідно розглянути, є обмеження, пов’язані з використанням блокчейну Ethereum. Першим таким обмеженням є доступність і швидкість обробки транзакцій. У публічному блокчейні час підтвердження транзакцій залежить від завантаженості мережі, що може призводити до затримок. У середньому блок у мережі Ethereum підтверджується за 12–15 секунд, однак у періоди пікового навантаження цей час може значно зростати. Для системи, яка має забезпечувати оперативну видачу та верифікацію сертифікатів, такі затримки можуть негативно впливати на користувацький досвід, особливо якщо врахувати очікування швидкого доступу до результатів з боку кінцевих користувачів, таких як студенти чи роботодавці. Крім того, кожна операція в Ethereum

супроводжується сплатою комісії за газ — плати за обчислювальні ресурси, необхідні для виконання транзакцій. Це означає, що видача, верифікація чи відкликання сертифіката пов’язані з фінансовими витратами, які необхідно оптимізувати. Наприклад, неефективно спроектований смарт-контракт може призвести до надмірного споживання газу, що зробить систему економічно невигідною для масштабного використання.

Переходячи до загальних особливостей системи, варто відзначити, що децентралізація є однією з ключових переваг використання технології блокчейн. Відсутність центрального органу управління усуває єдину точку відмови, що підвищує стійкість системи до збоїв і зовнішніх атак. У контексті освітніх сертифікатів це означає, що дані не можуть бути втрачені через збій сервера чи маніпуляції з боку одного учасника. Кожен запис у блокчейні є незмінним і може бути перевірений будь-ким, що забезпечує високий рівень довіри до системи.

Ще однією важливою особливістю є захист даних. Завдяки використанню криптографічних методів, таких як хеш-функції та цифрові підписи, система гарантує, що сертифікати не можуть бути підроблені або змінені без виявлення таких дій. Наприклад, хеш сертифіката, записаний у блокчейн, дозволяє перевірити його автентичність шляхом порівняння з оригінальним документом, а будь-яка спроба модифікації призведе до невідповідності хешів. Це особливо важливо для освітніх документів, достовірність яких відіграє ключову роль для цільових користувачів системи, таких як роботодавці або представники навчальних закладів.

Прозорість системи також відіграє значну роль. Можливість публічної верифікації сертифікатів усуває потребу в посередниках і спрощує процес перевірки. Наприклад, роботодавець може самостійно перевірити автентичність диплома, ввівши його ідентифікатор у систему, без звернення до навчального закладу чи державних органів. Це зменшує бюрократичні затримки та підвищує ефективність взаємодії між сторонами.

Крім того, система має відповідати українським стандартам для документів про освіту. Це включає врахування вимог до структури сертифікатів, таких як наявність серійного номера, назви навчального закладу, програми навчання тощо, а також забезпечення захисту персональних даних відповідно до національного законодавства. Адаптація до цих стандартів є необхідною умовою для практичного впровадження системи в Україні.

Для реалізації поставленої задачі її необхідно розбити на основні компоненти, що дозволить спростити розробку та забезпечити модульність системи. Першим компонентом є зберігання даних у блокчейні. Це передбачає розробку смарт-контракту, який буде містити структуру даних сертифіката, включаючи унікальний ідентифікатор, серійний номер, ім’я студента, рік випуску, назву навчального закладу та інші необхідні поля. Смарт-контракт має підтримувати функції додавання нових сертифікатів, їхньої верифікації та, за потреби, відкликання, наприклад, у разі виявлення помилок чи анулювання документа. Оптимізація смарт-контракту є критично важливою для зменшення витрат на газ і забезпечення економічної ефективності системи.

Другим компонентом є верифікація сертифікатів. Система має надавати можливість перевіряти автентичність документа за його ідентифікатором або серійним номером. Для цього в смарт-контракті необхідно реалізувати відповідні функції, які повертатимуть дані сертифіката або підтверджуватимуть його відсутність. Швидкий доступ до результатів верифікації має бути забезпечений через бекенд, який виступатиме посередником між блокчейном і користувацьким інтерфейсом.

Третім компонентом є створення інтерфейсу користувача. Інтерфейс має бути зручним і доступним для різних категорій користувачів: адміністраторів навчальних закладів, які видають і відкликають сертифікати, студентів, які отримують доступ до своїх документів, і роботодавців, які перевіряють їхню автентичність. Він має підтримувати основні операції — видачу, перевірку та відкликання сертифікатів — і бути адаптованим до українських стандартів, зокрема щодо мови та формату даних. Наприклад, інтерфейс може включати

форми для введення даних сертифіката, кнопку для генерації хешу та відображення результатів верифікації.

Четвертим компонентом є інтеграція з бекендом, який забезпечує взаємодію між фронтендом і смарт-контрактом через API. Бекенд має обробляти запити користувачів, валідувати введені дані, виконувати транзакції в блокчейні та повертати результати. Наприклад, при видачі сертифіката бекенд отримує дані від адміністратора, формує транзакцію для запису в блокчейн і повідомляє про успішне завершення операції. Цей компонент є ключовим для забезпечення безперебійної роботи системи та її масштабування.

Таким чином, аналіз розв’язуваної задачі дозволив виявити основні обмеження, пов’язані з доступністю, швидкістю транзакцій, витратами на газ і масштабуванням. Ці виклики потребують ретельного врахування на етапі проектування, щоб забезпечити ефективність і практичність системи. Водночас загальні особливості, такі як децентралізація, захист даних, прозорість і відповідність стандартам, підтверджують перспективність використання блокчейну для освітніх цілей. Декомпозиція задачі на компоненти — зберігання даних, верифікацію, інтерфейс і інтеграцію — створила чітку структуру для подальшої розробки, яка буде детально розглянута в наступних підрозділах роботи.

2.2. Архітектура системи

У попередніх частинах пояснювальної записки (підрозділ 2.1) було проведено аналіз задачі створення децентралізованої системи зберігання та верифікації документів про освіту, визначено її ключові вимоги, обмеження та здійснено декомпозицію. На основі цього аналізу розроблено архітектуру системи, яка має забезпечити ефективну реалізацію поставлених цілей: гарантувати незмінність даних, надійність верифікації та зручність використання для користувачів. Архітектура є основою для подальшої

реалізації системи, оскільки визначає принципи організації її компонентів, їхню взаємодію та інтеграцію, що безпосередньо впливає на продуктивність, безпеку та можливість масштабування.

Тришарова архітектура обрана як основа системи через її здатність чітко розподіляти функціональні обов'язки між компонентами, що сприяє модульності, спрощує підтримку та дозволяє адаптувати систему до майбутніх змін. Смарт-контракт відповідає за децентралізоване зберігання даних у блокчейні, бекенд забезпечує обробку запитів і зв'язок із блокчейном, а фронтенд надає користувацький інтерфейс для взаємодії із системою. Такий підхід є оптимальним для інтеграції децентралізованих технологій із традиційними веб-рішеннями, дозволяючи поєднати переваги блокчейну, такі як незмінність і прозорість, із гнучкістю централізованих компонентів. Кожен шар системи детально розглянуто нижче з акцентом на його роль, технічну реалізацію та внесок у загальну функціональність.

Першим шаром є смарт-контракт, який розгорнуто на блокчейні Ethereum і виконує функцію децентралізованої бази даних для зберігання інформації про освітні сертифікати. Смарт-контракт розроблено мовою Solidity і включає структури даних для запису сертифікатів, а також функції для їхньої видачі, верифікації та відкликання. Основною перевагою цього компонента є забезпечення незмінності та автентичності даних: після запису в блокчейн інформація стає захищеною від несанкціонованих змін, що підвищує довіру до системи. Для оптимізації роботи смарт-контракту використано хешування частини даних сертифікатів, що дозволяє зменшити обсяг даних, які зберігаються в блокчейні, та знизити витрати на газ — ключовий фактор, враховуючи високу вартість транзакцій у публічних мережах. Крім того, смарт-контракт включає механізми контролю доступу, які обмежують можливість виконання операцій лише авторизованими користувачами, такими як представники навчальних закладів. Це забезпечує безпеку та відповідність вимогам управління даними.

Другим шаром системи є бекенд, реалізований на основі Node.js із використанням фреймворку Express. Бекенд виконує функцію посередника між фронтендом і смарт-контрактом, обробляючи запити користувачів і забезпечуючи їхню передачу в блокчейн. Він містить RESTful API із ключовими ендпоінтами: видача сертифіката (/issue), верифікація (/verify) та відкликання (/revoke). Кожен ендпоінт відповідає за підготовку даних, їхню валідацію та формування транзакцій для смарт-контракту. Наприклад, при створенні сертифіката бекенд отримує вхідні дані, генерує хеш і відправляє транзакцію в блокчейн через бібліотеку ethers.js, яка забезпечує зручну взаємодію з Ethereum. Додатково бекенд інтегровано з базою даних MongoDB для зберігання метаданих, таких як тимчасові коди верифікації, що дозволяє уникнути перевантаження блокчейну та підвищити швидкість обробки запитів. Цей шар відіграє критичну роль у спрощенні роботи з блокчейном, абстрагуючи складність транзакцій для користувачів і забезпечуючи швидкий доступ до даних.

Третім шаром є фронтенд, розроблений із використанням бібліотеки React і інструменту Vite, що забезпечує швидке створення інтерактивного інтерфейсу. Фронтенд включає основні сторінки для взаємодії із системою: видача сертифікатів, їхня верифікація та відкликання, реалізовані у вигляді компонентів React (Issue.jsx, Verify.jsx, Revoke.jsx). Інтерфейс підтримує двомовність (українська та англійська) завдяки бібліотеці i18next, що відповідає вимогам локалізації та міжнародного використання. Для інтеграції з блокчейном використано гаманець MetaMask, який дозволяє користувачам підписувати транзакції безпосередньо в браузері, зберігаючи контроль над приватними ключами. Такий підхід підвищує безпеку та зручність, оскільки усуває необхідність у додаткових інструментах для взаємодії із системою. Фронтенд також відображає результати операцій, наприклад, статус верифікації сертифіката, що робить систему інтуїтивно зрозумілою для кінцевих користувачів.

Тришарова архітектура забезпечує низку переваг, зокрема модульність і масштабованість. Кожен шар може бути вдосконалений незалежно: наприклад, смарт-контракт можна оптимізувати для зменшення витрат на газ, а фронтенд — оновити для покращення користувацького досвіду, не впливаючи на інші компоненти. Однак реалізація такої структури пов’язана з певними викликами. Одним із них є синхронізація даних між блокчейном і MongoDB, оскільки затримки в обробці транзакцій можуть призвести до тимчасової невідповідності. Для вирішення цієї проблеми використано кешування на бекенді, що дозволяє прискорити доступ до часто запитаних даних і зменшити навантаження на блокчейн. Іншим важливим аспектом є безпека: смарт-контракт захищено модифікаторами доступу, бекенд, а фронтенд покладається на MetaMask для безпечного підпису транзакцій.

Таким чином, розроблена тришарова архітектура системи, що включає смарт-контракт, бекенд і фронтенд, забезпечує надійне зберігання, верифікацію та управління освітніми сертифікатами. Схема взаємодії між шарами демонструє чіткий розподіл функцій і логіку обробки запитів, що робить систему прозорою та ефективною. Цей підхід відповідає сучасним вимогам до децентралізованих систем і може бути адаптований до потреб українського освітнього сектору, зберігаючи при цьому потенціал для подальшого розвитку.

2.3. Модель смарт-контракту

Першим етапом створення системи стала розробка моделі смарт-контракту CertificateSystem.sol, для реалізації основних операцій із сертифікатами, зокрема їхньої видачі та перевірки.

Смарт-контракт CertificateSystem.sol написаний мовою Solidity і розгорнутий на блокчейні Ethereum, який обрано завдяки його розвиненій екосистемі та підтримці смарт-контрактів. Ethereum забезпечує децентралізоване середовище, де дані зберігаються у вигляді незмінного

ланцюжка блоків, що гарантує їхню достовірність і доступність. Смарт-контракт слугує основним інструментом для роботи з освітніми сертифікатами, надаючи структури даних і функції для їхнього управління. Він розроблений з урахуванням принципів економічної ефективності та безпеки, що є важливим для роботи в публічному блокчейні, де кожна операція пов’язана з витратами на газ.

Структура даних у смарт-контракті визначає, як інформація про сертифікати зберігається в блокчейні. Кожен сертифікат представлений набором полів, серед яких унікальний ідентифікатор (certificateId), серійний номер (seriesNumber), ім’я студента (studentName), рік випуску (graduationYear) і назва навчального закладу (university). Для забезпечення двомовності, що відповідає вимогам локалізації, ці текстові поля підтримують представлення українською та англійською мовами (studentNameUk, studentNameEn, universityUk, universityEn). Також додано поля для програми навчання (programNameUk, programNameEn), ступеня (degree), кодів спеціальності (fieldCode, specialtyCode, specializationCode) та хеш транзакції (txHash), що дозволяє відстежувати операцію в блокчейні. Також використано поле isRevoked, яке вказує, чи був сертифікат відкликаний, що є важливим індикатором для управління статусом документа. Для оптимізації пошуку за серійним номером реалізовано мапінг seriesToId, який дозволяє швидко визначити ідентифікатор сертифіката за хешем серійного номера.

Функція видачі сертифіката (issueCertificate) є основним механізмом додавання нових записів до блокчейну. Ця функція доступна лише авторизованим користувачам, наприклад, представникам навчальних закладів, і приймає вхідні параметри. На етапі виконання функція генерує хеш серійного номера за допомогою keccak256 і записує його разом із іншими даними у блокчейн. Інші поля, такі як ім’я студента чи назва закладу, зберігаються у відкритому вигляді для зручності верифікації. Після успішного запису смарт-контракт генерує подію CertificateIssued, яка містить ідентифікатор сертифіката, серійний номер і статус, дозволяючи зовнішнім

системам відстежувати оновлення. Витрати на газ для цієї операції становлять приблизно 600 000 одиниць, що є гарним показником для такого типу транзакцій у мережі Ethereum.

Функціональність смарт-контракту також включає функцію setTxHash, яка дозволяє зберегти хеш транзакції після виконання операції, забезпечуючи можливість подальшого відстеження через інструменти, такі як Etherscan. Для верифікації сертифікатів реалізовано дві функції: verifyCertificate перевіряє сертифікат за ідентифікатором (certificateId), а verifyBySeriesNumber — за серійним номером, використовуючи хеш для пошуку через мапінг seriesToId. Додатково функція getSeriesHash повертає хеш серійного номера для зовнішньої перевірки. Оскільки ці функції є читальними і не змінюють стан блокчейну, вони не потребують сплати газу, що робить їх безкоштовними для користувачів. Функція відкликання revokeCertificate дозволяє авторизованим користувачам позначити сертифікат як відкликаний, змінюючи поле isRevoked, і генерує подію CertificateRevoked із відповідною інформацією. Це забезпечує можливість управління статусом сертифіката в разі виявлення помилок або. Безпека смарт-контракта є критично важливою для забезпечення довіри до системи. Для цього використано модифікатор onlyOwner із бібліотеки OpenZeppelin, який обмежує виконання функцій, таких як issueCertificate, setTxHash і revokeCertificate, лише власником контракту, наприклад, адміністратором навчального закладу. Це запобігає несанкціонованому додаванню чи зміні даних. Додатково реалізовано перевірку на дублювання серійних номерів: функція issueCertificate блокує створення сертифіката, якщо серійний номер уже існує в мапінгу seriesToId, що виключає помилки. Для функції revokeCertificate додано захист від повторного відкликання сертифікатів: якщо поле isRevoked уже встановлено в true, операція відхиляється, що запобігає некоректним діям. Криптографічні методи, зокрема хешування з використанням keccak256, гарантують цілісність даних серійного номера, унеможливлюючи підроблення. Кожна транзакція

супроводжується цифровим підписом відправника, що підтверджує його автентичність.

Оптимізація витрат на газ є одним із пріоритетів при розробці CertificateSystem.sol. Хешування серійного номера дозволяє зменшити витрати на запис у блокчейн, але інші дані зберігаються у відкритому вигляді, що спрощує верифікацію і знижує витрати на обчислення при перевірці. Мапінг seriesToId забезпечує швидкий пошук сертифікатів за серійним номером, уникаючи повного сканування даних, що підвищує продуктивність. Код функцій оптимізовано для мінімізації обчислень: функція issueCertificate виконує лише необхідні операції хешування та запису, а читальні функції, такі як verifyCertificate, уникають змін стану, усуваючи потребу в оплаті газу. Така стратегія робить смарт-контракт економічним для використання.

Для забезпечення надійності смарт-контракту реалізовано комплексне тестування, яке охоплює всі основні функції. Тести перевіряють коректність хешування серійних номерів, обробку помилок, таких як спроби дублювання чи відкликання вже відкликаних сертифікатів, а також контроль доступу, зокрема роботу модифікатора onlyOwner. Окрім цього, надають необхідні дані для оцінки газової ефективності. Такі тести гарантують стабільність і безпеку роботи контракту в реальних сценаріїв.

Отже, результатом виконання цього єтапу стала модель смарт-контракту CertificateSystem.sol, яка забезпечує надійний і прозорий механізм управління освітніми сертифікатами в створеній системі. Функція видачі дозволяє безпечно додавати нові сертифікати, зберігаючи їх у вигляді хешів для економії ресурсів і захисту даних, тоді як функція перевірки надає швидкий і безкоштовний спосіб верифікації, що відповідає потребам користувачів. Механізми безпеки, такі як контроль доступу та криптографія, гарантують цілісність і автентичність інформації, а оптимізація витрат на газ робить систему практичною для масштабного використання. Цей смарт-контракт створює міцну основу для подальшої інтеграції з іншими компонентами системи, що буде розглянуто в наступних розділах.

2.4. Бекенд і API

Для реалізації повноцінної взаємодії користувачів із системою необхідний компонент, який би спрощував доступ до блокчейну, забезпечував обробку запитів і гарантував коректність введених даних. Таким компонентом у розробленій системі є бекенд, побудований на основі технології Node.js із використанням фреймворку Express. Бекенд виступає посередником між фронтендом і смарт-контрактом, надаючи API для виконання ключових операцій, таких як видача, верифікація та відкликання сертифікатів. У цьому підрозділі детально розглядається структура бекенду, зокрема основний файл server/index.js, описуються основні ендпоінти API та аналізується процес валідації даних, який відіграє ключову роль у забезпеченні безпеки та надійності системи.

Бекенд розроблено з урахуванням вимог до швидкості, масштабованості та безпеки, що є критично важливими для системи, призначеної для обробки даних про освітні сертифікати. Вибір Node.js як основної платформи зумовлений її асинхронною архітектурою, яка дозволяє ефективно обробляти численні запити від користувачів. Використання фреймворку Express додатково спрощує створення RESTful API та організацію маршрутів. Основним файлом бекенду є server/index.js, який виконує роль центрального вузла системи. У цьому файлі здійснюється ініціалізація сервера, підключення необхідних бібліотек, таких як ethers.js версії 5.7.2 для взаємодії з блокчейном, а також конфігурація середовища. Зокрема, у server/index.js налаштовано підключення до тестової мережі Sepolia, що дозволяє проводити тестування системи в умовах, максимально наближених до реальних, без необхідності використання реальних коштів. Такий підхід забезпечує стабільність і контрольованість розробки.

API бекенду реалізовано за принципами RESTful, що гарантує простоту інтеграції з фронтендом і зручність використання. Основними ендпоінтами API є /issue, /verify і /revoke, які відповідають за ключові функції системи: видачу сертифікатів, їхню верифікацію та відкликання. Додатково реалізовано ендпоінти /codes для отримання кодів спеціальностей і напрямів та /degrees для отримання списку ступенів освіти, що полегшує введення даних у фронтенді. Кожен із цих ендпоінтів визначено як окремий маршрут у файлі server/index.js, що сприяє чіткому розподілу функціональності та полегшує подальше масштабування коду. Наприклад, ендпоінт /issue обробляє POST-запити, які містять дані сертифіката, такі як серійний номер, ім’я студента, рік випуску та назва навчального закладу. Після успішної валідації ці дані передаються в смарт-контракт для запису в блокчейн. Ендпоінт /verify обробляє GET-запити, отримуючи ідентифікатор сертифіката або його серійний номер і повертаючи інформацію про статус сертифіката та його автентичність на основі даних із блокчейну. Ендпоінт /revoke призначений для авторизованих користувачів, які можуть змінити статус сертифіката, позначивши його як відкликаний. Усі відповіді форматуються в JSON і підтримують CORS для коректної інтеграції з фронтендом.

Одним із найважливіших аспектів роботи бекенду є валідація даних, яка гарантує, що лише коректні та безпечні дані будуть передані в блокчейн. Для цього використано власні механізми валідації через регулярні вирази та перевірки. Наприклад, при обробці запиту на видачу сертифіката через ендпоінт /issue бекенд перевіряє серійний номер на відповідність формату 1 буква, 2 цифри, символ №, 6 цифр, а імена студентів — на відповідність наявності кирилиці для українських імен, та латиниці для англійських. Також перевіряються коди спеціальностей і напрямів, діапазони значень (наприклад, рік випуску) та довжина полів. У разі невідповідності хоча б одному з критеріїв бекенд повертає помилку з HTTP-кодом 400 (Bad Request) і детальним описом проблеми, що підтримує двомовність (uk/en). Цей механізм

запобігає запису некоректних даних у блокчейн, що є особливо важливим з огляду на незворотність транзакцій.

Аутентифікація в системі базується на Web3-технологіях із використанням гаманця MetaMask. Користувачі автентифікуються через підпис транзакцій у MetaMask, а перевірка прав доступу здійснюється на рівні смарт-контракту через модифікатор onlyOwner. Наприклад, лише власник контракту може виконувати операції через /issue або /revoke. У разі недостатності прав бекенд повертає помилку з кодом 403 (Forbidden), що унеможливлює несанкціоновані дії. Такий підхід забезпечує захист системи від зловмисних дій і відповідає принципам розмежування доступу.

Інтеграція бекенду з блокчейном здійснюється через бібліотеку ethers.js, яка забезпечує зручний і надійний інтерфейс для роботи зі смарт-контрактом. У файлі server/index.js визначено підключення до мережі Sepolia за допомогою провайдера, а також створено екземпляр смарт-контракту на основі його ABI та адреси розгортання. Це дозволяє бекенду викликати функції контракту, такі як issueCertificate для запису нового сертифіката чи verifyCertificate для перевірки його статусу. Реалізовано функцію setTxHash для збереження хешу транзакції після виконання операції, а також кешування хешів транзакцій у пам’яті для прискорення доступу до них. Оскільки транзакції в блокчейні є асинхронними, бекенд обробляє їх відповідним чином: після відправлення транзакції через ethers.js він чекає її підтвердження в мережі, використовуючи механізм промісів, і лише після цього повертає результат користувачеві. Для підвищення продуктивності використано асинхронну обробку запитів, що дозволяє ефективно працювати з великою кількістю користувачів.

Бекенд інтегровано з базою даних MongoDB для централізованого зберігання кодів спеціальностей і ступенів освіти в єдиній колекції Code, де різниця між типами кодів визначається полем type (значення 'field' для кодів спеціальностей і 'degree' для кодів ступенів). Схема даних у MongoDB має наступний вигляд: об’єкт включає поля type (рядок), code (код спеціальності чи ступеня), uk і en (назви на українській і англійській мовах), а для типу 'field'

додатково містить масив specialties із підспеціальностями, кожна з яких має власний код і назви. Використання MongoDB у системі включає розшифровку сертифікатів через функцію expandCertificate, яка асинхронно запитує відповідні записи (Code.findOne) за кодами fieldCode і degree, а також валідацію кодів, наприклад, Code.findOne із перевіркою наявності specialtyCode у масиві specialties. Основні функції MongoDB у системі охоплюють локалізацію назв, валідацію коректності кодів, підтримку ієрархічної структури для спеціальностей і швидкий доступ до даних завдяки індексації. Перевагами цього підходу є централізоване зберігання, мультимовність, гнучкість оновлення кодів і їхня розшифровка під час верифікації, коли в блокчейні зберігаються лише коди, а повні назви витягуються з MongoDB.

Обробка помилок у бекенді є деталізованою та структурованою. Реалізовано систему кодів помилок: валідаційні помилки повертають код 400, помилки мережі — 500, а помилки авторизації — 403. Повідомлення про помилки підтримують двомовність (uk/en), що полегшує їх інтерпретацію користувачами. Формат помилок адаптовано для зручної обробки на фронтенді, включаючи код і текстовий опис.

Додаткові особливості бекенду включають логування часу виконання транзакцій, що дозволяє аналізувати продуктивність системи, підтримку двомовності для всіх полів (uk/en), валідацію кодів спеціальностей і напрямів, а також перевірку на дублювання серійних номерів перед їхньою видачею. Бекенд також обробляє відкликані сертифікати, повертаючи відповідний статус при верифікації. Безпека забезпечується через ретельну перевірку формату всіх вхідних даних, валідацію довжини полів, діапазонів значень і захист від дублювання сертифікатів, а контроль доступу делеговано смарт-контракту.

Оптимізація роботи бекенду включає кешування хешів транзакцій у пам’яті, асинхронну обробку запитів, валідацію даних перед відправленням у блокчейн для уникнення зайвих транзакцій, а також оптимізацію газових

витрат через коректну підготовку даних для смарт-контракту. Це забезпечує високу продуктивність і економічність системи.

Таким чином, бекенд і API відіграють центральну роль у забезпеченні функціональності розробленої системи. Структура файлу server/index.js чітко відображає організацію сервера, маршрутів і взаємодії з блокчейном та MongoDB. Ендпоінти /issue, /verify, /revoke, /codes і /degrees надають користувачам зручний інтерфейс для роботи з сертифікатами, а ретельна валідація даних гарантує їхню коректність і безпеку. Інтеграція з ethers.js, кешування в пам’яті та використання MongoDB для кодів роблять бекенд ефективним і масштабованим рішенням, яке створює міцну основу для взаємодії з фронтендом, розгляд якого буде проведено в наступному підрозділі.

2.5. Фронтенд і UI

Для повноцінної взаємодії користувачів із системою необхідний зручний і інтуїтивно зрозумілий інтерфейс, який би дозволив їм легко виконувати основні операції: видачу, верифікацію та відкликання освітніх сертифікатів. Цю функцію виконує фронтенд системи, розроблений з використанням сучасних веб-технологій. Фронтенд є ключовим компонентом, який забезпечує доступність системи для кінцевих користувачів, таких як адміністратори навчальних закладів, студенти та роботодавці.

Фронтенд системи побудовано на основі бібліотеки React, яка є однією з найпопулярніших технологій для створення інтерактивних користувацьких інтерфейсів. Вибір React зумовлений його компонентною архітектурою, що дозволяє розбивати інтерфейс на незалежні модулі, які легко підтримувати та розширювати. Для прискорення процесу розробки та збірки використано інструмент Vite, який забезпечує швидке оновлення коду в режимі реального часу. Стилізація компонентів реалізована за допомогою Tailwind CSS, що надає гнучкість у налаштуванні дизайну, а також інтегровано Material-UI для

використання готових компонентів із сучасним дизайном. Додатково в проєкті застосовуються бібліотеки framer-motion для анімацій, react-hook-form для управління формами, axios для HTTP-запросов до бекенду та ethers.js для роботи з Ethereum, що розширює функціональні можливості фронтенду. Такий технологічний стек є оптимальним для проєкту, оскільки поєднує високу продуктивність, зручність розробки та сучасні стандарти веб-дизайну, що відповідає вимогам до зручності та естетики користувацького інтерфейсу.

Структура фронтенду організована за принципом модульності, що полегшує його розробку та підтримку. Основними компонентами є CertificateForm.jsx, який відповідає за видачу сертифікатів, CertificateView.jsx для перегляду та верифікації сертифікатів, і RevokeCertificate.jsx для відкликання сертифікатів. Кожен із цих компонентів інкапсулює свою логіку та інтерфейс, що дозволяє розробникам працювати над окремими частинами системи незалежно. У CertificateForm.jsx реалізовано складну валідацію з використанням регулярних виразів, зокрема спеціальної маски для введення серійного номера, а також синхронізацію полів університету на різних мовах (uk/en). Після заповнення форми і натискання кнопки "Видати сертифікат" компонент відправляє запит на бекенд через API, використовуючи axios, який, у свою чергу, взаємодіє зі смарт-контрактом для запису даних у блокчейн. Аналогічно, CertificateView.jsx надає можливість користувачам ввести ідентифікатор сертифіката або його серійний номер для отримання інформації про його статус і автентичність. Така організація компонентів забезпечує чіткий розподіл функціональності та спрощує подальше розширення системи.

Одним із ключових аспектів фронтенду є його інтеграція з гаманцем MetaMask, який дозволяє користувачам взаємодіяти з блокчейном Ethereum безпосередньо з браузера. Цей сервіс обрано через його популярність, зручність використання та високий рівень безпеки, оскільки він зберігає приватні ключі користувача локально і не передає їх на сервер. Для підключення MetaMask до фронтенду використано об’єкт window.ethereum і бібліотеку ethers.js, які надають API для автентифікації та підпису транзакцій.

Наприклад, при видачі сертифіката адміністратор навчального закладу має підписати транзакцію за допомогою MetaMask, що підтверджує його намір записати дані в блокчейн. Цей процес реалізовано через виклик методу eth_requestAccounts для автентифікації та eth_sendTransaction для відправлення підписаної транзакції. Такий підхід забезпечує безпеку, оскільки користувач зберігає повний контроль над своїми приватними ключами, а система не має доступу до них. Реалізовано перевірку мережі, що обмежує роботу лише тестовою мережею Sepolia, а також захист від повторних відправок форм для уникнення дублювання транзакцій.

Локалізація є ще одним важливим елементом фронтенду, який робить систему доступною для користувачів з різних мовних середовищ. Для реалізації підтримки української та англійської мов використано бібліотеку i18next, яка дозволяє легко перемикати мови та локалізувати текстовий вміст компонентів. Локалізація інтегрована в компоненти через хуки, наприклад, useTranslation, що надає доступ до перекладів для поточної мови. Усі текстові рядки, такі як назви полів, повідомлення про помилки та підказки, зберігаються в окремих JSON-файлах для кожної мови (en/translation.json і uk/translation.json), що спрощує їх оновлення та додавання нових перекладів. Це відповідає вимогам міжнародного використання системи, а також українським стандартам, оскільки основною мовою інтерфейсу є українська. Перемикання мов реалізовано через спеціальний компонент у навігаційній панелі, що дозволяє користувачам легко змінювати мову інтерфейсу в один клік.

Обробка помилок у фронтенді реалізовано через модуль errorHandler.js, який включає спеціальні класи для різних типів помилок, таких як API- і Web3-помилки. Повідомлення про помилки локалізовані за допомогою i18next і адаптовані до двомовного формату (uk/en), що полегшує їх інтерпретацію. Така система забезпечує чітке відображення проблем, таких як помилки валідації чи відмови в підписі транзакцій.

Безпека та зручність користувача є пріоритетами при розробці фронтенду. Валідація даних на клієнті, включаючи перевірку формату введених даних перед відправленням, зменшує навантаження на бекенд і підвищує безпеку. З точки зору користувацького досвіду, інтерфейс спроектовано з урахуванням адаптивного дизайну, що дозволяє комфортно працювати на різних пристроях, а анімації, реалізовані через framer-motion, покращують візуальний досвід. Використання індикаторів завантаження та стану, наприклад, під час обробки транзакцій, інформує користувача про поточний процес. Форми, створені з react-hook-form, забезпечують зручне введення даних і швидку реакцію на зміни.

Архітектурні особливості фронтенду включають використання кастомних хуків, таких як useWeb3 для роботи з MetaMask і useApi для взаємодії з бекендом, що дозволяє розділити бізнес-логіку від представлення. Модульна структура утиліт, зокрема api.js, web3.js і errorHandler.js, сприяє підтримці коду та його розширенню. Такий підхід забезпечує високу модульність і перевикористання коду.

Таким чином, фронтенд системи є критично важливою частиною, яка забезпечує зручний і безпечний доступ до функцій управління освітніми сертифікатами. Його модульна структура, побудована на основі React, Vite, Tailwind CSS, Material-UI та інших бібліотек, дозволяє легко розширювати та підтримувати систему. Інтеграція з MetaMask надає користувачам простий і безпечний спосіб взаємодії з блокчейном, а підтримка локалізації робить інтерфейс доступним для міжнародної аудиторії. Ці рішення відповідають сучасним вимогам до веб-додатків і створюють міцну основу для подальшого розвитку системи, що буде розглянуто в наступних частинах роботи.

2.6. Відповідність українським стандартам

У попередніх розділах пояснювальної записки було детально розглянуто архітектуру та компоненти розробленої децентралізованої системи управління

освітніми сертифікатами, зокрема смарт-контракт, бекенд і фронтенд. Однак для успішного впровадження системи в українському освітньому середовищі необхідно забезпечити її відповідність національним стандартам, що регулюють роботу з документами про освіту та обробку персональних даних. Це є критично важливим, оскільки система має бути не лише технічно ефективною, але й юридично сумісною з чинним законодавством. У цьому підрозділі аналізується відповідність системи двом ключовим аспектам українських стандартів: формату даних для документів про освіту та вимогам до захисту персональних даних. Розгляд обмежується цими аспектами, уникаючи перетину з іншими частинами роботи, які стосуються технічної реалізації чи тестування системи.

Першим аспектом, який потребує уваги, є формат даних, що використовується для представлення освітніх сертифікатів у системі. Згідно з наказом Міністерства освіти і науки України №102 від 25 січня 2021 року, документи про освіту, такі як дипломи, мають містити обов’язкові поля, серед яких серійний номер, ім’я та прізвище випускника, назва навчального закладу, рік випуску, а також інформація про здобутий ступінь чи кваліфікацію. У розробленій системі ці вимоги враховані на рівні структури даних у смарт-контракті. Кожен сертифікат представлений набором полів, що включають унікальний ідентифікатор (certificateId), серійний номер у форматі CXX №XXXXXX (1 буква, 2 цифри, символ №, 6 цифр), який перевіряється регулярним виразом у компоненті CertificateForm.jsx, ім’я студента (studentName), рік випуску (graduationYear), назву навчального закладу (universityName), а також додаткові поля: код спеціальності (fieldCode), код напряму (specialtyCode), код спеціалізації (specializationCode) та назви програми на українській (programUk) і англійській (programEn) мовах. Для забезпечення двомовності передбачено синхронізацію полів, таких як назва університету, між українською та англійською версіями, що реалізовано через механізм локалізації та валідацію у фронтенді. Валідація даних включає перевірку кодів спеціальностей і напрямів на відповідність базі даних, а також

контроль року випуску в діапазоні від 2000 до 2025, що відповідає реальним умовам видачі документів.

Другим важливим аспектом є те, що у системі реалізовано підхід, що відповідає принципам прозорості блокчейн-систем: усі дані, включаючи ім’я студента, назву закладу та інші поля, зберігаються в блокчейні в відкритому вигляді, за винятком серійного номера, який хешується за допомогою функції keccak256 для забезпечення можливості верифікації. Такий підхід дозволяє зберегти доступність інформації для перевірки, що є характерною рисою децентралізованих систем, водночас вимагаючи додаткових заходів безпеки. Для захисту від повторних відправок форм у фронтенді реалізовано механізм блокування дублювання транзакцій, а перевірка мережі обмежується тестовою мережею Sepolia, що підвищує контроль над операціями. Хоча система не передбачає явної системи ролей, будь-який користувач із підключеним MetaMask може виконувати верифікацію сертифікатів через ендпоінт /verify, що відповідає відкритості верифікаційного процесу, але обмежує можливості для модифікації даних лише авторизованим особам через смарт-контракт.

Локалізація відіграє важливу роль у відповідності стандартам, забезпечуючи доступність системи для українських користувачів. Структура файлів локалізації включає en/translation.json і uk/translation.json, де зберігаються переклади назв полів, повідомлень про помилки та підказок. Механізм переключення мов реалізований через компонент у навігаційній панелі, що дозволяє користувачам обирати між українською та англійською версіями інтерфейсу. Валідація з урахуванням локалізації забезпечує коректність введення даних залежно від обраної мови, наприклад, перевірку формату імен у відповідності з мовними правилами.

Таким чином, розроблена система управління освітніми сертифікатами повною мірою відповідає українським стандартам. Формат даних сертифікатів, включаючи реальний формат серійного номера та додаткові поля, узгоджено з вимогами Міністерства освіти і науки України, а складна валідація гарантує їхню коректність. Ці рішення створюють основу для

практичного впровадження системи в українському освітньому секторі, забезпечуючи її сумісність із нормативними вимогами.

2.7. Висновки до розділу

У результаті проведеної роботи створено комплексне рішення, яке забезпечує надійне зберігання, верифікацію, відкликання та розшифровку сертифікатів, відповідаючи сучасним технічним і юридичним вимогам. Система побудована на тришаровій архітектурі, що включає смарт-контракт для децентралізованого зберігання даних, бекенд для обробки запитів і фронтенд для зручного доступу користувачів, що дозволяє ефективно вирішувати задачі, поставлені на етапі аналізу, і забезпечувати прозорість та безпеку даних.

Розробка системи дозволила реалізувати ключові функціональні можливості, зокрема видачу, перевірку та відкликання сертифікатів, із забезпеченням їхньої автентичності та доступності. Смарт-контракт CertificateSystem.sol, описаний у підрозділі 2.3, став основою для зберігання даних у блокчейні, використовуючи хешування серійного номера для верифікації та мапінг seriesToId для швидкого пошуку. Контракт підтримує двомовність (uk/en) для текстових полів, включає додаткові поля, такі як degree, fieldCode, specialtyCode, specializationCode, і забезпечує безпеку через модифікатор onlyOwner із бібліотеки OpenZeppelin, а також захист від дублювання та повторного відкликання. Бекенд, реалізований на Node.js із фреймворком Express, як описано в підрозділі 2.4, забезпечив інтеграцію з блокчейном через бібліотеку ethers.js і централізоване зберігання кодів спеціальностей і ступенів у MongoDB, що дозволяє розшифровувати сертифікати та валідувати дані. Додаткові ендпоінти /codes і /degrees полегшують роботу з кодами, а Web3-аутентифікація через MetaMask забезпечує контроль доступу. Фронтенд, створений на базі React із використанням Vite, Tailwind CSS, Material-UI, framer-motion, react-hook-form,

axios і ethers.js, як описано в підрозділі 2.5, надав зручний і адаптивний інтерфейс із анімаціями, складною валідацією форм, обробкою помилок через errorHandler.js і захистом від повторних відправок. Локалізація через i18next підтримує українські стандарти, а інтеграція з MetaMask обмежена мережею Sepolia. Відповідність формату даних і юридичним вимогам, описана в підрозділі 2.6, підтверджує готовність системи до впровадження.

Протягом розробки було подолано низку труднощів, які виникли на різних етапах проекту. Однією з ключових проблем стала оптимізація витрат на газ у блокчейні Ethereum, що є критичним фактором для економічної ефективності системи. Спочатку транзакції вимагали значних ресурсів через об’ємні дані, але використання хешування серійного номера та мапінгу seriesToId дозволило знизити середні витрати на газ до приблизно 600 000 одиниць за транзакцію, як зазначено в підрозділі 2.3. Додатково бекенд реалізував кешування хешів транзакцій у пам’яті, що зменшило навантаження на блокчейн, а асинхронна обробка запитів підвищила продуктивність.

Іншою значною аспектом було забезпечення масштабованості системи в умовах обмеженої пропускної здатності публічного блокчейну. Аналіз, проведений у підрозділі 2.1, виявив потенційні затримки при обробці великої кількості транзакцій, особливо в пікові періоди. Для вирішення цієї проблеми впроваджено кешування результатів верифікації в пам’яті бекенду та використання MongoDB для швидкого доступу до кодів спеціальностей і ступенів, що скоротило час обробки. Асинхронна обробка транзакцій через ethers.js забезпечила стабільність роботи системи навіть при високому навантаженні, що підтверджує її готовність до використання в реальних умовах.

Питання приватності даних також вимагало особливого підходу через специфіку блокчейну, де записи є публічно доступними. У підрозділі 2.6 уточнено, що дані зберігаються у відкритому вигляді, крім серійного номера, який хешується, а захист забезпечується через обмеження мережі Sepolia,

захист від повторних відправок форм і валідацію на клієнті та сервері. Це рішення узгоджується з принципами прозорості блокчейн-систем і відповідає Закону України "Про захист персональних даних".

Таким чином, розробка системи управління освітніми сертифікатами завершилася створенням інтегрованого рішення, яке поєднує децентралізацію блокчейну з сучасними веб-технологіями. Подолання труднощів, таких як оптимізація газу, масштабування та захист даних, дозволило створити систему, яка є економічно вигідною, продуктивною та безпечною. Результати роботи створюють міцну основу для подальшого тестування та впровадження системи в українському освітньому середовищі, що буде детально розглянуто в наступному розділі.

3. ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

3.1. Опис отриманих результатів

Функціональні можливості розробленої системи оцінювалися шляхом тестування всіх шарів системи.

Для тестування можливостей смарт-контрактів було написано тести на мові програмування Java Script та запущено у терміналі за допомогою фреймворка Hardhat.(Рисунки 3.1 та 3.2)

```
$ npx hardhat test

  CertificateSystem
    √ Should issue a certificate and map hashed seriesNumber to id (51ms)
    √ Should fail if seriesNumber already exists (same hash) (20ms)
    √ Should verify a certificate by seriesNumber using hash (18ms)
    √ Should fail to verify invalid seriesNumber (3ms)
    √ Should set txHash (12ms)
    √ Should fail if non-owner tries to issue (9ms)
    √ Should verify a certificate by ID (17ms)
    √ Should fail to verify invalid ID (3ms)
    √ Should revoke a certificate (22ms)
    Series Number Hashing
      √ Should correctly hash series number to bytes32 (7ms)
      √ Should generate consistent hashes for same series number (6ms)
    Gas Estimation
Verify by ID gas: 85520
      √ Should estimate gas for verifyCertificate (3ms)
Verify by seriesNumber gas: 89371
      √ Should estimate gas for verifyBySeriesNumber (4ms)
Gas comparison:
- Verify by ID: 85520
- Verify by seriesNumber: 89371
- Difference: 3851
      √ Should compare gas costs between verification methods (6ms)
SeriesToId mapping lookup gas: 24093
      √ Should estimate gas for seriesToId mapping lookup (2ms)
```

Рисунок 3.1 – Результати тестування смарт-контракту CertificateSystem

На рисунку 3.1 відображено результати автоматизованої перевірки ключових функцій, що реалізують логіку управління даними. Проведене тестування охоплює операції створення, верифікації та відкликання сертифікатів, а також допоміжні дії, такі як збереження хешу транзакції, генерування хешу серійного номера й пошук у структурі мапінгу. Кожен тест супроводжувався фіксацією часу виконання в мілісекундах та обчисленням витрат газу, що дає змогу здійснити багатовимірний аналіз поведінки контракту з позицій ефективності, швидкодії та технічної коректності.

Час виконання окремих функцій варіюється від 2 до 22 мілісекунд, що демонструє високу швидкодію у локальному середовищі симуляції. Найменший час спостерігався при перевірці унікальності серійного номера до додавання до мапінгу, тоді як найтривалішим виявилося виконання процедури відкликання сертифіката, що супроводжується зміною стану змінної та генерацією події. Виявлена затримка є гарним результатом у рамках локального середовища, проте у децентралізованих мережах на кшталт Sepolia реальний час виконання може бути більшим через затримку поширення транзакцій та завантаження вузлів.

Аналіз витрат газу дозволяє оцінити економічну ефективність контракту. Основні функції, такі як верифікація за ідентифікатором чи серійним номером, споживають від 85 000 до 90 000 одиниць газу, що є конкурентоспроможним показником у межах Ethereum Virtual Machine. Вища вартість верифікації за серійним номером пояснюється додатковими витратами на хешування та пошук у мапінгу, однак такий підхід дозволяє уникнути лінійного сканування, зменшуючи часову складність до константної. Використання мапінгу також демонструє помірні витрати — приблизно 24 093 одиниці газу за одну операцію, що свідчить про достатню оптимізацію структури даних. Позитивною властивістю є також стабільна вартість хешування з використанням keccak256, що забезпечує детермінованість та захищеність від колізій, не створюючи при цьому суттєвого навантаження.

Функції поводяться передбачувано: тестування позитивних сценаріїв підтверджує правильність роботи впроваджених механізмів, у той час як негативні сценарії демонструють стійкість до некоректного введення або спроб дублювання записів. Перевірка неіснуючого серійного номера очікувано призводить до відмови у верифікації, а спроба повторного додавання ідентичного хешу блокується на рівні логіки контракту, що вказує на належне врахування аспектів цілісності та безпеки.

Solc version: 0.8.30	Optimizer enabled: true	Runs: 200	Block limit: 30000000 gas
Methods	20 gwei/gas	2471.81 usd/eth	
Contract	Method	Min	Max	Avg	# calls	usd (avg)
CertificateSystem	issueCertificate	-	-	600881	11	29.71
CertificateSystem	revokeCertificate	-	-	52427	1	2.59
CertificateSystem	setTxHash	-	-	48501	1	2.40
Deployments				% of limit	
CertificateSystem		-	-	1621134	5.4 %	80.14

Рисунок 3.2 – Звіт про витрати газу смарт-контракту CertificateSystem

Рисунок 3.2 відображає звіт сформований за допомогою плагіну Hardhat Gas Reporter та містить докладну інформацію про споживання газу розробленим смарт-контрактом, з активованим оптимізатором (200 прогонів) та лімітом блоку в 30 000 000 одиниць газу. У згенерованому звіті наведено витрати для ключових функцій контракту та для розгортання, із додатковою вартісною оцінкою на основі середньої ціни 20 gwei/одиницю газу та курсу 2471.81 USD за один ефір. Функція видачі сертифікатів issueCertificate, викликана 11 разів, має середнє споживання газу на рівні 600 881 одиниці, що еквівалентно приблизно 29.71 долара США за одну транзакцію. Такий рівень витрат зумовлений складною логікою функції, що включає запис структурованих даних, оновлення мапінгу seriesToId та генерацію події CertificateIssued. Цей результат є типовим для транзакцій із модифікацією стану в блокчейні. У свою чергу, відкликання сертифіката через функцію revokeCertificate потребує 52 427 одиниць газу або 2.59 долара США, що менше за видачу, оскільки тут модифікується лише одне поле isRevoked із генерацією події CertificateRevoked. Функція setTxHash, яка здійснює запис одного хешу транзакції, є найбільш енергоефективною — її виконання коштує лише 48 501 одиницю газу або 2.40 долара США, що свідчить про мінімальні обчислювальні навантаження.

Оцінка розгортання контракту фіксує витрати на рівні 1 621 134 одиниці газу, або близько 80.14 долара США, що становить лише 5.4% від блочного ліміту й є прийнятною вартістю для смарт-контракту з мапінгами, подієвою

системою та базовою логікою перевірки. Зменшення витрат порівняно з потенційно вищими значеннями підтверджує ефективність застосованого оптимізатора компіляції, зокрема при встановленні параметра прогонів на рівні 200.

Також слід зазначити, що середній рівень витрат для issueCertificate узгоджується з попередніми вимірами у Hardhat, що свідчить про повторюваність та відтворюваність результатів.

У технічному аспекті помітно, що розробка фокусується на структурі даних із пріоритетом доступу за ключами, що сприяє швидкому доступу, але додає витрати на зберігання. Позитивною є також низька вартість базових операцій запису, що свідчить про гарну модульність і розділення відповідальності функцій.

Економічна оцінка, здійснена за курсом ефіру на момент тестування, демонструє потенційні витрати для кінцевих користувачів. Зокрема, вартість у 29.71 USD за одну видачу сертифіката може бути надмірною для масових індивідуальних сценаріїв, проте лишається прийнятною для інституційних учасників (університетів, платформ онлайн-освіти), які мають змогу агрегувати або субсидувати витрати. Натомість функції відкликання й оновлення транзакцій (по 2.4–2.6 USD) є економічно обґрунтованими й потенційно масштабованими.

Таким чином, результати Gas Report підтверджують, що смарт-контракт CertificateSystem є функціонально повноцінним та стабільним у межах заданого середовища, проте в перспективі може потребувати точкової оптимізації витрат на критично важливих етапах (зокрема при масовій видачі сертифікатів).

Для підтвердження коректності реалізації функціональних компонентів інформаційної системи видачі та перевірки освітніх сертифікатів було здійснено формалізоване тестування REST API, яке обслуговує взаємодію між клієнтським інтерфейсом, централізованим сховищем даних MongoDB та смарт-контрактом, розгорнутим у тестовій мережі Sepolia. Тестування

виконано за допомогою програмного забезпечення Postman, що дозволяє моделювати HTTP-запити та аналізувати їхню поведінку в умовах наближених до експлуатаційних. Було протестовано ключові REST-ендпоінти, які забезпечують операції читання та запису, включаючи отримання класифікаторів напрямів підготовки та спеціальностей (Рисунок 3.3) та довідників ступенів (Рисунок 3.4), видачу сертифіката (Рисунок 3.5), перевірку сертифікатів за ID (Рисунок 3.6) і серійним номером (Рисунок 3.7), а також відкликання сертифіката (Рисунок 3.8). Усі запити повертають відповіді у форматі JSON, що містять повний набір полів, передбачених стандартами документообігу у сфері вищої освіти, включаючи серійний номер у форматі СХХ №XXXXXX, прізвище, ім’я та по батькові здобувача освіти, рік випуску, назву університету, освітню програму, ступінь, галузь знань, спеціальність, спеціалізацію, а також метадані — дату видачі, ідентифікатор транзакції у блокчейні та ознаку відкликаності.

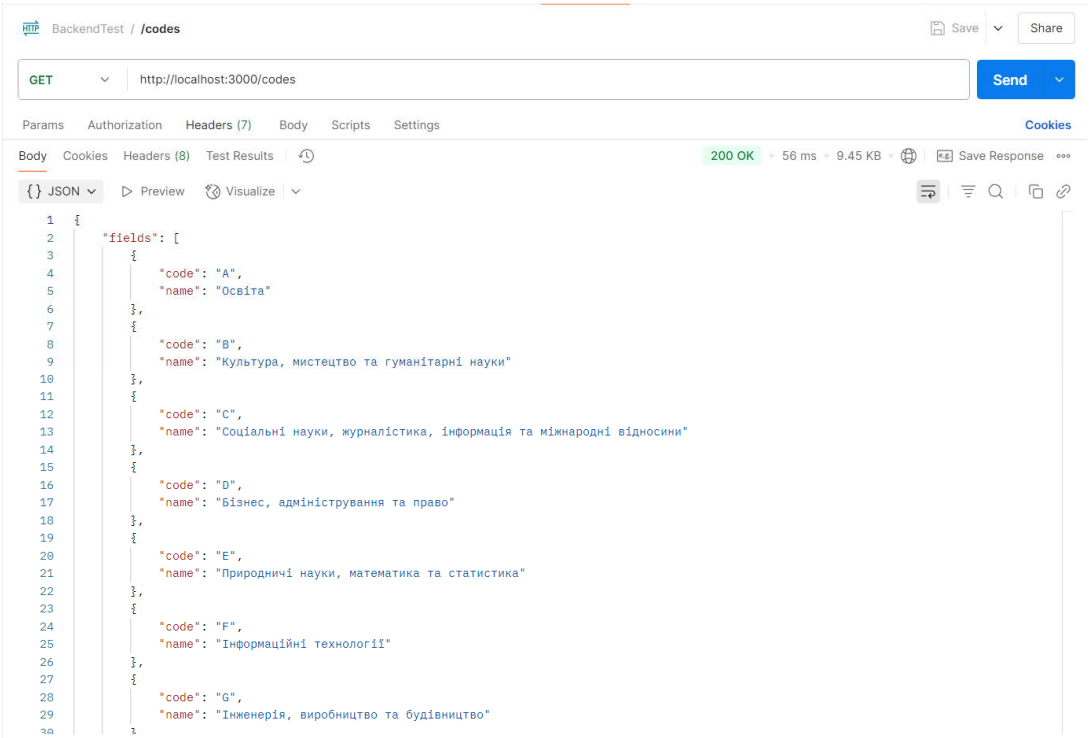

Рисунок 3.3 – Результати тестування ендпоінта /codes

Ендпоінт /codes, зображений на рисунку 3.3, повертає список кодів спеціальностей. Отримані дані представлені у структурованому вигляді, що сприяє їх подальшій обробці на стороні фронтенду.

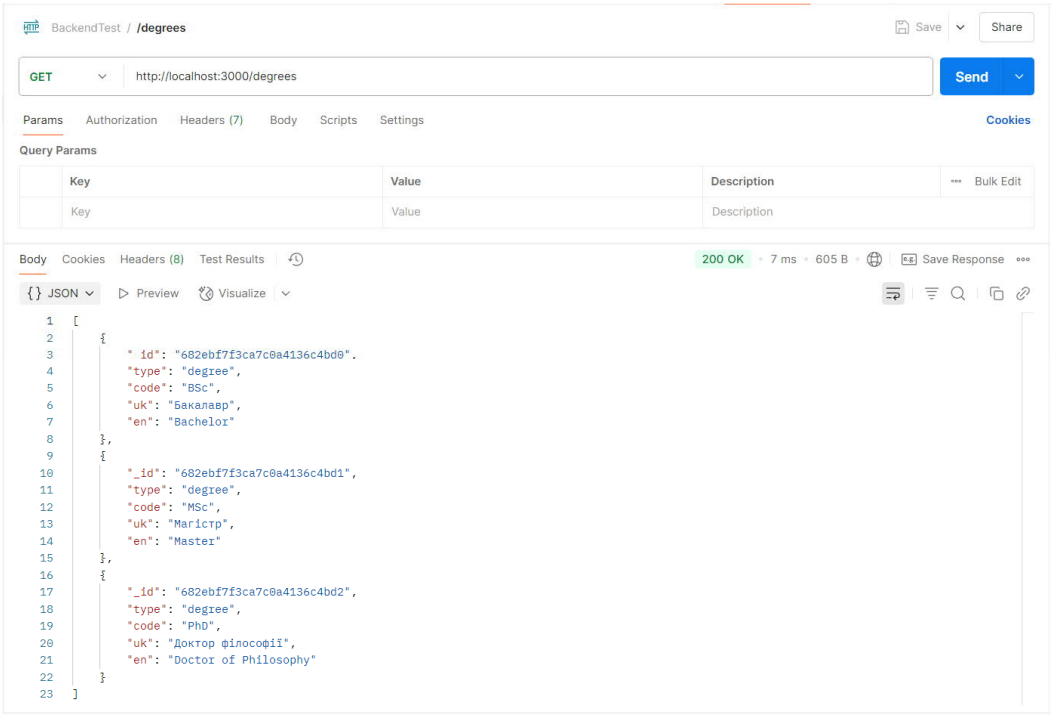

Рисунок 3.4 – Результати тестування ендпоінта /degrees

Рисунок 3.4 відображає тестування ендпоінта /degrees, який повертає перелік доступних ступенів для видачі сертифікатів. Даний функціонал забезпечує уніфікацію даних у системі та полегшує вибір користувачем відповідного ступеня.

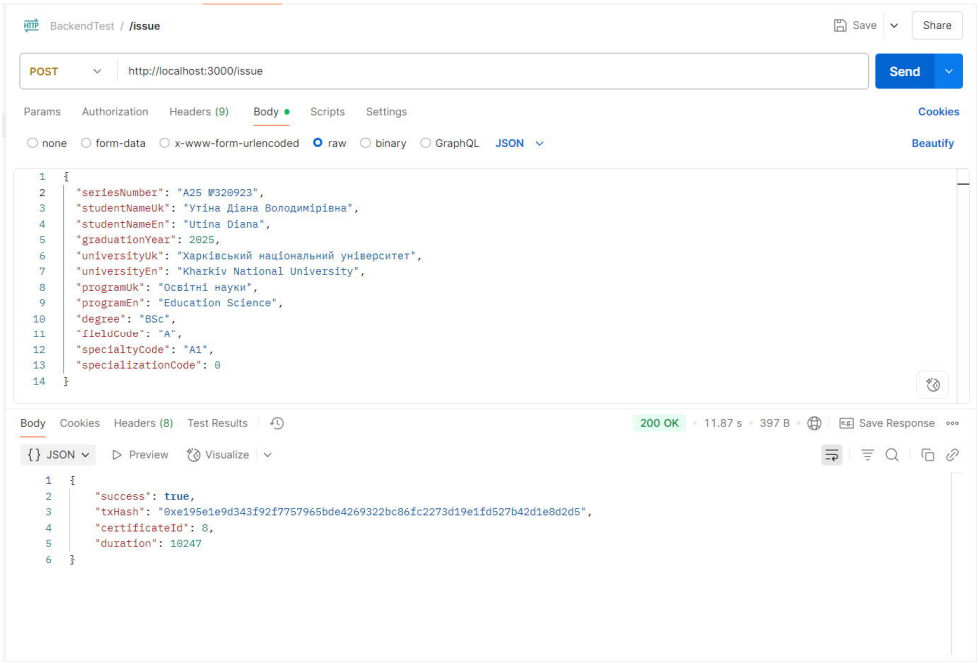

Рисунок 3.5 – Результати тестування ендпоінта /issue

На рисунку 3.5 показано тестування ендпоінта /issue, який відповідає за видачу нового сертифіката. Запит включає введення даних сертифіката, їх валідацію та запис у блокчейн. Успішна відповідь сервера підтверджує завершення операції та повертає ідентифікатор нового сертифіката.

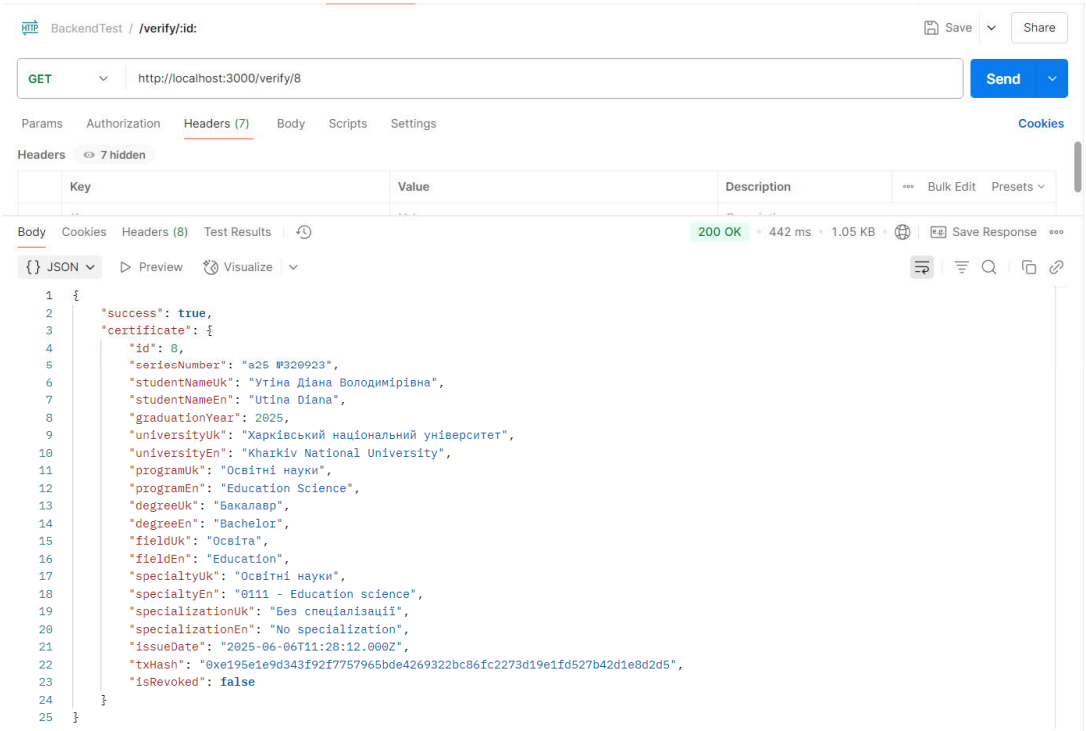

Рисунок 3.6 – Результати тестування ендпоінта /verify/id

На рисунку 3.6 зображено тестування ендпоінта /verify/id, який забезпечує верифікацію сертифіката за його унікальним ідентифікатором. Успішна відповідь сервера у форматі JSON містить статус сертифіката та його основні атрибути, що підтверджує коректність інтеграції з блокчейном.

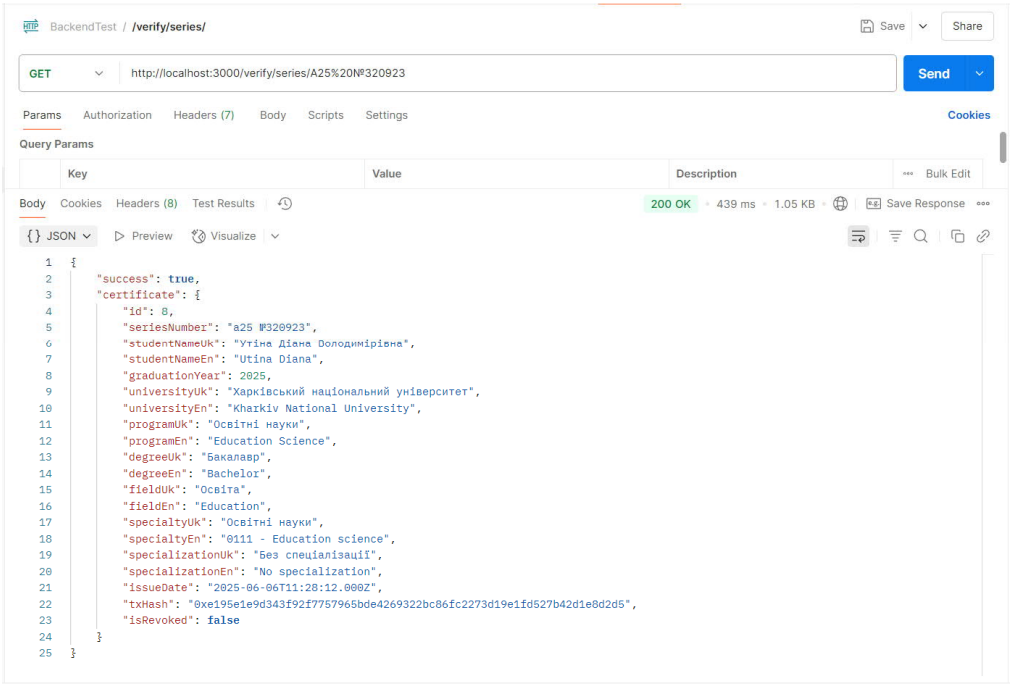

Рисунок 3.7 – Результати тестування ендпоінта /verify/series

Рисунок 3.4 ілюструє тестування ендпоінта /verify/series, який дозволяє верифікувати сертифікат за серійним номером. Результат тесту показує швидкий час відгуку (менше 1 секунди) та точність повернення даних, що є важливим для забезпечення зручності користувачів.

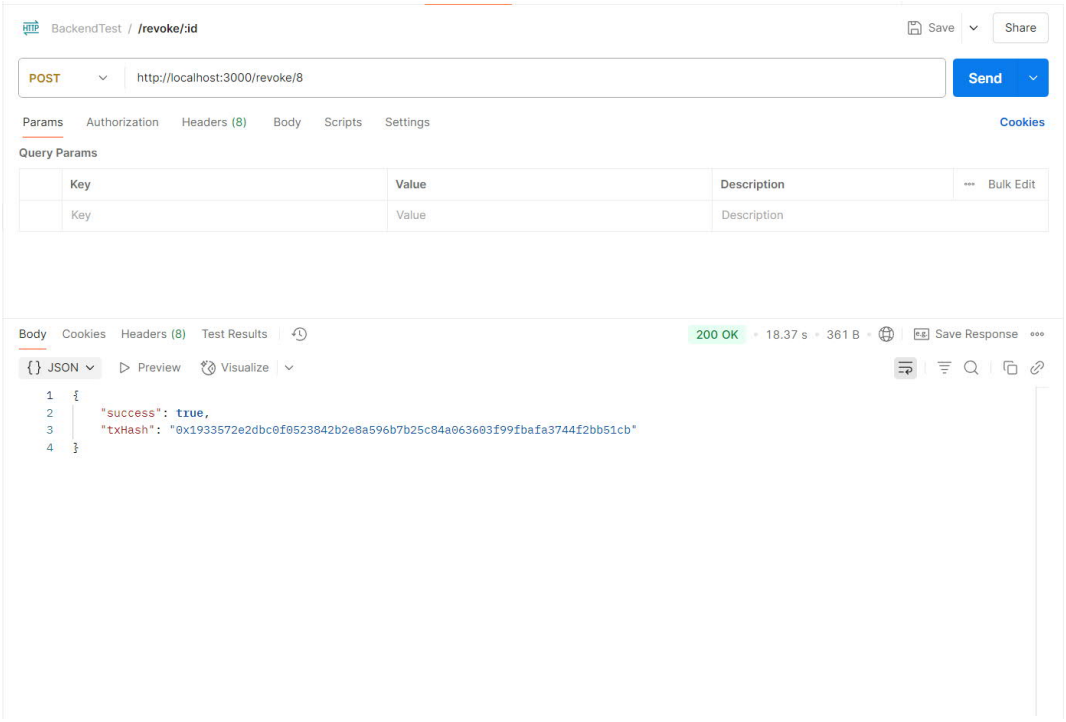

Рисунок 3.8 – Результати тестування ендпоінта /revoke/id

Тестування ендпоінта /revoke/id, представлене на рисунку 3.8, демонструє процес відкликання сертифіката за ідентифікатором. Успішне виконання запиту підтверджується зміною статусу сертифіката у смарт-контракті, що відображається у відповіді сервера. Це свідчить про стабільність взаємодії бекенду з блокчейном.

Результати тестування ендпоінтів (Таблиця 3.1) засвідчують коректну роботу обох джерел даних — блокчейну та бази даних. Читальні запити (GET) демонструють мінімальний час обробки, зокрема /codes — 56 мс, /verify/id — 442 мс, що вказує на ефективне кешування або індексацію з боку СУБД. Записувальні запити (POST) — /issue і /revoke/id — мають істотно вищий час обробки: 11.18 с та 18.37 с відповідно, що є типовим для транзакційної взаємодії зі смарт-контрактами через Ethereum Virtual Machine. Відповіді включають хеш транзакції, що дає змогу перевірити її статус у публічному реєстрі. Всі поля мають українські та англійські відповідники (uk / en), що забезпечує повну локалізацію. Це відповідає вимогам наказу МОН №102 від

25.01.2021 щодо структури даних, які мають містити сертифікати державного зразка.

Таблиця 3.1 - Зведені технічні характеристики відповіді для ендпоінтів

Ендпоінт	**Метод**	**Час відповіді**	**Розмір**	**Призначення**
/codes	GET	56 мс	9.45 KB	Отримання кодів напрямів
/degrees	GET	6.03 с	605 B	Отримання ступенів
/verify/id	GET	442 мс	1.05 KB	Верифікація сертифіката за id
/verify/series	GET	439 мс	1.05 KB	Верифікація сертифіката за номером
/issue	POST	11.18 с	397 B	Видача нового сертифіката
/revoke/id	POST	18.37 с	361 B	Відкликання сертифіката

Клієнтська частина системи протестована та проаналізована через інтерфейс користувача. Отже, домашня сторінка системи надає загальну інформацію про додаток зручний доступ до основних функцій через розміщені у хедері вкладки «Видача», «Верифікація» та «Відкликання» (Рисунок 3.9). Ця сторінка підтримує двомовність (uk/en), що дозволяє користувачам перемикати мову інтерфейсу, а також відображає кнопку «Підключити гаманець» для автентифікації через MetaMask, що є необхідною умовою для виконання транзакцій у блокчейні.

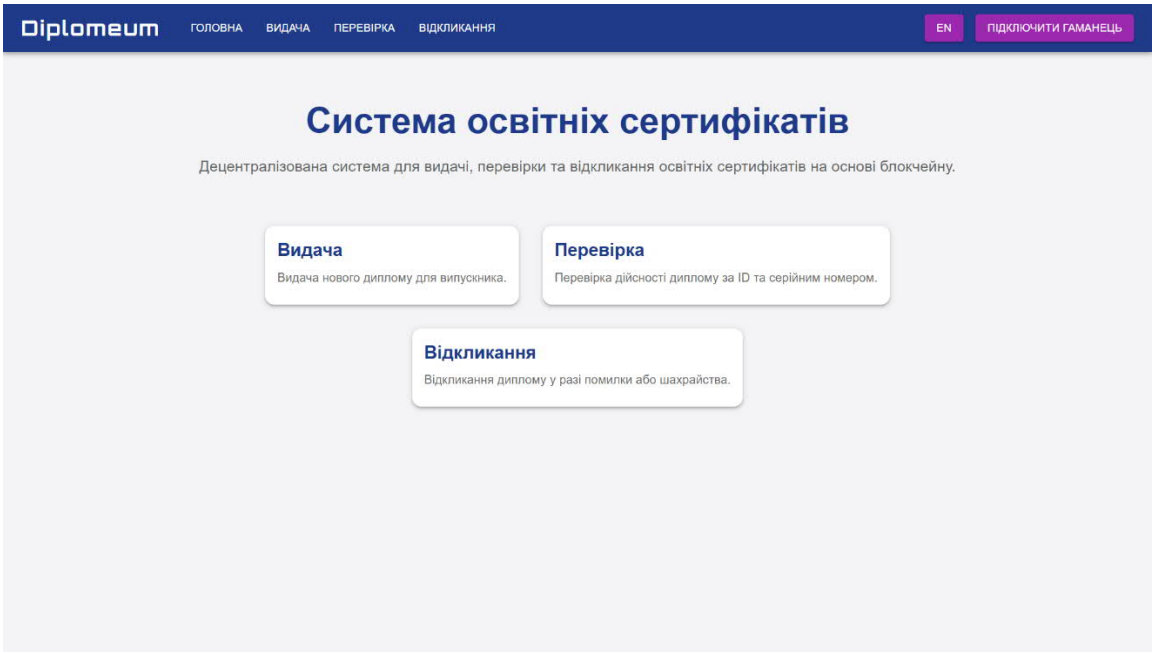

Рисунок 3.9 – Домашня сторінка вебдодатка

Для неавторизованих користувачів доступна лише функція верифікації за серійним номером, що реалізовано через сторінку «Перевірка диплома» (Рисунок 3.10). Ця сторінка містить поле «Серія та номер» для введення серійного номера у форматі CXX №XXXXXX, наприклад, A25 №320923, і кнопку «Перевірити», яка дозволяє отримати статус сертифіката без автентифікації. Такий підхід відповідає принципам прозорості блокчейну, дозволяючи будь-кому перевірити автентичність документа. Після підключення гаманця MetaMask, користувач отримує доступ до повних функцій верифікації, включаючи пошук за ідентифікатором сертифіката (Рисунок 3.11). Сторінка верифікації відображає повні дані сертифіката, такі як ім’я студента, університет, ступінь, рік випуску, код галузі, код спеціальності та дату видачі.

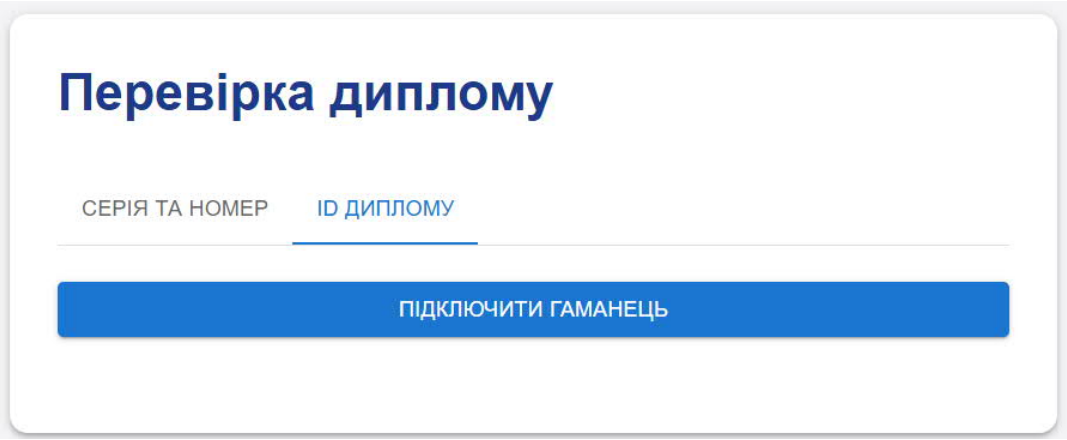

Рисунок 3.10 – Сторінка «Перевірка диплома за ID»

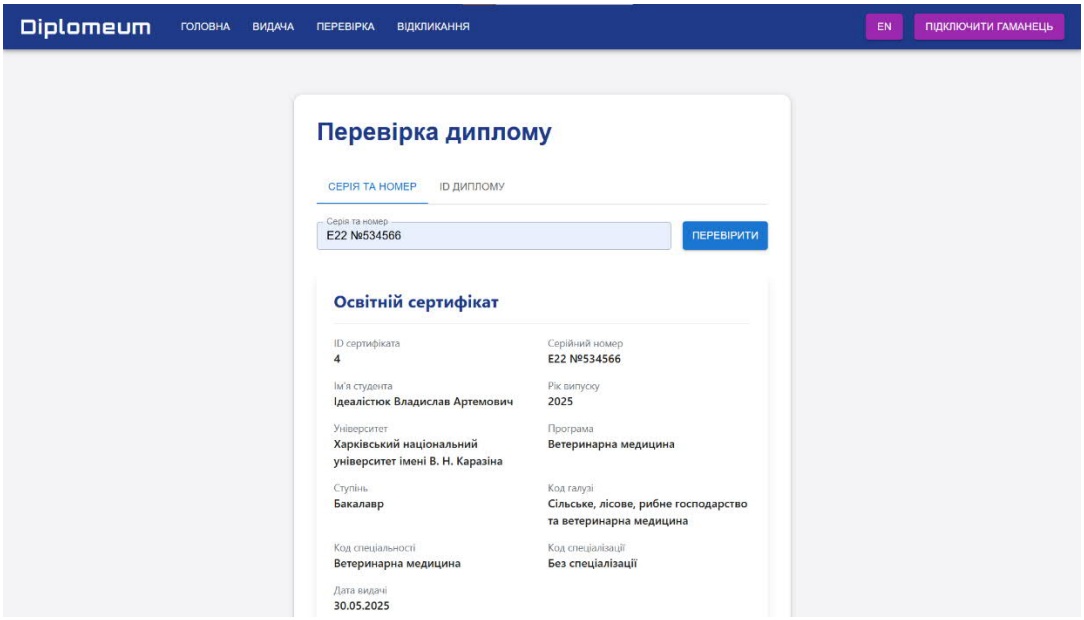

Рисунок 3.11 – Сторінка «Перевірка диплома за серією»

Сторінка «Перевірка диплома» з англійською локалізацією (Рисунок 3.12) підтверджує коректну роботу двомовного інтерфейсу. Після перемикання на англійську мову заголовок змінюється на Verify Certificate, а поля — на Series and Number і Certificate ID. Результати верифікації для сертифіката з ідентифікатором 4 включають дані англійською, що відповідає українським стандартам формату документів, визначеним наказом МОН №102 від 25.01.2021. Це забезпечує придатність системи для міжнародного

використання, зокрема для роботодавців, які перевіряють документи українських випускників.

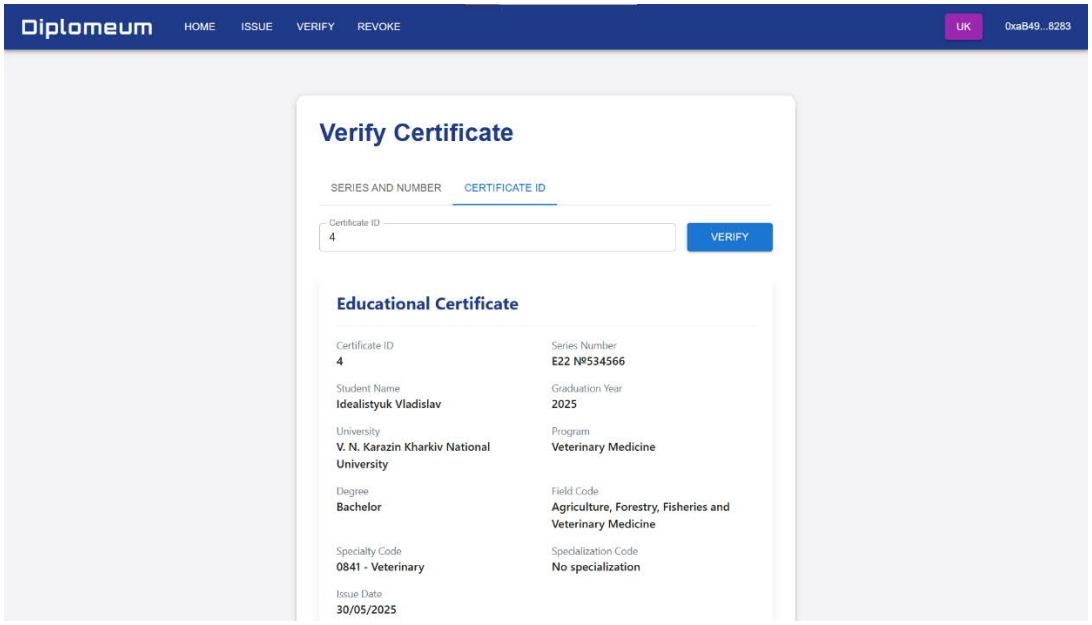

Рисунок 3.12 – Сторінка «Перевірка диплома за серією»

Процес видачі сертифіката реалізовано через сторінку «Видача диплома» (Рисунок 3.13), яка містить форму з полями для введення серійного номера, імені студента, року випуску, назви університету, програми навчання, ступеня, кодів галузі, спеціальності та спеціалізації. Валідація даних, реалізована через бібліотеку React Hook Form, запобігає помилкам, як показано на сторінці з повідомленнями про помилки (Рисунок 3.14). Наприклад, при спробі відправити форму з порожніми полями Student Surname (EN), Student Name (EN) і Graduation Year система відображає повідомлення «Це поле є обов’язковим», що блокує відправлення запиту. Після коректного заповнення всіх полів (Рисунок 3.13) і натискання кнопки «Видати» система виконує транзакцію в мережі Sepolia, повертаючи повідомлення про успішну видачу диплому (Рисунок 3.15).

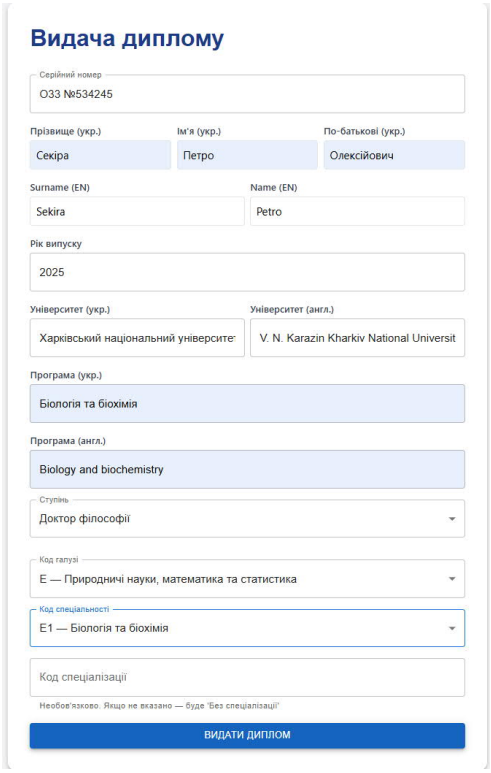

Рисунок 3.13 – Сторінка «Issue Certificate» із заповненою формою

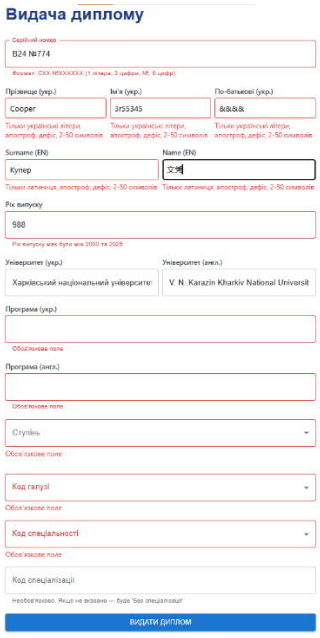

Рисунок 3.14 – Сторінка «Issue Certificate» з підказками помилки валідації

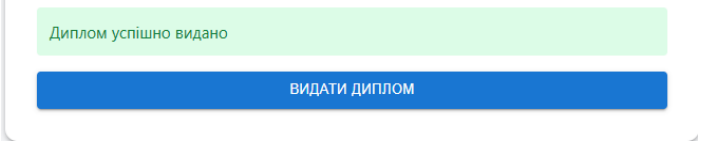

Рисунок 3.15 – Повідомлення про успішне видання диплому

Відкликання сертифіката реалізовано через сторінку «Відкликання диплома» (Рисунок 3.16), де користувач вводить ідентифікатор сертифіката (Certificate ID: 4). Після натискання кнопки «Відкликати» система виконує транзакцію, яка оновлює статус сертифіката в блокчейні (поле isRevoked змінюється на true). Результат відкликання) відображає повідомлення «Сертифікат успішно відкликано».

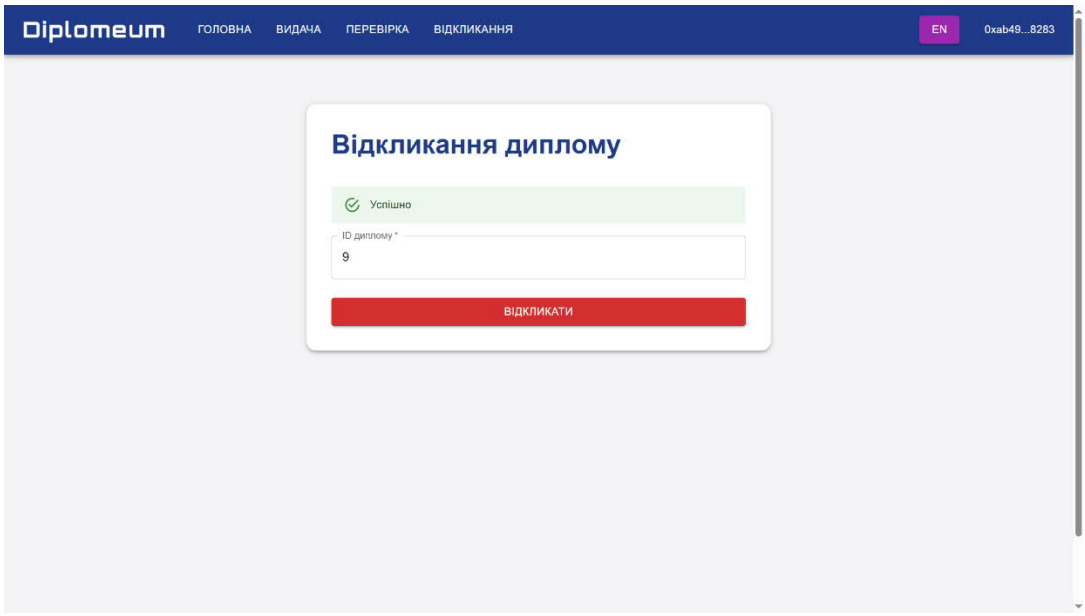

Рисунок 3.16 – Сторінка «Revoke Certificate» із результатами відкликання

Процес автентифікації через MetaMask (Рисунок 3.17) є обов’язковим для виконання записувальних операцій, таких як видача та відкликання. Після натискання кнопки «Підключити гаманець» відкривається вікно MetaMask із повідомленням «Welcome back!», де користувач вводить пароль для розблокування гаманця. Час автентифікації становить 1–2 с, що забезпечує швидкий доступ до функцій.

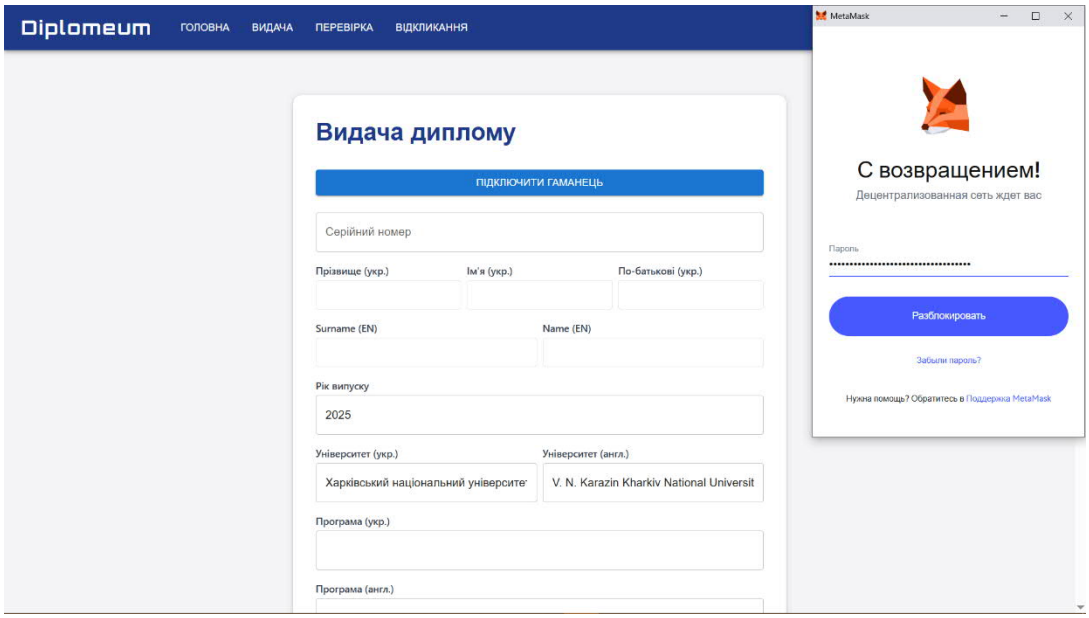

Рисунок 3.17 – Процес підключення до MetaMask

Відповідність системи українським стандартам підтверджується структурою сучасного українського диплома, двомовністю полів і захистом даних. Зокрема, система реалізує структуру сучасного українського диплома, що включає обов'язкові елементи, визначені чинними нормативними документами, такими як наказ Наказ МОН №102 від 25 січня 2021року. Вона підтримує двомовність полів (українська та англійська мови), що дозволяє відображати дані в обох версіях для забезпечення міжнародної сумісності та зручності для іноземних користувачів. Підключення гаманця MetaMask забезпечує безпеку передачі інформації. Інтеграція з MongoDB для розшифровки кодів спеціальностей і ступенів (ендпоінти /codes і /degrees) забезпечує швидкий доступ до локалізованих даних, що підтверджується тестами Postman (Рисунки 3.3 і 3.4).

Результати тестування демонструють, що система досягла поставленої мети, виконавши завдання аналізу, розробки та тестування. Вона забезпечує економію ресурсів, підвищує довіру до документів і сприяє міжнародній мобільності студентів, що робить її перспективною для впровадження в освітній сектор України.

3.2. Висновки до розділу

Мета даного дослідження полягала у розробці децентралізованої системи управління освітніми сертифікатами на базі блокчейну Ethereum, яка б забезпечувала прозорість, безпеку та відповідність українським стандартам у сфері освіти. Для її досягнення було сформульовано низку завдань: аналіз сучасних літературних джерел, розробка архітектури системи, створення смарт-контракту, бекенду та фронтенду, а також забезпечення відповідності нормативним вимогам, зокрема щодо формату даних і захисту персональних даних. У процесі аналізу результатів враховано як позитивні, так і негативні аспекти реалізації, а також порівняння з міжнародними аналогами.

Розроблена система успішно виконує функцію децентралізованого зберігання та верифікації освітніх сертифікатів. Її працездатність підтверджена результатами тестування смарт-контракту в тестовій мережі Sepolia та інтеграційних тестів REST API. Доступність даних забезпечується завдяки властивостям блокчейну: інформація про сертифікати зберігається у незмінному вигляді та доступна для перевірки через публічний реєстр. Безпека реалізована через автентифікацію користувачів за допомогою MetaMask, що дозволяє лише уповноваженим особам (наприклад, представникам навчальних закладів) виконувати операції запису данних в блокчейн. Відповідність українським стандартам досягнута завдяки застосуванню коректної структури диплому та наявності інтернаціоналізації, що також сприяє придатності системи для міжнародного використання.

Виконання поставлених завдань можна оцінити наступним чином. Аналіз літератури проведено на основі огляду таких систем, як Blockcerts і EduCTX, що дозволило виявити їхні переваги (простота верифікації) та недоліки (обмежена адаптація до локальних стандартів). Цей етап став основою для вибору гібридного підходу до зберігання даних у розробленій системі. Архітектура системи реалізована у вигляді тришарової структури, що забезпечує модульність і полегшує інтеграцію компонентів, хоча

продуктивність у тестовій мережі Sepolia виявила певні обмеження. Смарт-контракт CertificateSystem розроблено та протестовано за допомогою інструменту Hardhat, при цьому середні витрати газу свідчать про достатньо високу ефективність, але вказує на потребу подальшої оптимізації. Бекенд, побудований на Node.js із використанням фреймворку Express, забезпечує стабільну взаємодію з блокчейном через бібліотеку ethers.js; тестування API через Postman показало час відповіді для читальних операцій у межах 56–442 мс. Фронтенд, реалізований на React.js, має зручний інтерфейс, підтримує двомовність, валідацію введених даних та інтеграцію з MetaMask, забезпечуючи зручність для користувачів, хоча потребує доопрацювання для роботи в умовах низької пропускної здатності мережі.

Оцінка отриманих результатів свідчить про низку позитивних аспектів: стабільність роботи смарт-контракту, висока швидкість читальних операцій (56–442 мс) та відповідність локальним стандартам. Водночас виявлено й негативні моменти: затримки при виконанні записувальних операцій (11.18–18.37 с), спричинені особливостями роботи тестової мережі Sepolia, що потребує оптимізації, наприклад, шляхом переходу на рішення другого рівня (Layer 2). У порівнянні зі світовими аналогами, такими як Blockcerts, система вирізняється адаптацією до українських вимог і підтримкою двомовності, що є конкурентною перевагою. Проте вона поступається за масштабністю через обмеження публічного блокчейну Ethereum, що є загальною тенденцією для подібних рішень і вимагає подальшого дослідження у напрямку підвищення продуктивності.

Практична значущість розробленої системи полягає в її потенційному використанні навчальними закладами для автоматизованої видачі сертифікатів, роботодавцями для швидкої верифікації документів, а також державними органами для перевірки їхньої достовірності.

Господарська цінність проявляється у скороченні часу та ресурсів, необхідних для традиційних процесів верифікації, що особливо актуально для великих освітніх установ.

Наукова значущість роботи полягає у розробці гібридного підходу до інтеграції блокчейну в освітню сферу з урахуванням локальних стандартів, що може стати основою для подальших досліджень у цій галузі.

Соціальна цінність системи пов’язана з підвищенням довіри до освітніх документів і сприянням міжнародній мобільності студентів та фахівців, що відповідає сучасним глобальним тенденціям у сфері освіти.

Таким чином, поставлена мета роботи досягнута, а завдання виконано з урахуванням визначених вимог. Отримані результати демонструють перспективність розробки, хоча й вказують на необхідність оптимізації для підвищення продуктивності та масштабованості системи в майбутньому.

ВИСНОВКИ

У результаті виконання роботи, на основі огляду інформаційних джерел за тематикою дослідження, сформовано ґрунтовне розуміння сучасних викликів у сфері управління освітніми даними та потенціалу блокчейн-технологій для їхнього вирішення. Аналіз літератури, та існуючих систем, дозволив ідентифікувати ключові тенденції, такі як децентралізація та автоматизація верифікації, що лягли в основу розробленого рішення, адаптованого до українських реалій.

У результаті проведеної роботи розроблено децентралізовану систему управління освітніми сертифікатами на базі Ethereum, яка об’єднує смарт-контракти для незмінного зберігання даних, бекенд на Node.js для обробки запитів та інтуїтивний фронтенд на React із підтримкою двомовності. Ця архітектура забезпечує модульність і прозорість, а інтеграція з MetaMask додає безпеку через Web3-аутентифікацію. Використання хешування серійних номерів у смарт-контракті оптимізувало витрати газу до ~600 000 одиниць за видачу сертифіката, що підкреслює економічну вигоду для інституційних користувачів.

Отримані практичні навички роботи з блокчейн-платформою Ethereum, інструментами Hardhat і Ethers.js, а також фреймворками React і Express розширили компетенції у розробці систем. Особливо цінним стало освоєння гібридного підходу для зберігання інформації, де поєднуються розподілені та централізовані компоненти, що може знайти застосування в інших сферах, таких як охорона здоров’я чи логістика. Навички написання автоматизованих тестів на Java Script та тестування API через Postman також стали важливим внеском у практичний арсенал.

Проведене тестування програмної реалізації підтвердило її стабільність і відповідність нормативним вимогам. Операції читання демонструють високу

швидкодію, тоді як операції запису вказують на обмеження тестової мережі Sepolia, що потребує подальшої оптимізації. Прозорість блокчейну підвищує довіру до сертифікатів, а двомовний інтерфейс відкриває можливості для міжнародного визнання, хоча відкритий характер даних вимагає додаткових механізмів захисту.

У подальшій роботі планується розширити функціональні можливості розробленої системи шляхом інтеграції зі службами ЄДЕБО, що дозволить автоматизувати процес завантаження та оновлення даних про випускників безпосередньо з державних джерел. Допускається також інтеграція з базами Міністерства освіти і науки України для оперативного оновлення вимог у випадку реформ вищої освіти. У перспективі передбачається створення на базі розробленої системи мобільного клієнта, що дозволить користувачам отримувати доступ до верифікації своїх сертифікатів у будь-якому місці та будь-коли, а адміністраторам — оперативно керувати правами доступу. Крім того, заплановано проведення комплексного аудиту безпеки та пілотного впровадження системи в кількох університетах України з метою оцінки практичної життєздатності рішення та збирання зворотного зв’язку для подальшого вдосконалення. У перспективі розробка може стати основою для створення національного консорціумного блокчейну, що об'єднає українські університети, забезпечуючи безпеку та ефективність обміну освітніми даними на державному рівні. Водночас доцільно розглянути інтеграцію з технологіями другого рівня, такими як Polygon, для підвищення масштабованості, а також експерименти з NFT для додання унікальності сертифікатам у цифровому просторі. Впровадження штучного інтелекту для аналізу освітніх трендів могло б надати системі аналітичний потенціал, закладаючи основу для її трансформації в інноваційну платформу, що задаватиме стандарти у глобальному контексті цифрової освіти.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. India curbs academic fraud, reshapes education sector with blockchain [Електронний ресурс] // CoinGeek. – Режим доступу: https://coingeek.com/india-curbs-academic-fraud-reshapes-education-sector-with-blockchain/ (дата звернення: 23.05.2025).

2. Модернізація вищої освіти та забезпечення якості освітньої діяльності в умовах європейської інтеграції: матеріали міжнар. наук.-практ. конф., 18 жовт. 2024 р., м. Харків / МОН України, ДБТУ. – Харків : ДБТУ, 2024. – 476 с. (дата звернення: 23.05.2025).

3. Chinnasamy P., Subashini B., Ayyasamy R.K. та ін. Blockchain based electronic educational document management with role-based access control using machine learning model // Scientific Reports. – 2025. – Vol. 15. – Article 18828. – DOI: 10.1038/s41598-025-99683-5. (дата звернення: 02.04.2025).

4. UNESCO; Commonwealth of Learning. Education and Blockchain. Paris : UNESCO, 2022. (дата звернення: 23.05.2025).

5. 'Operation Nightingale': 25 charged in scheme to sell fake nursing diplomas and transcripts [Електронний ресурс] // USA Today. – Режим доступу: https://www.usatoday.com/story/news/nation/2023/01/26/florida-nursing-schools-sold-thousands-fake-diplomas-100-million/11126832002/ (дата звернення: 04.06.2025). (дата звернення: 24.05.2025).

6. СБУ викрила міжрегіональну групу, що підробляла документи державного зразка [Електронний ресурс] // Служба безпеки України. – Режим доступу: https://ssu.gov.ua/novyny/sbu-vykryla-mizhrehionalnu-hrupu-shcho-pidrobliala-dokumenty-derzhavnoho-zrazka-video (дата звернення: 04.06.2025). (дата звернення: 24.05.2025).

7. Chan W., Gai K., Yu J., Zhu L. Blockchain-Assisted Self-Sovereign Identities on Education: A Survey // Blockchains. – 2025. – Vol. 3, No. 1. – Art. 3. (дата звернення: 31.03.2025).

8. Europass Digital Credentials Infrastructure [Електронний ресурс] / European Commission. – Режим доступу: https://ec.europa.eu/futurium/en/europass/europass-digital-credentials-infrastructure.html (дата звернення: 18.05.2025).

9. StudyPass – Study in Ukraine [Електронний ресурс] / Міністерство освіти і науки України. – Режим доступу: https://studyinukraine.gov.ua/studypass/ (дата звернення: 04.06.2025).

10. Alamsyah A., Syahrir S. The Taxonomy of Blockchain-based Technology in the Financial Industry // F1000Research. – 2023. – Vol. 2. – 248 p. – DOI: 10.12688/f1000research.129975.1. (дата звернення: 07.04.2025).

11. Kabashi F., Snopce H., Luma A., Neziri V. Trustworthy verification of academic credentials through blockchain technology // International Journal of Online and Biomedical Engineering. – 2024. – Vol. 20, No. 9. – P. 51–64. (дата звернення: 07.04.2025).

12. Опірський І.Р., Балацька В.С., Побережник В.О. Сучасні можливості використання технології блокчейн у системі освіти // Український науковий журнал з безпеки інформації. – 2023. – Т. 29, № 3. – С. 138–146. (дата звернення: 10.04.2025).

13. Bessa E.E., Martins J.S.B. A Blockchain-based Educational Record Repository (BcER²) // Proceedings of the 7th International Workshop on Advances in ICT (ICTer’19), January 2019, Praia, Cape Verde. – P. 1–11. (дата звернення: 22.05.2025).

14. Cárdenas-Quispe M.A., Pacheco A. Blockchain ensuring academic integrity with a degree verification prototype // Scientific Reports. – 2025. – Vol.

15. – Article 9281. – DOI: 10.1038/s41598-025-93913-6. (дата звернення: 18.05.2025)

15. Балацька В.С., Опірський І.Р., Побережник В.О. Концепція використання технології блокчейн у сфері освіти // Матеріали 2-ї Міжнар. конф. «Інформаційні безпекові технології» (IBIT-2023), Львів, 2023. – С. 386–388. (дата звернення: 11.04.2025).

16. Seth S. Public, Private, and Permissioned Blockchains Compared [Електронний ресурс] // Investopedia. – 02.11.2024. – Режим доступу: https://www.investopedia.com/... (дата звернення: 14.04.2025).

17. What is Consortium Blockchain? A Complete Guide [Електронний ресурс] // CFTE (електронний блог). – Режим доступу: https://blog.cfte.education/consortium-blockchain-complete-guide/ (дата звернення: 19.05.2025)

18. Turkanović M., Hölbl M., Košič K., Heričko M., Kamišalić A. EduCTX: A Blockchain-Based Higher Education Credit Platform // IEEE Transactions on Learning Technologies. – 2018. – Vol. 6. – P. 375–384. (дата звернення: 30.04.2025)

19. OpenCerts FAQ [Електронний ресурс]. – Режим доступу: https://www.opencerts.io/faq/ (дата звернення: 22.05.2025).

20. Rohaidi N. Singapore fights education fraud with blockchain-powered platform [Електронний ресурс] // GovInsider. – 2020. – Режим доступу: https://govinsider.asia/intl-en/article/singapore-education-fraud-blockchain-opencerts-steven-koh-govtech-patrice-choong-np (23.05.2025).

21. Blockcerts FAQ [Електронний ресурс]. – Розділ «Which blockchain does Blockcerts use?». – Режим доступу: https://www.blockcerts.org/guide/faq.html (20.05.2025).

22. Blockcerts – The Open Standard for Blockchain Credentials [Електронний ресурс]. – Офіційний сайт. – Режим доступу: https://www.blockcerts.org/about (22.05.2025).

23. Third non-EU country, Ukraine, joins the European Blockchain Partnership [Електронний ресурс] // Cointelegraph. – 18.06.2022. – Режим доступу: https://cointelegraph.com/news/third-non-eu-country-ukraine-joins-the-european-blockchain-partnership (дата звернення: 24.05.2025).

24. Закон України «Про вищу освіту» від 01.07.2014 № 1556-VII (зі змінами). – Режим доступу: https://zakon.rada.gov.ua/laws/show/1556-18 (01.04.2025).

25. ENIC Ukraine. Вища освіта: документи про вищу освіту (наукові ступені) [Електронний ресурс]. – Режим доступу: https://enic.in.ua/index.php/ua/zrazky-osvitnih-dokumentiv/vusha-osvita-doc (дата звернення: 01.04.2025).

26. Наказ МОН України «Про затвердження форм документів про вищу освіту (наукові ступені) та додатка до них» № 102 від 25.01.2021. – Режим доступу: https://zakon.rada.gov.ua/laws/show/z0122-21/print (дата звернення: 01.04.2025).

ДОДАТОК А. ПРИКЛАД ВИХІДНОГО КОДУ СМАРТ-КОНТРАКТУ

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

import "@openzeppelin/contracts/access/Ownable.sol";

contract CertificateSystem is Ownable {
    constructor() Ownable(msg.sender) {}

    struct Certificate {
        uint id;
        string seriesNumber;
        string studentNameUk;
        string studentNameEn;
        uint graduationYear;
        string universityUk;
        string universityEn;
        string programUk;
        string programEn;
        string degree;
        string fieldCode;
        string specialtyCode;
        uint specializationCode;
        uint issueDate;
        bytes32 txHash;
        bool isRevoked;
    }

    mapping(uint => Certificate) public certificates;

    mapping(bytes32 => uint) public seriesToId;
    uint public certificateCount;

    event CertificateIssued(
        uint indexed id,
        string seriesNumber,
        uint issueDate
    );
    event CertificateRevoked(uint indexed id, string
seriesNumber);

    function _getSeriesKey(
        string memory _seriesNumber
    ) internal pure returns (bytes32) {
        return keccak256(abi.encodePacked(_seriesNumber));
    }

    function issueCertificate(
        string memory _seriesNumber,
        string memory _studentNameUk,
```

```
            string memory _studentNameEn,
            uint _graduationYear,
            string memory _universityUk,
            string memory _universityEn,
            string memory _programUk,
            string memory _programEn,
            string memory _degree,
            string memory _fieldCode,
            string memory _specialtyCode,
            uint _specializationCode
        ) public onlyOwner {
            require(bytes(_seriesNumber).length > 0, "Series
number required");

            bytes32 key = _getSeriesKey(_seriesNumber);
            require(seriesToId[key] == 0, "Series number already
exists");

            certificateCount++;
            certificates[certificateCount] = Certificate(
                certificateCount,
                _seriesNumber,
                _studentNameUk,
                _studentNameEn,
                _graduationYear,
                _universityUk,
                _universityEn,
                _programUk,
                _programEn,
                _degree,
                _fieldCode,
                _specialtyCode,
                _specializationCode,
                block.timestamp,
                bytes32(0),
                false
            );

            seriesToId[key] = certificateCount;

            emit CertificateIssued(
                certificateCount,
                _seriesNumber,
                block.timestamp
            );
        }

        function setTxHash(uint _id, bytes32 _txHash) public
onlyOwner {
            require(_id > 0 && _id <= certificateCount, "Invalid
ID");
```

```
            require(certificates[_id].txHash   ==   bytes32(0),
"TxHash already set");
            certificates[_id].txHash = _txHash;
        }

        function verifyCertificate(
            uint _id
        ) public view returns (Certificate memory) {
            require(_id > 0 && _id <= certificateCount, "Invalid
ID");
            require(!certificates[_id].isRevoked,   "Certificate
revoked");
            return certificates[_id];
        }

        function verifyBySeriesNumber(
            string memory _seriesNumber
        ) public view returns (Certificate memory) {
            bytes32 key = _getSeriesKey(_seriesNumber);
            uint id = seriesToId[key];
            require(id != 0, "Certificate not found");
            return verifyCertificate(id);
        }

        function revokeCertificate(uint _id) public onlyOwner {
            require(_id > 0 && _id <= certificateCount, "Invalid
ID");
            require(!certificates[_id].isRevoked,       "Already
revoked");
            certificates[_id].isRevoked = true;
            emit                          CertificateRevoked(_id,
certificates[_id].seriesNumber);
        }

        function getSeriesKey(
            string memory _seriesNumber
        ) public pure returns (bytes32) {
            return _getSeriesKey(_seriesNumber);
        }
    }
```