

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

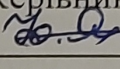
БАКАЛАВРСЬКА ДИПЛОМНА РОБОТА

на тему:

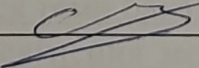
Розробка програми групового цифрового підписування смарт контрактів

Виконав: студент 4 курсу, гр. ІКІТС-206
спеціальності 125– Кібербезпека
Освітня програма – Кібербезпека
інформаційних технологій та систем
(шифр і назва напряму підготовки, спеціальності)

 Гончарук О.М.
(прізвище та ініціали)

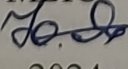
Керівник: д.т.н., професор
 Яремчук Ю.Є.
(прізвище та ініціали)

« 6 » серпня 2024 р.

Рецензент: к.т.н., доцент, професор каф. ОТ
 Захарченко С.М.
(прізвище та ініціали)

« 6 » серпня 2024 р.

Допущено до захисту

Голова секції УБ кафедри МБІС
д.т.н., проф. Яремчук Ю.Є. 
« 6 » серпня 2024р.

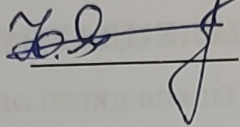
Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

Рівень вищої освіти I-й (бакалаврський)
Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека

Освітньо-професійна програма - Кібербезпека інформаційних технологій та систем

ЗАТВЕРДЖУЮ

Голова секції УБ, кафедра МБІС



д.т.н., проф. Яремчук Ю.Є.
“ 22 ” березня 2024 р.

З А В Д А Н Н Я

на бакалаврську дипломну роботу студенту

Гончарук Олександр Михайлович

(прізвище, ім'я, по-батькові)

1. Тема роботи Розробка програми групового цифрового підписування смарт контрактів

Керівник роботи Яремчук Юрій Євгенович д.т.н., професор

(прізвище, ім'я, по-батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “11” березня 2024 року
№ 80

2. Термін подання студентом роботи _____

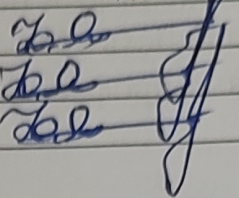
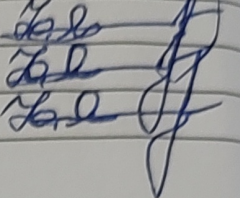
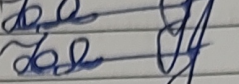
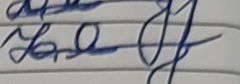
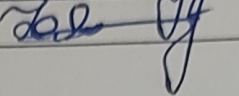
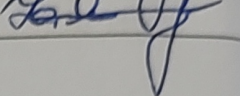
3. Вихідні дані до роботи Електронні джерела та наукові статті по темі. Існуюче програмне забезпечення, яке стосується теми бакалаврської дипломної роботи

4. Зміст текстової частини: Для досягнення мети було поставлено такі задачі: провести аналіз предметної області для вибору найкращих інструментів для вирішення поставленого завдання, провести аналіз необхідних матеріалів для роботи з мовами програмування, провести аналіз необхідних матеріалів для роботи з базами даних, обрати архітектуру проектованою інформаційної системи, провести аналіз предметної області, виробити специфікацію вимог проектованої інформаційної системи, провести програмну реалізацію інтерфейсу користувача системи, провести програмну реалізацію бекенд частини системи для взаємодії з базою даних, провести тестування розробленої системи, провести аналіз результатів роботи

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень)

схема алгоритму реєстрації, схема алгоритму авторизації, схема створення нового смарт-контракту, схема підпису наявного смарт-контракту

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ I	Яремчук Ю.Є. д.т.н., професор		
Розділ II	Яремчук Ю.Є. д.т.н., професор		
Розділ III	Яремчук Ю.Є. д.т.н., професор		

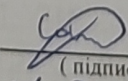
7. Дата видачі завдання 22 березня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

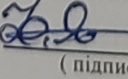
№	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз предметної області	23.03.2024	30.03.2024	Виконано
2	Аналіз методів розв'язання задачі	1.04.2024	10.04.2024	Виконано
3	Постановка задач розробки програмного модулю	11.04.2024	20.04.2024	Виконано
4	Дослідження інформаційних джерел	21.04.2024	10.05.2024	Виконано
5	Аналіз варіантів роботи модулів	11.05.2024	15.05.2024	Виконано
6	Розробка алгоритму роботи	16.05.2024	20.05.2024	Виконано
7	Реалізація програмного модулю	21.05.2024	24.05.2024	Виконано
8	Тестування програмного модулю	24.05.2024	26.05.2024	Виконано
9	Оформлення матеріалів до захисту БДР	26.05.2024	31.05.2024	Виконано

Студент

Керівник роботи


(підпис)

Гончарук О.М.
(прізвище та ініціали)


(підпис)

Яремчук Ю.Є.
(прізвище та ініціали)

АНОТАЦІЯ

Об'єкт розробки: інформаційна система групового цифрового підписування смарт контрактів.

Предмет розробки : веб – орієнтовані програмні засоби для проектування та розробки інформаційної системи групового цифрового підписування смарт контрактів.

Мета дипломного проекту: розробка інформаційної системи для автоматизації процесів групового цифрового підписування смарт контрактів.

У першому розділі наведено теоретичні відомості про використані інструменти для вирішення задачі, проведено огляд аналіз існуючих аналогів , проведено аналіз та порівняльну характеристику технологій для розробки проектного ПЗ, зроблено вибір інструментів для вирішення поставленого завдання на основі проведеного аналізу.

У другому розділі проведено аналіз предметної області та специфікації вимог, надана коротка характеристика об'єкту автоматизації та предметної галузі, наведено специфікацію вимог до системи.

У третьому розділі проведено практичну реалізацію проектного системи, надано інформаційне, технічне та системне забезпечення розробки.

Актуальність розробки проектного системи визначається великим попитом на подібні системи, оскільки автоматизація процесів робить процеси підписання контрактів більш гнучкими, зрозумілими, відкритими та стандартизованими, а також надає значні переваги : мобільність, безконтактність, ціна, масштабованість і т.ін.

Список ключових слів: АВТОМАТИЗАЦІЯ, ІНФОРМАЦІЙНА СИСТЕМА, СМАРТ – КОНТРАКТ, БАЗА ДАНИХ, PHP, HTML, CSS, MYSQL.

ABSTRACT

Object of development: information system for group digital signing of smart contracts.

Development subject: web-oriented software tools for designing and developing an information system for group digital signing of smart contracts.

The goal of the diploma project: development of an information system for automating processes of group digital signing of smart contracts.

The first section provides theoretical information about the tools used to solve the problem, an analysis and comparative characteristics of technologies for the development of the designed software are carried out, a selection of tools for solving the task is made based on the analysis.

In the second section, an analysis of the subject area and specification of requirements was carried out, a brief description of the automation object and the subject area was provided, an overview and analysis of existing analogues was carried out, and a specification of requirements for the system was given.

In the third section, practical implementation of the designed system was carried out, informational, technical and system development support was provided.

The relevance of the development of the designed system is determined by the great demand for such systems, since the automation of processes makes the processes of signing contracts more flexible, understandable, open and standardized, and also provides significant advantages: mobility, contactlessness, price, scalability, etc.

List of keywords: AUTOMATION, INFORMATION SYSTEM, SMART - CONTRACT, DATABASE, PHP, HTML, CSS, MYSQL.

ЗМІСТ

ВСТУП.....	3
1. ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО ВИКОРИСТАНІ ІНСТРУМЕНТИ	5
ДЛЯ ВИРІШЕННЯ ЗАДАЧІ	5
1.1 Огляд та аналіз існуючих аналогів групового цифрового підписування смарт контрактів..	6
1.2 Коротка характеристика об'єкту інформаційна система групового цифрового підписування смарт контрактів	13
1.3 Аналіз існуючих варіантів вирішення поставленого завдання	14
1.4 Аналіз існуючих СУБД для вирішення поставленої задачі.....	23
1.5 Вибір інструментів для вирішення поставленого завдання на основі проведеного аналізу	29
1.6 Висновок до розділу 1	30
2. ПРОЕКТУВАННЯ ЦИФРОВОГО ПІДПISУВАННЯ СМАРТ КОНТРАКТІВ.....	31
2.1. Розробка архітектури проектованої системи	31
2.2 Розробка запропонованих алгоритмів роботи системи.	35
2.4 Проектування бази даних.....	42
2.5 Висновок до розділу 2.....	45
3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТОВАНОГО ПРОГРАМНОГО ПРОДУКТУ	47
3.1 Призначення програмних модулів	48
3.2 Програмна реалізація header и footer для веб-сторінок	49
3.3 Програмна реалізація інтерфейсу головної сторінки	50
3.4 Програмна реалізація інтерфейсу взаємодії зі смарт - контрактом.....	54
3.5 Програмна реалізація логіки зміни статусу смарт - контракта.....	58
3.6 Програмна реалізація інтерфейсу створення нового смарт - контракту.....	61
3.7 Програмна реалізація інтерфейсу написання відгуків щодо використання системи	62
3.8 Програмна реалізація логіки реєстрації у системі	64
3.9 Програмна реалізація сторінки авторизації	67
3.10 Висновок до розділу 3	69
ВИСНОВОК	71
СПИСОК ЛІТЕРАТУРИ	72
ДОДАТКИ	74
ДОДАТОК А. Технічне завдання.....	75
ДОДАТОК Б. Програмний код.....	82
ДОДАТОК В. Ілюстративний матеріал	93
ДОДАТОК Г. Протокол перевірки на антиплагіат.....	101

ВСТУП

Автоматизація будь-якого виду надання послуг – завдання на сьогоднішній день відоме та поширене. У сучасному світі не один бізнес не може обійтися без інтеграції в свою роботу інформаційної системи в якомусь її представленні, оскільки це істотно підвищує як зручність і якість взаємодії всередині бізнесу, так і значно підвищує охоплення клієнтської аудиторії та можливості взаємодії з нею, реклами.

Можна говорити про велику кількість успішно впроваджених проектів у різних галузях, у різних за масштабом компаніях, з застосуванням різних засобів впровадження процесів автоматизації. Коли ми говоримо про розробку інформаційної системи групового цифрового підписування смарт контрактів, ми маємо на увазі розробку інтернет-сервісу для автоматизації процесів створення та підписання контрактів умовними групами користувачів з автоматичною функцією активації контракту при наявності підписів всіх зацікавлених сторін контракту. Фактично поняття веб-сервісу стало невід'ємною частиною будь-якого бізнесу у сфері документообігу. Результати аналізу існуючих даних свідчать, що нині більшість як великих мережових компаній, або невеликих локальних тією чи іншою мірою переглянули свій підхід до структури та процесів керування своїм бізнесом у бік автоматизації.

Метою кваліфікаційної роботи є розробка інформаційної системи для автоматизації процесів групового цифрового підписування смарт контрактів. Для досягнення мети в роботі поставлено й вирішено такі теоретичні й практичні завдання:

- Провести аналіз предметної області для вибору найкращих інструментів для вирішення поставленого завдання
- Провести аналіз необхідних матеріалів для роботи з мовами програмування
- Провести аналіз необхідних матеріалів для роботи з базами даних
- Обрати архітектуру проектованою інформаційної системи
- Провести аналіз предметної області

- Виробити специфікацію вимог проектованої інформаційної системи
- Провести програмну реалізацію інтерфейсу користувача системи
- Провести програмну реалізацію бекенд частини системи для взаємодії з базою даних
- Провести тестування розробленої системи
- Провести аналіз результатів роботи

Об'єктом дослідження є інформаційна система групового цифрового підписування смарт контрактів.

Предметом дослідження є веб – орієнтовані програмні засоби для проектування та розробки інформаційної системи групового цифрового підписування смарт контрактів.

Методи дослідження. У процесі роботи застосовувались загальнонаукові та спеціальні методи, які дозволили вивчити предмет та об'єкт дослідження, дослідити напрями та шляхи оптимізації процесу проектування та розробки інформаційної системи групового цифрового підписування смарт контрактів.

Практичне значення одержаних результатів. Практичне значення проектованої системи визначається великим попитом на подібні системи, оскільки автоматизація процесів робить процеси підписання контрактів більш гнучкими, зрозумілими, відкритими та стандартизованими, а також надає значні переваги : мобільність, безконтактність, ціна, масштабованість і т.ін.

1. ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО ВИКОРИСТАНІ ІНСТРУМЕНТИ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ

У сучасному світі технології блокчейн і смарт контракти стали невід'ємною частиною розвитку багатьох сфер, включаючи фінанси, право, управління даними та інші. Однією з ключових проблем, які виникають у процесі застосування смарт контрактів, є забезпечення їх надійного та безпечного підписання учасниками угод. Зокрема, у випадках, коли необхідне групове підписання смарт контрактів, важливо забезпечити прозорість, достовірність та автентичність цифрових підписів усіх учасників. У цьому контексті розглядається інформаційна система групового цифрового підписування смарт контрактів.

Метою даного розділу є огляд та аналіз існуючих аналогів групового цифрового підписування смарт контрактів, а також вибір оптимальних інструментів для реалізації цієї задачі. Для досягнення поставленої мети будуть розглянуті наступні аспекти:

1. Коротка характеристика об'єкту дослідження - інформаційної системи групового цифрового підписування смарт контрактів.
2. Аналіз існуючих варіантів вирішення задачі групового цифрового підписування.
3. Аналіз існуючих систем управління базами даних (СУБД), які можуть бути використані для забезпечення зберігання та управління даними в рамках даної системи.
4. Вибір інструментів для реалізації поставленого завдання на основі проведеного аналізу.

Цей розділ спрямований на формування цілісного уявлення про поточний стан технологій у сфері групового цифрового підписування смарт контрактів, а також на виявлення найбільш перспективних підходів та інструментів для створення ефективної та безпечної інформаційної системи.

1.1 Огляд та аналіз існуючих аналогів групового цифрового підписування смарт контрактів

Перш ніж почати розробляти певну систему, потрібно виконати пошук її аналогів, визначити їх переваги та недоліки, щоб не допустити ті ж помилки та використовувати переваги проаналізованих систем. Предметною галузю проектованої системи є підписування смарт - контрактів, тому розглянемо веб-ресурси, що реалізують схожий функціонал.

Проаналізовано три веб-ресурси, що реалізують функції предметної галузі:

- 1) Платформа «DocuSign».
- 2) Платформа «Adobe Sign».
- 3) Платформа «HelloSign».

Розглянемо докладніше подані вище аналоги. Першою розглянемо платформу DocuSign :

На рисунку 1.1 зображено інтерфейс користувача наведеної інформаційної системи.

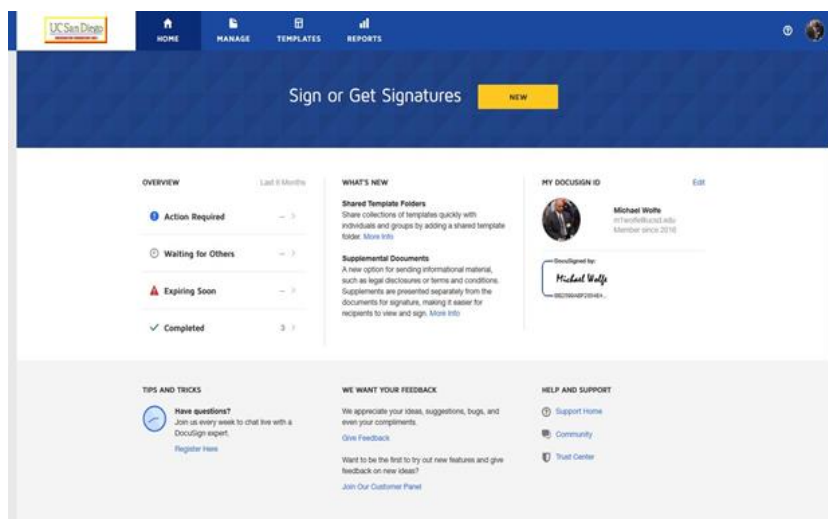


Рисунок 1.1 - Інтерфейс користувача DocuSign

DocuSign - це платформа для електронного підписування документів, яка надає зручний та безпечний спосіб створювати, підписувати та надсилати документи для підписування в онлайні. [1]

Основні можливості DocuSign включають:

- Створення документів: Платформа дозволяє створювати нові документи або завантажувати існуючі документи для підписування.
- Електронний підпис: DocuSign дозволяє користувачам електронно підписувати документи без потреби в ручному підписі. Електронний підпис відповідає всім законодавчим вимогам для юридичної валідності.
- Надсилання для підпису: Після створення або завантаження документів користувач може надіслати їх для підписування відповідним особам.
- Стеження за статусами: DocuSign надає засоби для стеження за статусами підписання документів. Користувач може переглядати історію підписів та отримувати сповіщення про зміни статусів.
- Інтеграція з іншими системами: DocuSign має API, який дозволяє інтегрувати платформу з іншими веб-системами та додатками. [1]

Розглянемо переваги та недоліки використання платформи DocuSign:

Переваги:

1. Зручність: DocuSign надає зручний і легкий у використанні інтерфейс для створення, підпису та надсилання документів.
2. Широкий функціонал: Платформа пропонує широкий спектр функцій, включаючи створення, підпис та стеження за статусами документів.
3. Електронний підпис: DocuSign забезпечує електронний підпис, який відповідає законодавчим вимогам і є юридично визнаним у багатьох країнах.
4. Безпека: Платформа використовує різні методи шифрування та захисту даних для забезпечення безпеки конфіденційної інформації.
5. Інтеграція: DocuSign має API для інтеграції з іншими системами, що дозволяє легко впроваджувати платформу у вже існуючі робочі процеси.

Недоліки:

1. Вартість: Використання платформи DocuSign може бути витратним, особливо для невеликих компаній або індивідуальних користувачів.

2. Залежність від інтернету: DocuSign потребує підключення до Інтернету для доступу до своїх функцій, що може становити проблему в умовах обмеженого або незабезпеченого зв'язку.

3. Спірність електронного підпису: У деяких країнах електронний підпис може викликати сумніви щодо його юридичної валідності у певних випадках.

4. Обмеження функціоналу в безкоштовних версіях: Деякі функції можуть бути доступні лише в платних версіях DocuSign, що може становити обмеження для користувачів без підписки.

Ця платформа надає зручний та безпечний спосіб електронного підписування документів, що робить її привабливим вибором для організацій будь-якого розміру. Хоча DocuSign має свої переваги та недоліки, вона залишається однією з найпопулярніших та надійних платформ для електронного підписування документів, особливо у сфері бізнесу.

Наступним розглянемо платформу Adobe Sign.

На рисунку 1.2 зображено інтерфейс користувача наведеної інформаційної системи.

Get documents signed
Send an agreement to others for e-signing, approval, or other processing.

Send from: HR (Primary Group) **1**

Recipients*
Complete in Order ☐ Complete in Any Order ☒ [Add Me](#) [Add Recipient Group](#) **2**

Enter recipient email

Show CC

Message* **3**
Agreement Name
Please review and complete this document.

Files* **5** [Add Files](#)
Drag & Drop Files Here

☐ Preview & Add Signature Fields

Options **4**
☐ Password Protect
☐ Set Reminder
Signature Type
☒ Electronic ☐ Written
Recipients' Language
English: US

Send

Рисунок 1.2 - Інтерфейс користувача Adobe Sign

Adobe Sign - це платформа для електронного підписування документів, що належить до складу Adobe Document Cloud. Вона дозволяє користувачам

створювати, підписувати та надсилати документи для підпису в онлайні, що спрощує процес підписування і управління документами. [2]

Основні можливості Adobe Sign включають:

- Створення документів: Платформа дозволяє користувачам створювати нові документи або завантажувати існуючі для підпису.
- Електронний підпис: Adobe Sign надає можливість користувачам електронно підписувати документи без необхідності вручного підпису.
- Надсилання для підпису: Після підготовки документів користувач може надіслати їх для підписування відповідним особам.
- Система стеження за статусами: Adobe Sign надає інструменти для стеження за статусами підписання документів, що дозволяє користувачам відстежувати процес підписування та контролювати його.
- Інтеграція з іншими сервісами: Платформа підтримує інтеграцію з іншими сервісами Adobe, такими як Adobe Acrobat та Adobe Creative Cloud, що полегшує роботу з документами і додатковими сервісами Adobe. [2]

Розглянемо переваги та недоліки платформи Adobe Sign:

Переваги:

1. Широкий функціонал: Adobe Sign має розгорнутий набір функцій, який включає створення, підписування та надсилання документів для підпису, а також стеження за їх статусами.

2. Інтеграція з іншими сервісами Adobe: Платформа добре інтегрується з іншими сервісами Adobe, що робить процес роботи з документами ще більш зручним.

3. Електронний підпис: Adobe Sign забезпечує можливість використовувати електронний підпис, що дозволяє користувачам підписувати документи в онлайні без необхідності ручного підписування.

4. Безпека даних: Платформа використовує різні методи шифрування та захисту даних для забезпечення безпеки конфіденційної інформації.

5. Інтеграція з іншими системами: Adobe Sign має API для інтеграції з іншими веб-системами та додатками, що дозволяє легко впроваджувати платформу у різноманітні бізнес-процеси.

Недоліки:

1. Вартість: Використання Adobe Sign може бути витратним, особливо для малих підприємств або індивідуальних користувачів.

2. Залежність від інтернет-підключення: Adobe Sign потребує підключення до Інтернету для доступу до своїх функцій, що може становити проблему в умовах обмеженого або непостійного зв'язку.

3. Спірність електронного підпису: У деяких країнах можуть виникати сумніви щодо юридичної валідності електронного підпису.

4. Обмеження функціоналу в безкоштовній версії: Деякі функції можуть бути доступні лише в платних версіях Adobe Sign, що може стати обмеженням для користувачів без підписки.

Adobe Sign - це потужний інструмент для електронного підписування документів, який пропонує широкий спектр функцій і зручний інтерфейс. Використання Adobe Sign може значно спростити процес підписування документів та збереження цифрових копій, що робить його популярним вибором серед бізнес-користувачів і індивідуальних клієнтів. Хоча Adobe Sign має свої переваги та недоліки, вона залишається однією з найпопулярніших та надійних платформ для електронного підписування документів, особливо у сфері бізнесу. [2]

Останнім кроком розглянемо платформу HelloSign.

На рисунку 1.3 зображено інтерфейс користувача наведеної інформаційної системи.

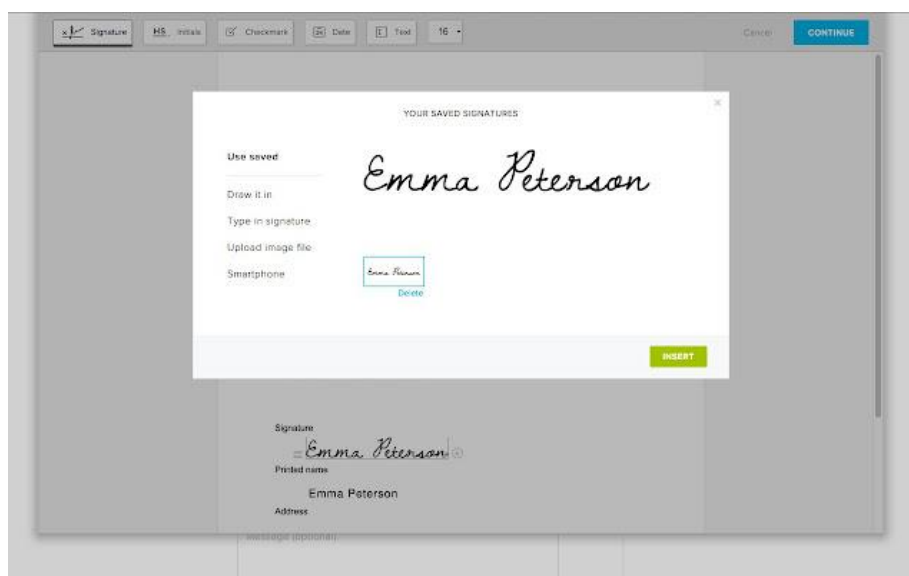


Рисунок 1.3 – Інтерфейс користувача HelloSign

HelloSign - це платформа для електронного підписування документів, яка надає зручні інструменти для створення, підпису та надсилання документів для підписування в онлайні. [3]

Основний функціонал HelloSign включає:

- Створення документів: HelloSign дозволяє користувачам створювати нові документи або завантажувати вже існуючі для підписування.
- Електронний підпис: Платформа надає можливість використовувати електронний підпис для підписування документів в онлайні.
- Надсилання для підпису: Після підготовки документів користувач може надіслати їх для підписування відповідним особам.
- Система стеження за статусами: HelloSign надає засоби для стеження за статусами підписання документів, що дозволяє користувачам контролювати процес підписування та отримувати сповіщення про зміни статусів.
- Інтеграція з іншими сервісами: HelloSign підтримує інтеграцію з різними сервісами та платформами, що полегшує роботу з документами та управління процесом підписування. [3]

Розглянемо переваги та недоліки системи HelloSign:

Переваги:

1. Простий у використанні інтерфейс: HelloSign має інтуїтивно зрозумілий інтерфейс, що робить процес створення та підписування документів легким і зручним для користувачів.

2. Електронний підпис: Платформа надає можливість використовувати електронний підпис, що забезпечує швидке та ефективне підписування документів в онлайні.

3. Система стеження за статусами: HelloSign надає засоби для стеження за статусами підписання документів, що дозволяє користувачам легко контролювати процес підписування та отримувати сповіщення про зміни статусів.

4. Інтеграція з іншими сервісами: Платформа підтримує інтеграцію з іншими сервісами та платформами, такими як Google Drive, Dropbox, Salesforce тощо, що полегшує роботу з документами та управління процесом підписування.

5. Безпека даних: HelloSign забезпечує високий рівень захисту конфіденційної інформації за допомогою шифрування та інших методів безпеки.

Недоліки:

1. Обмежений безкоштовний план: Безкоштовна версія HelloSign має обмежений функціонал, і деякі додаткові функції доступні лише за плату.

2. Вартість платних планів: Платні плани HelloSign можуть бути відносно дорогими для невеликих бізнесів або індивідуальних користувачів.

3. Залежність від інтернет-підключення: Використання HelloSign вимагає підключення до Інтернету, що може стати обмеженням у випадках відсутності зв'язку.

4. Спірність електронного підпису: В деяких юрисдикціях електронний підпис може викликати сумніви щодо її юридичної валідності.

HelloSign - це зручна та надійна платформа для електронного підписування документів, яка надає широкий функціонал для створення, підписування та надсилання документів в онлайні. Її простий у використанні інтерфейс робить процес підписування документів ефективним та зручним для користувачів. Хоча HelloSign має свої недоліки, вона залишається популярним інструментом для

електронного підписування документів, особливо для бізнес-користувачів, які шукають простий та ефективний спосіб управління документами.

Після огляду систем-аналогів було проаналізовано всі переваги та недоліки цих ресурсів. З наведеного аналізу можна зробити висновок щодо комплексу задач, які треба реалізувати в проектованій системі.

Проектована система повинна мати можливість в автоматичному режимі надавати можливість користувачам повністю адмініструвати процес підписування смарт - контрактів, при цьому забезпечуючи максимально інтуїтивний та простий інтерфейс та безкоштовність використання. Тому під час розробки є необхідність зосередитись на максимально зручному способі взаємодії з користувачем та мінімізації витрат на розробку. Також значною перевагою є можливість отримати зворотній відгук від користувача, тобто потрібно також впровадити функціонал написання відгуку щодо використання системи.

Основним недоліком є відсутність перерахованих вище функцій, а також не інтуїтивний інтерфейс та велика вартість використання, що ускладнює роботу з системою.

1.2 Коротка характеристика об'єкту інформаційна система групового цифрового підписування смарт контрактів

Проектований об'єкт є інформаційною системою для автоматизації процесів підписування смарт контрактів між віддаленими групами користувачів.

Смарт-контракт - це програмний код, який зберігається на блокчейні і автоматизує, виконує та забезпечує виконання угод між сторонами без посередництва. Основна ідея полягає в тому, що умови угоди вбудовуються безпосередньо в код контракту, який виконується автоматично при виконанні певних умов. Цей код зазвичай виконується в децентралізованій обчислювальній мережі, такий як блокчейн, і забезпечує безпосередню взаємодію між учасниками угоди. Коли виконуються певні умови, передбачені в коді смарт-контракту, відбувається автоматичний перехід від одного стану угоди до іншого, без необхідності додаткових дій з боку учасників.

Це дозволяє уникнути потреби в довірі до третьої сторони для виконання угоди, оскільки логіка виконання угоди закладена безпосередньо в коді контракту. Крім того, смарт-контракти забезпечують високий рівень безпеки та недерегульованість, оскільки вони виконуються в децентралізованому середовищі блокчейну.

Проектована інформаційна система має надавати можливість умовним користувачам зареєструватися у додатку, створити новий смарт-контракт, який потребує групового цифрового підписування, побачити у режимі online інформацію щодо обраної угоди, додати учасників для підписування угоди, виконувати перевірку підписів та активацію або скасування угоди.

На сьогоднішній день існує велика потреба в автоматизації процесів підписування смарт контрактів між віддаленими групами користувачів, тому потрібно мати власну стратегію, щоб забезпечити тривалий та ефективний веб – додаток автоматизації цієї галузі. Для того, щоб групі користувачів заключити контракт, їм необхідно їхати до точки збору, займатися безліччю бумажної роботи, що в наш час сучасних технологій не є оптимальним рішенням. Для вирішення цієї проблеми найбільш ефективним способом буде створення інформаційної системи, з функціоналом, описаним вище.

Дана система дозволить користувачеві автоматизувати основні процеси підписування смарт контрактів між віддаленими групами користувачів, що сприяє підвищенню зручності та швидкості обслуговування потенційних клієнтів.

1.3 Аналіз існуючих варіантів вирішення поставленого завдання

Коли починається етап розробки програми групового цифрового підписування смарт контрактів, перше з чим розробник обов'язково зіштовхнеться це HTML (мова гіпертекстової розмітки) і CSS (каскадні таблиці стилів), але це всього лише інструменти для розмітки сторінки, для побудови чогось більшого нам обов'язково доведеться зробити вибір мови програмування, що найбільше підходить для вирішення нашого завдання.

HTML та CSS є мовами розмітки, які використовуються для створення електронних документів (або сторінок) та для розробки сайтів відповідно. Тим

часом, мови веб-програмування складніші і можуть бути поділені на дві категорії: внутрішні та зовнішні мови веб-розробки.

Зазвичай веб-розробка включає кодування на стороні сервера (внутрішня частина), кодування на стороні клієнта (зовнішня частина) та технології баз даних.

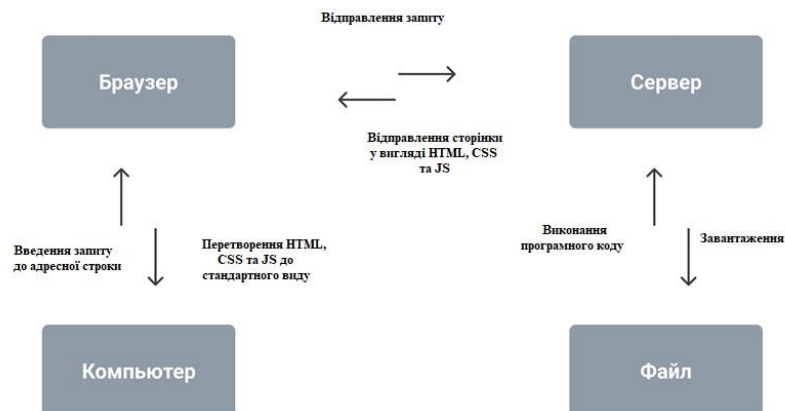


Рисунок 1.4 - Схема роботи типового веб- додатку

Як приклад візьмемо LinkedIn - найбільшу у світі професійну мережу. Щоб дозволити кандидатам з різних регіонів спілкуватися з іншими, шукати роботу або навіть набувати нових навичок, LinkedIn був запрограмований трьома різними мовами веб-програмування: JavaScript, Java та Scala, а потім використав Voldemort як розподілену базу даних для зберігання величезної кількості профілів.

Відповідно, мови для веб-розробки можна визначити як "складні логічні інструкції та процеси", які допомагають створювати сайти, що відповідають певним вимогам.

Отже, розглянемо найпопулярніші мови веб-розробки та визначимо, яка з них найкраще підійде для вирішення нашого завдання.

JavaScript є однією з найбільш важливих мов програмування в сфері веб-розробки, надаючи динамічність та інтерактивність веб-сторінкам. Ця мова дозволяє розробникам створювати ефективні та потужні клієнтські застосунки, які взаємодіють з користувачем без необхідності перезавантаження сторінки.

JavaScript використовується для реалізації різноманітних функцій, таких як валідація форм, анімації, маніпуляція DOM (об'єктний представник документу) для

динамічного оновлення вмісту сторінки та взаємодії з сервером за допомогою технології AJAX.

Завдяки вбудованому движку JavaScript у браузерях, використання цієї мови веб-програмування дозволяє створювати багатофункціональні та естетично привабливі інтерфейси користувача. Зараз JavaScript активно застосовується в сучасному веб-середовищі, включаючи велику кількість бібліотек та фреймворків, таких як React, Angular та Vue.js, що спрощують розробку веб-застосунків.

Однією з ключових переваг JavaScript є його можливість використання на обох сторонах веб-застосунку: на стороні клієнта та на стороні сервера. Наприклад, за допомогою Node.js, розробники можуть використовувати JavaScript для написання серверного коду, розширюючи його застосування на всій веб-платформі.

Узагальнюючи, JavaScript визначається своєю гнучкістю та широким спектром застосувань, роблячи його ключовим елементом сучасної веб-розробки.

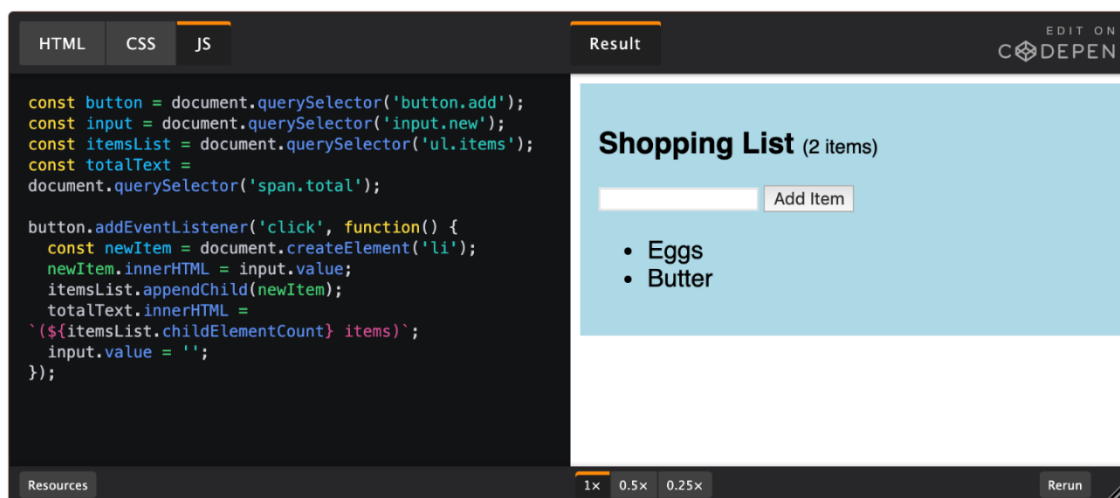


Рисунок 1.5 – JavaScript в контексті веб - розробки

Python є потужною та високорівневою мовою програмування, яка стала невід'ємною частиною веб-розробки завдяки своїй зручності та ефективності. Ця мова пропонує розробникам можливість швидко створювати стабільні та масштабовані веб-застосунки.

Однією з ключових переваг Python в контексті веб-розробки є фреймворки, які спрощують розробку та дозволяють створювати високоякісні веб-застосунки з

меншим обсягом коду. Flask та Django - це два найпопулярніших фреймворки для розвитку веб-застосунків на Python.

Django вражає своєю повнотою та конвенціями над конфігурацією, що прискорює розробку, особливо для великих проектів. З іншого боку, Flask визначається своєю легкістю та гнучкістю, що робить його ідеальним вибором для менших або простих веб-застосунків.

Поза фреймворками, Python використовується для реалізації різноманітних завдань у веб-розробці, включаючи обробку запитів, взаємодію з базами даних, створення API та розробку скриптів для автоматизації рутинних завдань.

Загальна простота та читабельність коду Python роблять його привабливим для розробників та сприяють швидкому розвитку проектів. Завдяки своїй універсальності, Python стає незамінним інструментом в сфері веб-розробки, дозволяючи створювати як великі корпоративні системи, так і невеликі стартапи.

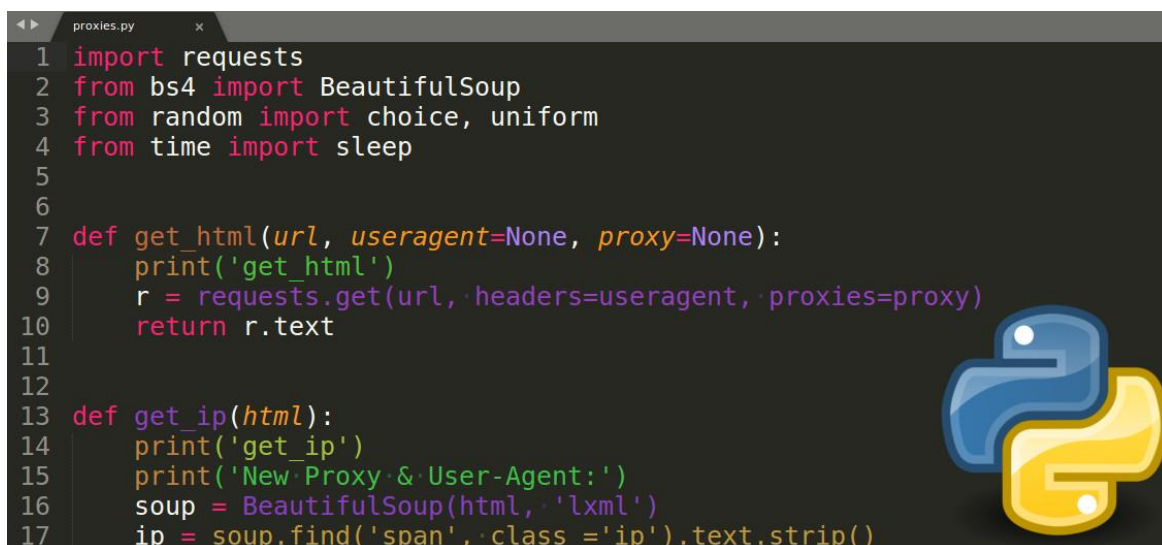


Рисунок 1.6 – Python в контексті веб - розробки

PHP (Hypertext Preprocessor) є однією з найпоширеніших мов програмування для веб-розробки, використовуваною для створення динамічних та інтерактивних веб-сайтів. Ця мова є основною складовою серверної сторони веб-додатків і широко використовується завдяки своїй простоті та ефективності.

Однією з головних переваг PHP є його легкість в освоєнні та швидкість в розробці. Мова підтримує велику кількість вбудованих функцій, які полегшують роботу з рядками, масивами, роботою з базами даних та обробкою форм.

PHP зазвичай використовується в поєднанні з сервером Apache та базою даних MySQL (або іншими системами управління базами даних), створюючи надійний стек технологій для веб-розробки, відомий як LAMP (Linux, Apache, MySQL, PHP).

Однією з сильних сторін PHP є його вбудована підтримка великої кількості веб-серверів та операційних систем, що робить його універсальним інструментом для розробки веб-додатків різного роду та масштабів.

Фреймворки, такі як Laravel, Symfony, або CodeIgniter, роблять роботу з PHP більш організованою та ефективною, дозволяючи розробникам швидше створювати високоякісні веб-застосунки.

Усі ці особливості роблять PHP важливим інструментом у світі веб-розробки, зокрема для створення динамічних сторінок, обробки форм, роботи з базами даних та реалізації серверної логіки.

```
<?php

namespace Email;

class Client
{
    public function send(string $emailAddress, Message $message): bool
    {
        if (!$this->validateParameters($emailAddress, $message)) {
            return false;
        }

        return $this->sendToMailServer([
            'to' => $emailAddress,
            'text' => $message->text(),
        ]);
    }
}
```

Рисунок 1.7 – PHP в контексті веб - розробки

Ruby є мовою програмування, яка отримала визнання завдяки своїй простоті та елегантності, і вона активно використовується в галузі веб-розробки. Особливо популярна у спільноті розробників стала фреймворк Ruby on Rails, який спрощує та прискорює процес створення веб-додатків.

Однією з ключових рис Ruby on Rails є концепція "зручності перед конфігурацією", яка дозволяє розробникам швидко розпочати роботу з проектом, не витрачаючи багато часу на налаштування. Rails пропонує велику кількість

готових рішень та конвенцій, що полегшують розробку та підтримку веб-застосунків.

Ruby дозволяє розробникам писати читабельний та ефективний код, що сприяє швидкій розробці та підтримці проектів. Мова підтримує об'єктно-орієнтований підхід до програмування та принцип DRY (Don't Repeat Yourself), що сприяє високій якості коду та його повторному використанню.

Ruby також визначається своєю гнучкістю та динамічністю, що дозволяє розробникам миттєво змінювати та розширювати функціонал веб-додатків. Мова активно використовується для розробки стартапів, соціальних мереж, електронної комерції та багатьох інших типів веб-застосунків.

Узагальнюючи, Ruby в контексті веб-розробки, особливо з використанням фреймворка Ruby on Rails, надає розробникам інструменти для швидкої та ефективної розробки високоякісних веб-додатків.

```

1  require "time"
2
3  class InvoiceItem
4    attr_reader :id, :item_id, :invoice_id, :created_at
5
6    attr_accessor :quantity, :unit_price, :updated_at
7
8    def initialize(params)
9      @id = params[:id].to_i
10     @item_id = params[:item_id].to_i
11     @invoice_id = params[:invoice_id].to_i
12     @quantity = params[:quantity].to_i
13     @unit_price = BigDecimal(params[:unit_price])
14     @created_at = Time.parse(params[:created_at].to_s)
15     @updated_at = Time.parse(params[:updated_at].to_s)
16   end
17
18   def unit_price_to_dollars
19     @unit_price.to_f
20   end
21 end
22

```

Рисунок 1.8 – Ruby в контексті веб – розробки

Java є однією з найпопулярніших та універсальних мов програмування, і вона широко використовується в сфері веб-розробки для створення потужних та масштабованих веб-додатків. Її використання поширюється від корпоративних систем до малих веб-застосунків.

Однією з ключових переваг Java є платформна незалежність, завдяки використанню віртуальної машини Java (JVM). Це означає, що код, написаний на

Java, може виконуватися на будь-якому пристрої чи сервері, який підтримує JVM, що робить його ідеальним для створення крос-платформених веб-додатків.

Java використовується в різноманітних веб-технологіях, включаючи JavaServer Faces (JSF), Spring Framework, та Apache Struts. Фреймворки, такі як Spring, надають розробникам потужні інструменти для створення ефективних та високоякісних веб-додатків. Spring, зокрема, дозволяє розробникам реалізовувати інверсію управління (IoC) та аспектно-орієнтоване програмування (AOP), що полегшує управління компонентами та забезпечує підтримку багатомодульності.

Java також використовується для створення масштабованих веб-сервісів за допомогою технології Java API for RESTful Web Services (JAX-RS) та платформи Java Message Service (JMS) для обміну повідомленнями між різними частинами веб-додатку.

Загальна надійність, широка підтримка спільноти та велика кількість інструментів роблять Java привабливим вибором для розробки складних веб-застосунків у корпоративному середовищі.

```
def get_symbols(file_name):
    with open(file_name, "r") as in_file:
        records = []
        count = 0
        symbol_set = ""
        for line in in_file:
            symbol_set = symbol_set + line[:-1] + ','
            count = count + 1
            if count % 50 == 0:
                records.append(symbol_set)
                symbol_set = ""

        symbols.append(symbol_set)
    return records
```

Рисунок 1.9 – Java в контексті веб – розробки

C# (C-Sharp) є мовою програмування, розробленою компанією Microsoft, і вона широко використовується в сфері веб-розробки для створення сучасних та потужних веб-додатків під платформу .NET. C# володіє великою популярністю, особливо серед розробників, які працюють з технологіями Microsoft.

Однією з ключових переваг C# є його інтеграція з платформою ASP.NET, яка надає розробникам різноманітні інструменти та фреймворки для швидкої розробки веб-застосунків. ASP.NET MVC (Model-View-Controller) і ASP.NET Core є двома основними фреймворками, що дозволяють створювати добре організовані та ефективні веб-додатки.

C# використовується для розробки як серверної, так і клієнтської частини веб-застосунків. За допомогою ASP.NET Web API, розробники можуть легко створювати RESTful веб-служби для взаємодії з клієнтами, а ASP.NET SignalR дозволяє реалізовувати в реальному часі обмін даними між сервером і клієнтом.

Мова також підтримує концепції об'єктно-орієнтованого програмування, які полегшують розробку та підтримку коду. Використання Microsoft Visual Studio, інтегрованого середовища розробки (IDE) для C#, робить процес програмування зручним та продуктивним.

Загалом, C# є потужною та сучасною мовою програмування для веб-розробки, особливо в екосистемі Microsoft, і вона використовується для створення різноманітних веб-додатків, від невеликих сайтів до складних корпоративних систем.

```
public async Task<User> RegisterAsync(string login, string password)
{
    var hash = Hash(password);
    var exist = await _repository.Where<User>(s => s.Login == login && s.Password == hash).AnyAsync();

    if (exist)
        throw new AppError("User already exist");

    var id = Guid.NewGuid();
    var user = new User(id, login, hash);

    await _repository.SaveAsync(user);
    await _emailService.SendAsync(user.Login, "Добро пожаловать", "Бла бла бла");

    return user;
}
```

Рисунок 1.10 - C# в контексті веб - розробки

Для аналізу актуальності використання зазначених мов програмування в контексті виконання поставленої задачі, проведемо їх порівняльну характеристику за відповідними критеріями.

Таблиця 1.1 - Порівняльна характеристика за парадигмами

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Імперативна	+	+	+	+	+	+
Об'єктно-орієнтована	+	+	+	+	+	+
Функціональна	+/-	+/-	+/-	+/-	+	+
Рефлексивна	+/-	+/-	+	+	+	+
Узагальна	+	+	+	+	+	+
Логічна	-	-	-	-	-	-
Декларативна	+/-	-	+/-	+	+	+
Розподілена	+/-	+	-	+	+/-	+/-

Таблиця 1.2 - Порівняльна характеристика за типізацією

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Статична типізація	+	+	-	-	-	-
Динамічна типізація	+	-	+	+	+	+
Явна типізація	+/-	+	-	+/-	+/-	-
Неявна типізація	+	-	+	+	+	+
Неявна	-	+	+	+	+	+

Таблиця 1.3 - Характеристика за можливостями керування пам'яттю

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Створення об'єктів	+	-	-	-	-	-
Некеровані вказівники	+	-	-	-	-	-
Ручне керування	+	-	-	-	-	-
Збирання сміття	+	+	+	+	+	+

Таблиця 1.4 – Функціональні можливості

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Декларації чистоти	-	-	-	-	-	-
First class функції	+	-	+	+	+	+
Анонімні функції	+	+	+	+	+	+
Лексичні замкнення	+	+	+	+	+	+
Часткове застосування	-	-	+	-	+	+

Карування	+	-	+	+	+	+
-----------	---	---	---	---	---	---

Таблиця 1.5 - Порівняльна характеристика за об'єктно – орієнтованими можливостями

Можливість	C#	Java	JavaScript	PHP	Python	Ruby
Інтерфейси	+	+	+	+	+	-
Мультиметоди	+/-	-	-	-	-	-
Міксини	-	+	+	+	+	+
Перейменування	-	-	-	-	-	-
Множинне спад - ння	-	-	-	-	+	-

Для визначення поточної актуальності використання мов програмування в контексті веб - розробки, проведемо аналіз існуючих веб – орієнтованих інформаційних систем.

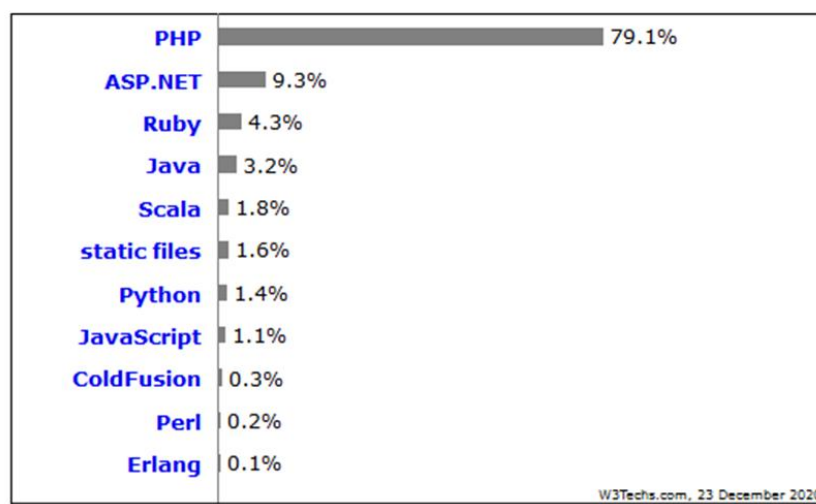


Рисунок 1.11 - Частота використання у існуючих інформаційних системах

З проведеного аналізу можна побачити, що лідерство у використанні в контексті веб - розробки впевнено утримує мова програмування PHP.

1.4 Аналіз існуючих СУБД для вирішення поставленої задачі

В процесі розробки будь-якої інформаційної системи, незалежно від використовуваних мов програмування чи конкретного типу проекту, бази даних виявляються невід'ємною частиною функціоналу. База даних – це організований набір інформації, який уможливорює зберігання та оптимізовану обробку даних.

Вона відіграє ключову роль у спрощенні управління інформацією та забезпеченні її ефективного використання.

Наприклад, у випадку онлайн-телефонного довідника база даних використовується для зберігання інформації про особи, телефонні номери та інші контактні дані. Також, компанії, які постачають електроенергію, використовують бази даних для ефективного управління процесами виставлення рахунків, вирішенням питань клієнтів та обробкою даних про можливі несправності чи проблеми.

У сучасному світі соціальних мереж, як Facebook, зберігання, обробка та подання величезної кількості даних, таких як інформація про користувачів, їх взаємодії, повідомлення та рекламні матеріали, неможлива без використання потужних баз даних. Ці приклади ілюструють важливість та різноманітність використання баз даних у різних сферах технологічного розвитку.

Отже, бази даних стають невід'ємною частиною кожного інформаційного проекту, відкриваючи можливості для систематизації, зберігання та ефективного управління великим обсягом даних. Наведені приклади вказують на важливість баз даних у різноманітних галузях, де вони використовуються для оптимізації різних процесів та забезпечення безперебійної роботи інформаційних систем. У світі, де обробка даних відіграє ключову роль, бази даних виступають невидимими, але важливими "архітекторами" успішної функціональності багатьох сучасних технологічних рішень



Рисунок 1.12 – Взаємодія компонентів інформаційної системи з базою даних

Розглянемо системи управління базами даних (СУБД), що користуються найбільшою популярністю, та проведемо аналіз, щоб визначити, яка з них оптимально підходить для вирішення нашого завдання.

Розглянемо MySQL, одну з найпопулярніших систем управління базами даних (СУБД), і докладніше розглянемо її характеристики та переваги.

MySQL є відкритою реляційною СУБД, яка визначається своєю надійністю, швидкістю та ефективністю. Вона широко використовується у веб-розробці, особливо у спільноті відкритого програмного забезпечення, завдяки своїй безкоштовності та доступності відкритого коду.

MySQL підтримує мову SQL для зручного управління та взаємодії з базою даних. Вона має гнучку архітектуру та може використовуватися на різних платформах, включаючи Linux, Windows та macOS.

Однією з ключових переваг MySQL є його висока продуктивність та швидке виконання запитів. Вона також підтримує транзакції для забезпечення консистентності та надійності даних.

MySQL є надійним вибором для великої кількості проектів, особливо там, де важлива швидкодія та масштабованість бази даних. З її допомогою можна реалізувати різноманітні завдання, від невеликих веб-сайтів до складних корпоративних систем.

Розглянемо MariaDB як ефективну систему управління базами даних (СУБД). MariaDB є відкритою реляційною СУБД, розробленою на основі відкритого коду MySQL після його придбання компанією Oracle. Однією з ключових переваг MariaDB є спроба зберегти сумісність з MySQL, забезпечуючи велику схожість між ними для полегшення процесу міграції проектів.

Система підтримує мову SQL і відрізняється високою продуктивністю та швидкістю виконання запитів. Як і її попередник, MariaDB працює на різних операційних системах, що робить її гнучкою для використання у різних середовищах.

Особливу увагу слід приділити підвищенню рівня безпеки та масштабованості MariaDB. Ці характеристики роблять її привабливим вибором для великих

корпоративних проектів та застосувань, де важливі аспекти безпеки та масштабованості баз даних.

Ця СУБД має активну спільноту користувачів та розробників, що сприяє постійному вдосконаленню та підтримці. Іншою важливою особливістю є те, що MariaDB надає ряд додаткових можливостей порівняно з MySQL, що може бути важливим фактором при виборі в залежності від конкретних потреб проекту.

Узагальнюючи, MariaDB є потужною та надійною системою управління базами даних, яка заслуговує уваги при розгляді варіантів для реалізації різних типів проектів.

Розглянемо Oracle, одну з провідних систем управління базами даних (СУБД) на сучасному ринку. Oracle Database, розроблена компанією Oracle Corporation, визначається своєю високою продуктивністю та широким набором функціональних можливостей.

Oracle Database є реляційною СУБД, яка використовує мову SQL для управління та взаємодії з базою даних. Однією з важливих особливостей Oracle є його підтримка для великих та високонавантажених систем, що робить його популярним в корпоративному середовищі.

Система Oracle володіє продуманою архітектурою, яка дозволяє оптимізувати роботу з великим обсягом даних та забезпечує високий рівень надійності. Вона також підтримує транзакції та має широкий спектр інтегрованих інструментів для адміністрування та моніторингу баз даних.

Oracle Database володіє розширеними можливостями у сферах аналітики, обробки даних у реальному часі та використання розподілених систем. Висока стійкість та підтримка корпоративних стандартів роблять Oracle затребуваною для великих та критично важливих систем.

Його комерційна природа та вартість можуть бути чинниками, які слід розглядати при виборі для проекту, проте Oracle Database залишається однією з найпотужніших та найрізноманітніших СУБД на ринку.

Розглянемо PostgreSQL, потужну та відкриту систему управління базами даних (СУБД), яка завоювала популярність завдяки своїм продуктивним можливостям та високій стабільності.

PostgreSQL є реляційною СУБД, яка активно розвивається в умовах відкритого програмного забезпечення. Вона визначається своєю дотриманістю стандартів SQL, що робить її відмінною вибором для проектів, де необхідно забезпечити стандартизований підхід до роботи з базою даних.

PostgreSQL підтримує широкий спектр функцій, включаючи розширені типи даних, індексацію повнотекстового пошуку та можливості геопросторового аналізу. Вона також відома своєю високою масштабованістю та здатністю працювати з великим обсягом даних та високими навантаженнями.

PostgreSQL використовує велику та активну спільноту розробників, що сприяє постійному вдосконаленню та підтримці. Зокрема, її модульна архітектура дозволяє додавати розширення та кастомізувати функціонал згідно з конкретними потребами проекту.

PostgreSQL володіє важливою перевагою в тому, що вона є вільною та відкритою для використання, що робить її привабливим вибором для різних типів проектів, включаючи великі корпоративні системи та стартапи.

У контексті вибору системи управління базами даних (СУБД) для реалізації інформаційної системи, важливо ретельно розглянути особливості кожної з розглянутих СУБД. Для виконання цього завдання проведемо їх порівняльну характеристику, наведену у таблиці 1.6.

Таблиця 1.6 – Порівняльна характеристика визначених СУБД

Назва	Переваги	Недоліки
MySQL	Велика спільнота користувачів, добра документація, висока продуктивність, підтримка	Обмежена підтримка складних операцій з даними, менша функціональність порівняно з деякими іншими СУБД.

	транзакцій, безкоштовність та відкритий код.	
Oracle	Висока продуктивність, розширені можливості, велика надійність, широкий функціонал для корпоративних застосувань.	Вартість ліцензій, великий обсяг системних ресурсів, більша складність налаштування.
MariaDB	Сумісність з MySQL, висока продуктивність, відкритий код, безкоштовність, активна спільнота.	Деяка нестабільність на ранніх версіях, менша кількість додаткових інструментів порівняно з MySQL.
PostgreSQL	Сумісність з SQL-стандартами, відкритий код, широкі можливості, велика кількість розширень, висока масштабованість.	Трохи менша продуктивність на великих обсягах даних порівняно з деякими конкурентами.

Проведемо аналіз усіх випадків використання СУБД та отримаємо наступний графік :

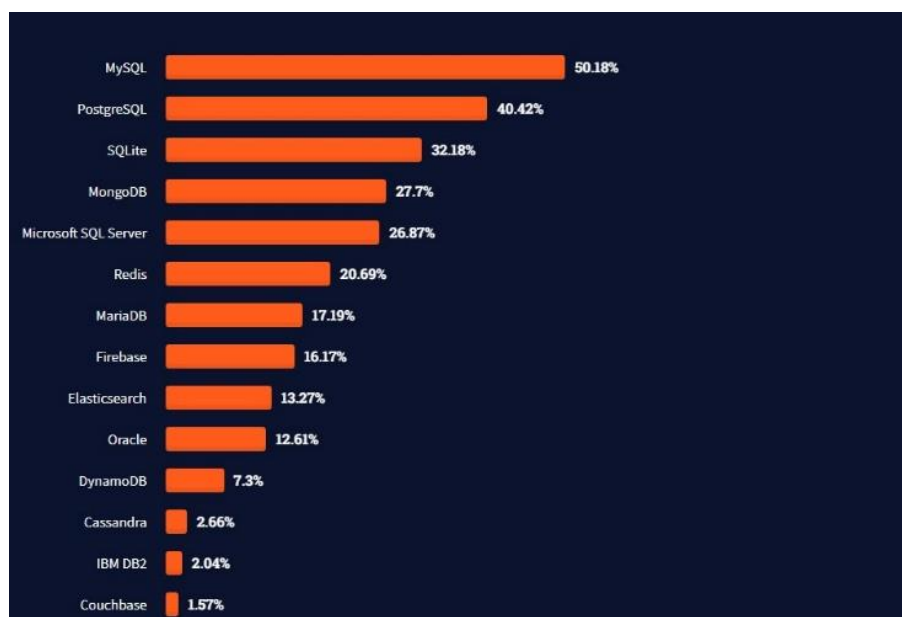


Рисунок 1.12 - Частота використання СУБД у веб - розробці

Для визначення поточної актуальності використання СУБД в контексті веб - розробки, проведемо аналіз існуючих веб – орієнтованих інформаційних систем. Як можна побачити, близько 50% поточних інформаційних систем використовують систему керування базами даних MySQL у своїй конструкції.

1.5 Вибір інструментів для вирішення поставленого завдання на основі проведеного аналізу

Провівши аналіз та порівняльну характеристику між актуальними для вирішення нашого завдання інструментами, маємо можливість зробити висновок, які з них використовуватимемо у процесі виконання роботи.

Для написання візуальної частини інтерфейсу програми групового цифрового підписування смарт контрактів буде використано HTML як основну розмітку для сторінок, які відображатимуться перед користувачем та CSS для стилізації цих сторінок та надання їм необхідного нам вигляду. В якості мови для виконання бекнд частини буде використаний PHP, проведений аналіз показав, що ця мова найбільш підходить для виконання поставленого завдання, так як вона має досить зрозумілу структуру, є абсолютно вільно використовуваною, має хороші показники в області продуктивності і відмінно підходить для написання веб - додатків, що недвозначно підтверджується частотою її застосування. В якості СУБД була обрана

MySQL, так як вона достатньо проста у використанні, має великий функціонал, безпечна, має хорошу швидкість роботи, а також має інтерфейс для взаємодії з PHP. Як IDEA для розробки було вибрано PHPStorm, а для візуалізації роботи з базою даних PHPMuAdmin.

1.6 Висновок до розділу 1

У цьому розділі проведено всебічний огляд та аналіз існуючих підходів до реалізації групового цифрового підписування смарт контрактів. Основні висновки, зроблені на основі дослідження, можна підсумувати наступним чином:

1. Характеристика об'єкту дослідження: Інформаційна система групового цифрового підписування смарт контрактів була детально розглянута, що дозволило визначити основні вимоги та функціональні можливості для забезпечення надійності та безпеки підписів.

2. Аналіз існуючих рішень: Розглянуто кілька сучасних підходів до вирішення задачі групового цифрового підписування.

3. Аналіз систем управління базами даних (СУБД): Проведено порівняльний аналіз кількох популярних СУБД

4. Вибір інструментів: На основі проведеного аналізу було вибрано оптимальні інструменти для реалізації поставленої задачі.

Загалом, результати даного розділу підтверджують, що сучасні технології дозволяють ефективно вирішити завдання групового цифрового підписування смарт контрактів. Виявлені підходи та інструменти забезпечують необхідний рівень безпеки, прозорості та достовірності, що є важливим для широкого впровадження та використання цих технологій у різних сферах. Розроблена система має потенціал для успішної інтеграції у практичну діяльність, сприяючи підвищенню ефективності та надійності угод, що укладаються за допомогою смарт контрактів.

2. ПРОЕКТУВАННЯ ЦИФРОВОГО ПІДПISУВАННЯ СМАРТ КОНТРАКТІВ

Розвиток цифрових технологій і впровадження блокчейн-рішень значно змінюють підходи до управління даними та взаємодії між сторонами в різних сферах діяльності. Однією з ключових задач при розробці нових інформаційних систем є визначення їх архітектури, що забезпечить ефективну, надійну та безпечну роботу. Це особливо актуально для систем, що використовують смарт контракти, де правильне проектування архітектури та алгоритмів є критичним для забезпечення їх функціональності.

У цьому розділі буде детально розглянуто наступні аспекти розробки інформаційної системи:

1. Розробка архітектури проекрованої системи – визначення основних концепцій, процесів та вимог, які є основою для розробки системи, побудова архітектури, яка включає всі необхідні компоненти та їх взаємодію для забезпечення безперебійної роботи системи.

2. Розробка запропонованих алгоритмів роботи системи – розробка алгоритмів, що визначають порядок та логіку виконання основних функцій системи.

3. Проектування бази даних – створення схеми бази даних, яка буде ефективно зберігати та управляти даними, необхідними для роботи системи.

Метою цього розділу є формування детального та всебічного уявлення про проектовану інформаційну систему, починаючи від опису її предметної області до проектування конкретних компонентів і алгоритмів. Це дозволить забезпечити всебічне розуміння та підготувати основу для подальшої реалізації системи, враховуючи всі необхідні технічні та функціональні вимоги.

2.1. Розробка архітектури проекрованої системи

Предметною областю даної дипломної є процеси автоматизації підписування смарт контрактів між віддаленими групами користувачів.

Для того, щоб повністю представити функціонал проекрованої інформаційної системи, виділено такі основні процеси:

- Реєстрація користувача.
- Авторизація користувача у особистому кабінеті.
- Можливість генерації ключів, необхідних для цифрового підпису кожного користувача.
- Виведення поточного списку смарт – контрактів на інтерфейс користувача.
- Можливість створення нового смарт - контракту.
- Можливість додавання учасників до смарт - контракту.
- Можливість автоматизованої перевірки підписів.
- Можливість активації контракту після отримання достатньої кількості дійсних підписів.

Можливість групового підписування смарт - контракту надає користувачеві можливість авторизуватися у додатку, створити необхідний йому смарт - контракт, додати учасників смарт контракту, додати цільову функцію активації контракту, отримати ключі підпису смарт – контракту та після отримання достатньої кількості дійсних підписів отримати автоматизоване виконання активації контракту.

Основні етапи процесу:

- Авторизація у додатку
- Створення смарт - контракту
- Додавання учасників
- Визначення функції активації смарт - контракту
- Отримання ключів
- Підпис контракту іншими учасниками
- Виконання смарт - контракту

Характеристика процесу групового підписування смарт - контракту наведена у таблиці 2.1.

Таблиця 2.1 - Характеристика процесу групового підписування смарт - контракту

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Групового підписування смарт - контракту
Основні учасники	Користувачі, Система, База даних
Вхідна подія	Запит на створення смарт - контракту
Вихідна подія	Виконання функції активації смарт - контракту
Вихідні документи	Виконаний смарт - контракт
Клієнт бізнес процесу	Користувач

Для реалізації проекрованої інформаційної системи групового цифрового підписування смарт контрактів було вирішено використовувати архітектуру клієнт – серверного застосунку. Архітектура клієнт-сервер є досить поширеною та ефективною моделлю для розробки інформаційних систем, включаючи системи веб – додатків підписування контрактів. Наведемо кілька обґрунтувань використання цієї архітектури у розробці інформаційної системи групового цифрового підписування смарт контрактів:

- Розподілена архітектура: Клієнт-серверна модель дозволяє розподілити функціональність системи між клієнтськими та серверними компонентами. У випадку смарт контрактів, клієнтські компоненти можуть бути реалізовані у вигляді веб-інтерфейсу а серверні компоненти – у вигляді бази даних. Це дозволяє покращити масштабованість та надійність системи, розподіляючи навантаження та забезпечуючи більш ефективне управління ресурсами.

- Централізоване управління даними: Використання сервера дозволяє централізовано зберігати та управляти даними смарт контрактів. Інформація щодо контрактів, користувачів, що їх підписують та інші аспекти процесу підписування смарт контрактів можуть бути збережені у базі даних на сервері,

що забезпечує консистентність та доступність даних для різних користувачів. Це спрощує управління та обробку даних, а також забезпечує безпеку та захист інформації.

- **Забезпечення доступу до даних:** Клієнт-серверна архітектура дозволяє різним користувачам отримувати доступ до даних одночасно.

- **Зменшення обробки на клієнтському пристрої:** Використання архітектури клієнт-сервер дозволяє зменшити навантаження на клієнтські пристрої, так як більшість обчислень та обробки даних відбуваються на сервері. Клієнтські пристрої можуть мати обмежені обчислювальні та мережеві ресурси, особливо у випадку мобільних пристроїв. Використання сервера для обробки складних операцій, проведення моделювання та інших завдань допомагає забезпечити більш швидку та ефективну роботу клієнтських додатків.

- **Забезпечення безпеки:** Клієнт-серверна архітектура може допомогти забезпечити безпеку в інформаційній системі групового цифрового підписування смарт контрактів. Сервер може бути налаштований для автентифікації та авторизації користувачів, контролю доступу до різних функцій та даних системи. Крім того, централізоване збереження даних на сервері дозволяє реалізувати резервне копіювання та захист інформації від втрати або пошкодження.

Принцип роботи обраної архітектури наведено на рисунку 2.1.

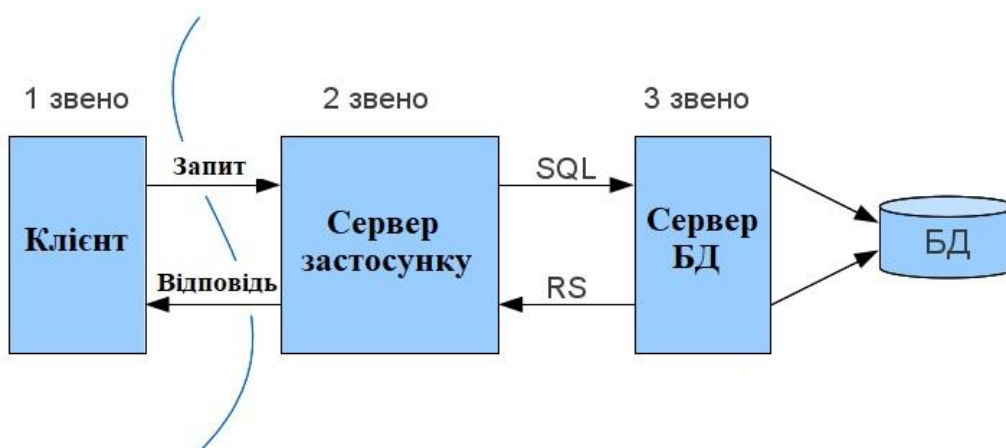


Рисунок 2.1 – Принцип роботи архітектури клієнт – сервер

2.2 Розробка запропонованих алгоритмів роботи системи.

В сучасному цифровому світі реєстрація та авторизація користувачів є невід'ємною частиною багатьох веб- та мобільних додатків. Ці процеси забезпечують ідентифікацію користувачів, захист персональних даних та надають доступ до персоналізованих сервісів. Розглянемо, як саме відбувається реєстрація нових користувачів та авторизація вже зареєстрованих, які кроки включає кожен процес, а також які механізми безпеки впроваджуються для захисту даних.

Перш за все розберемось як покроково проходить реєстрація користувача.

Крок 1 – Початок

Користувач розпочинає процес реєстрації, відкривши відповідну сторінку з формою реєстрації.

Крок 2 – Введення даних для реєстрації.

Користувач вводить свої дані у відповідні поля форми (логін, ім'я, прізвище, пароль, країна).

Крок 3 – Надсилання запиту на сервер.

Після заповнення всіх обов'язкових полів для реєстрації користувач натискає кнопку "Зареєструватися", відправляючи дані на сервер.

Крок 4 – Отримання запиту.

Сервер отримує запит на реєстрацію і починає обробку даних.

Крок 5 – Перевірка чи логін дозволений для реєстрації.

Сервер перевіряє, чи введений користувачем логін є унікальним та дозволеним для реєстрації.

Крок 6 – Логін дозволений?

Крок 6.1 – Ні. Надсилання помилки про заборону реєстрації.

Якщо логін зайнятий, користувачу надсилається повідомлення про помилку.

Крок 6.2 – Так.

Якщо логін дозволений, виконується наступний крок.

Крок 8 – Генерація секретного ключа.

Сервер генерує секретний ключ для цифрового підпису контрактів.

Крок 9 – Створення нового користувача в базі даних.

Сервер створює нового користувача у базі даних.

Крок 10 – Надсилання відповіді клієнту.

Сервер надсилає відповідь клієнту, що реєстрація успішна, разом із секретним ключем.

Крок 11 – Кінець.

Процес реєстрації завершено. Користувач може продовжити роботу в системі.

Розробимо блок-схему алгоритму реєстрації (рис.2.2)

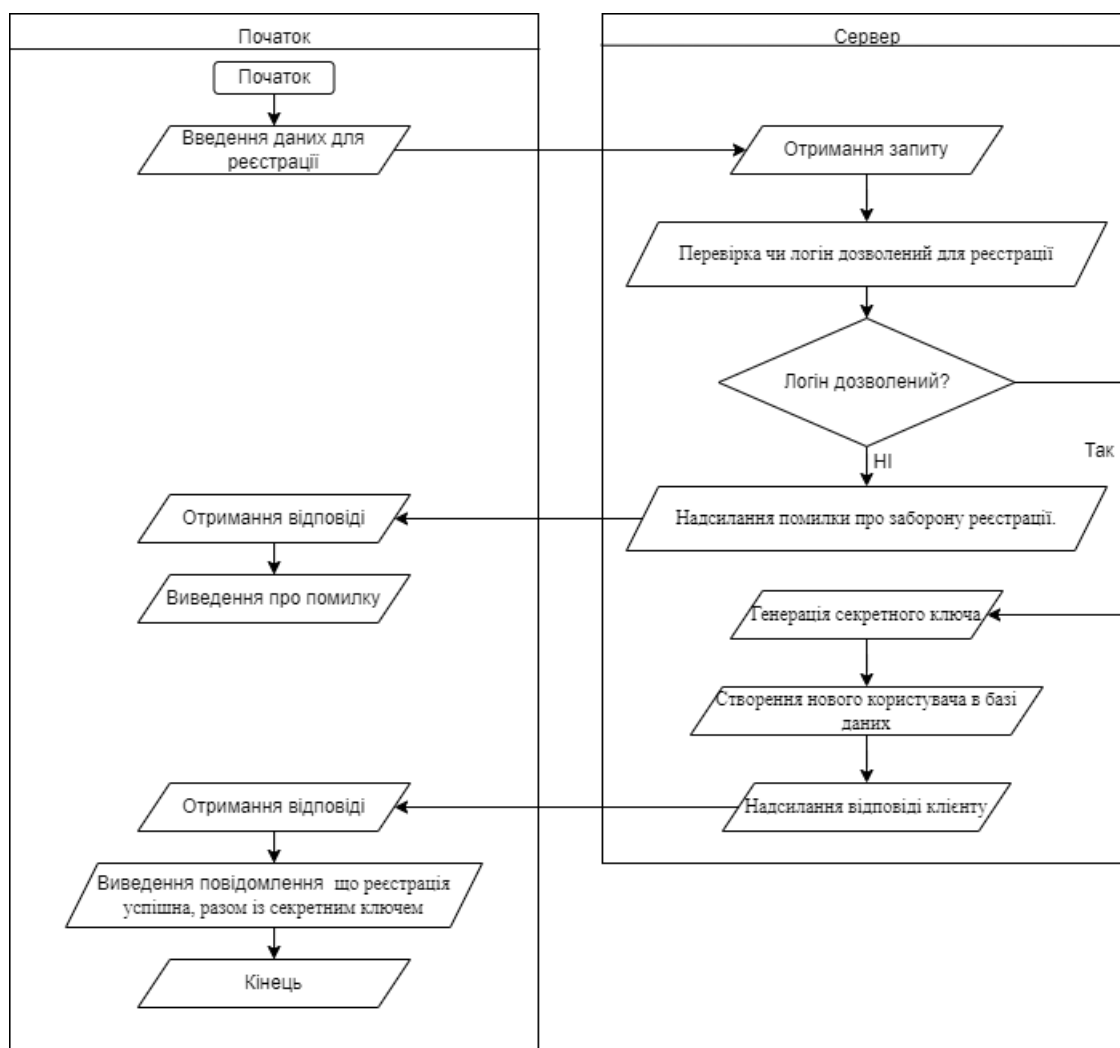


Рисунок 2.2 - блок-схема алгоритму реєстрації

Після чого розберемо покроково, як проходить авторизація користувача.

Крок 1 – Початок.

Користувач розпочинає процес авторизації, відкривши відповідну сторінку з формою авторизації.

Крок 2 – Введення даних для авторизації.

Користувач вводить свої дані у відповідні поля форми (логін та пароль).

Крок 3 – Надсилання запиту на сервер.

Після заповнення всіх обов'язкових полів користувач натискає кнопку "Увійти", відправляючи дані на сервер.

Крок 4 – Отримання запиту.

Сервер отримує запит на авторизацію і починає обробку даних.

Крок 5 – Перевірка наявності користувача.

Сервер перевіряє, чи існує користувач із введеним логіном.

Крок 6 – Користувач існує?

Крок 6.1 – Ні. Надсилання повідомлення про помилку.

Якщо користувача з введеним логіном не існує, сервер надсилає повідомлення про помилку.

Крок 6.2 – Так.

Якщо користувач існує, виконується наступний крок.

Крок 7 – Перевірка правильності паролю.

Сервер перевіряє, чи правильний пароль введено користувачем.

Крок 8 – Пароль правильний?

Крок 8.1 – Ні. Надсилання повідомлення про помилку.

Якщо пароль неправильний, сервер надсилає повідомлення про помилку.

Крок 8.2 – Так.

Якщо пароль правильний, виконується наступний крок.

Крок 9 – Створення сесії для користувача.

Сервер створює сесію для користувача і перенаправляє його на головну сторінку.

Крок 10 – Отримання відповіді.

Клієнт отримує відповідь і перенаправляється на головну сторінку.

Розробимо блок-схему алгоритму авторизації (рис.2.3)

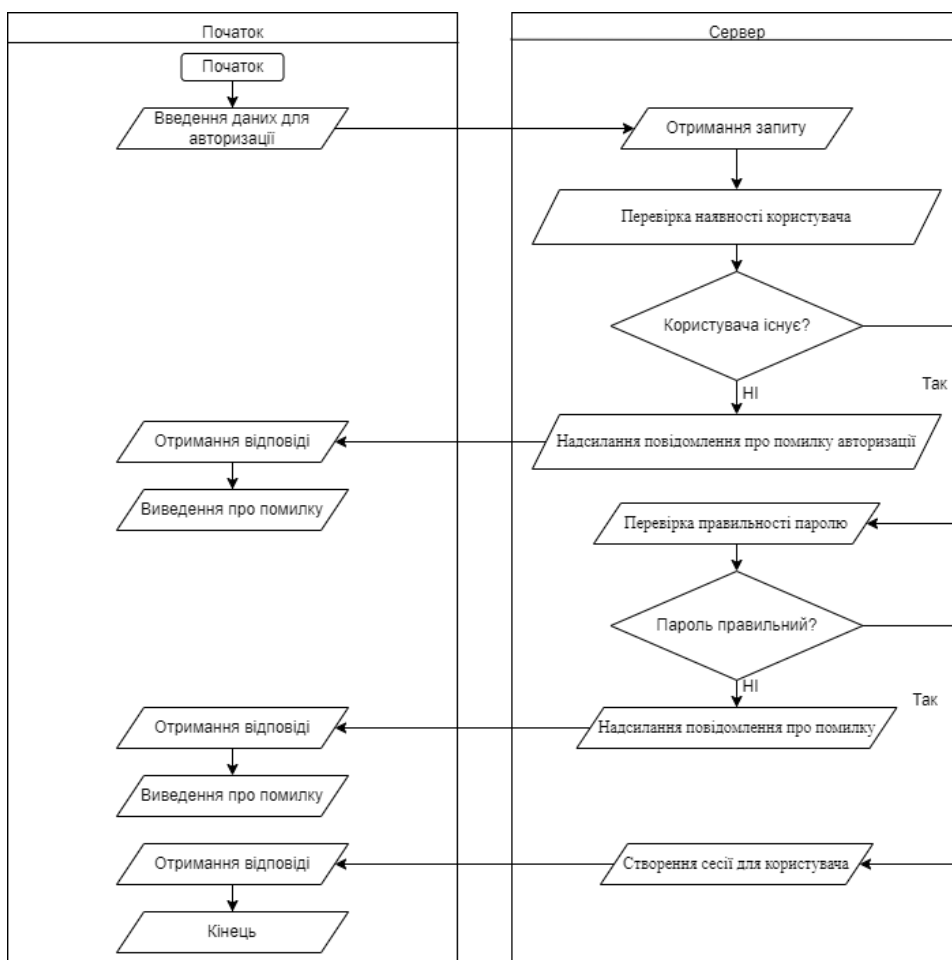


Рисунок 2.3 - блок-схема алгоритму авторизації

Розглянемо, як буде відбуватись генерація секретного ключа для підписування смарт-контрактів.

```
$random_key = substr(md5(uniqid(mt_rand(), true)), 0, 20);
```

Розберемо цей рядок коду покроково:

Крок - 1.

`mt_rand()`:

Функція `mt_rand()` генерує випадкове число. Це покращена версія функції `rand()`, яка забезпечує кращу продуктивність та якість випадкових чисел.

Крок 2.

`uniqid(mt_rand(), true)`:

Крок - 2.1 Функція `uniqid()` генерує унікальний ідентифікатор на основі поточного часу в мікросекундах.

Крок - 2.2. Передача `mt_rand()` як аргументу забезпечує ще більшу унікальність.

Крок - 2.3 Додавання true як другого аргументу змушує uniqid() генерувати більш довгий і унікальний рядок.

Крок – 3

```
md5(uniqid(mt_rand(), true));
```

Функція md5() обчислює хеш MD5 для рядка, що генерується uniqid(). Результат — це 32-символьний шістнадцятковий рядок.

Крок – 4

```
substr(md5(uniqid(mt_rand(), true)), 0, 20);
```

Функція substr() витягує підрядок з хешу MD5. У цьому випадку вона бере перші 20 символів.

Отже, рядок коду генерує випадковий 20-символьний ключ, який є досить унікальним і важковідгаданим.

Розглянемо процес створення нового смарт-контракту. Це включає кроки від створення контракту до його розгортання на блокчейн-платформі, забезпечуючи автоматичне виконання умов угоди.

Крок - 1 Початок.

Користувач розпочинає процес створення нового смарт-контракту, відкривши сторінку додавання нового контракту.

Крок - 2 Введення даних для контракту.

Користувач вводить необхідні дані для створення контракту, такі як назва, опис, посилання для активації, функція активації та обирає учасників контракту зі списку.

Крок - 3 Надсилання запиту на сервер.

Після заповнення форми користувач натискає кнопку "Створити контракт", і дані форми надсилаються на сервер для обробки.

Крок - 4 Отримання запиту.

Сервер отримує запит на створення нового контракту.

Крок - 5 Перевірка користувача.

Система перевіряє, чи користувач автентифікований. Якщо ні, він буде перенаправлений на сторінку авторизації.

Крок - 6 Виконання SQL-запиту для вставки даних.

Введені дані записуються до бази даних.

Крок - 7 Підтвердження створення контракту.

Якщо запит успішно виконано, користувач отримує повідомлення про успішне створення контракту.

Крок - 8 Перенаправлення на головну сторінку.

Після створення контракту користувач автоматично перенаправляється на головну сторінку.

Розробимо блок-схему створення нового смарт-контракту (рис.2.4)

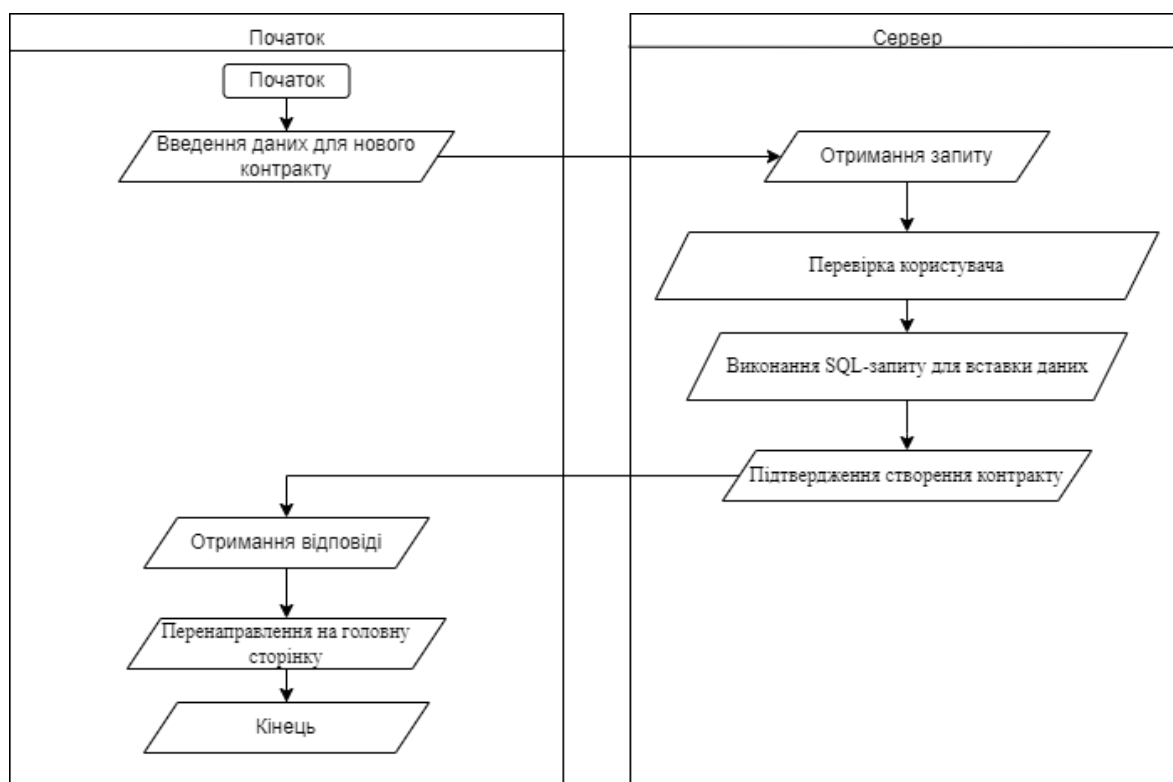


Рисунок 2.4 - блок-схема створення нового смарт-контракту

Розглянемо, як відбувається підписування смарт-контракту, та яка інформація відображається в контракті.

Крок 1 – Початок.

Користувач перевіряється на наявність сесії, інакше його перенаправляється на сторінку авторизації.

Крок 2 – Перевірка з'єднання.

Встановлюється з'єднання з базою даних. У разі невдалого з'єднання виводиться повідомлення про помилку.

Крок 3 – Перевірка, чи була відправлена форма.

Після відправлення форми POST отримується ідентифікатор контракту та цифровий ключ.

Крок 4 – Отримання логіну користувача з сесії.

Отримується логін користувача, який знаходиться в сесії.

Крок 5 – Отримання значення цифрового підпису користувача з бази даних.

Вибирається підпис користувача з бази даних за його логіном.

Крок 6 – Перевірка, чи ключі співпадають.

Порівнюється введений користувачем цифровий ключ з тим, що зберігається в базі даних.

Крок 7 – Зміна статусу підпису в базі даних.

Якщо ключі співпадають, статус підпису змінюється на "підписано" для даного користувача в базі даних контрактів.

Крок 8 – Перевірка, чи всі користувачі підписали контракт.

Перевіряється, чи всі користувачі, які пов'язані з контрактом, мають статус "підписано".

Крок 9 – Активація контракту.

Якщо всі користувачі підписали контракт, то статус контракту змінюється на "активовано". Користувач перенаправляється на посилання контракту.

Крок 10 – Закінчення.

Сесія закривається, з'єднання з базою даних закривається.

Розробимо блок-схему підпису наявного смарт-контракту (рис.2.5)



Рисунок 2.5 – блок-схема підпису наявного смарт-контракту

2.4 Проектування бази даних.

Для того, щоб зберігати інформацію щодо смарт - контрактів, користувачів тощо було реалізовано базу даних. Бази даних використовуються у інформаційних системах для зберігання, керування та організації великих обсягів даних. Наведемо основні причини використання баз даних у інформаційних системах:

- Зберігання даних: База даних є центральним місцем для зберігання даних, таких як інформація про користувачів, продукти, транзакції тощо. Вона дозволяє структуровано зберігати дані в таблицях, рядках та стовпцях, що спрощує організацію та управління інформацією.

- Керування даними: База даних надає механізми для керування даними, такими як додавання, оновлення та видалення записів. Це дозволяє змінювати та оновлювати дані в системі, забезпечуючи актуальність та цілісність інформації.
- Запити та аналітика: Бази даних дозволяють виконувати складні запити та аналізувати дані для отримання необхідної інформації. Запити можуть бути використані для вибору конкретних даних, сортування, фільтрації, обчислення агрегатних функцій та здійснення інших операцій для виявлення корисної інформації.
- Цілісність даних: Бази даних забезпечують механізми для забезпечення цілісності даних. Це означає, що дані в базі даних повинні задовольняти визначені правила та обмеження. Наприклад, можна встановити обмеження на типи даних, виконувати перевірки на унікальність значень, наявність обов'язкових полів тощо. Це допомагає забезпечити консистентність та точність

Структуру бази даних проектованої системи групового цифрового підписування смарт контрактів наведено у таблиці 2.2

Таблиця 2.2 – Структура реалізованої бази даних

Об'єкт	Власність	Тип	Ідентифікатор
Таблиця смарт - контрактів	Ідентифікатор	Integer	Id
	Зображення	Varchar	Img
	Назва	Varchar	Title
	Опис	Varchar	Descr
	Посилання на активацію	Varchar	Link
	Функція активації	Varchar	Activation
	Власник контракту	Varchar	Creator

	Користувачі, що приймають участь у підписанні контракту	Varchar	Users
	Статус контракту	Varchar	Status
Таблиця відгуків	Ідентифікатор	Integer	msgID
	Ім'я відправника	Varchar	senderfname
	Прізвище відправника	Varchar	senderlname
	Email відправника	Varchar	sendereMail
	Відгук	Varchar	Senderfeedback
	Email	Varchar	Email
Таблиця користувачів	Ідентифікатор	Integer	Id
	Логін	Varchar	Login
	Ім'я	Varchar	Name
	Прізвище	Varchar	Lname
	Пароль	Varchar	Password
	Країна	Varchar	Country
	Цифровий код	Varchar	Sign

Для забезпечення інформаційної безпеки особистих даних користувачів, ці дані буде зашифровано за допомогою алгоритмів симетричного шифрування. Симетричне шифрування передбачає використання одного й того самого ключа і для зашифрування, і для розшифрування. До симетричних алгоритмів застосовуються дві основні вимоги: повна втрата всіх статистичних закономірностей в об'єкті шифрування та відсутність лінійності. Прийнято розділяти симетричні системи на блокові та потокові. У блокових системах відбувається розбиття вихідних даних на блоки з подальшим перетворенням ключем. У поточних системах виробляється певна послідовність (вихідна гамма), яка у подальшому накладається саме повідомлення, і шифрування даних відбувається потоком у міру генерування гами. Схема зв'язку з використанням

симетричної криптосистеми представлена малюнку. Де M - відкритий текст, K - секретний ключ, що передається по закритому каналу, $E_p(M)$ - операція зашифрування, а $D_k(M)$ - операція розшифрування.

Операція підстановки виконує першу вимогу до симетричного шифру, позбавляючись будь-яких статистичних даних шляхом перемішування бітів повідомлення за певним заданим законом. Перестановка необхідна до виконання другої вимоги – надання алгоритму нелінійності.

Симетричні системи мають свої переваги, так і недоліки перед асиметричними. До переваг симетричних шифрів відносять високу швидкість шифрування, меншу необхідну довжину ключа за аналогічної стійкості, велику вивченість і простоту реалізації. Недоліками симетричних алгоритмів вважають насамперед складність обміну ключами через велику ймовірність порушення секретності ключа при обміні.

2.5 Висновок до розділу 2

У цьому розділі було проведено всебічний аналіз та розробку інформаційної системи, орієнтованої на впровадження смарт-контрактів. У результаті роботи було розглянуто та розроблено ключові компоненти системи, що є критично важливими для забезпечення її ефективної, надійної та безпечної функціональності.

Перш за все, було визначено архітектуру проекрованої системи. Цей процес включав аналіз основних концепцій, процесів та вимог, які є необхідними для створення стабільної та ефективної системи. Особливу увагу було приділено інтеграції всіх необхідних компонентів та забезпеченню їх коректної взаємодії, що дозволяє досягти безперебійної роботи системи.

Наступним етапом було розроблено алгоритми роботи системи, які визначають порядок та логіку виконання основних функцій. Ці алгоритми забезпечують точне виконання смарт-контрактів та інших важливих функцій системи, що є основою для її надійної роботи.

Окрім цього, було проведено проектування бази даних, яка є ключовим елементом для зберігання та управління даними, необхідними для функціонування

системи. Створена схема бази даних забезпечує ефективне зберігання та швидкий доступ до необхідної інформації, що є критично важливим для підтримки продуктивності та надійності системи.

Таким чином, результати цього розділу формують детальне та всебічне уявлення про проєктовану інформаційну систему. Виконані роботи забезпечують надійну основу для її подальшої реалізації, враховуючи всі необхідні технічні та функціональні вимоги. Це дозволяє впевнено рухатися до наступних етапів розробки, що включають безпосереднє впровадження та тестування системи у реальних умовах.

3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПРОЕКТОВАНОГО ПРОГРАМНОГО ПРОДУКТУ

У процесі розробки інформаційних систем важливим аспектом є створення зручних та функціональних інтерфейсів користувача, а також реалізація програмної логіки, яка забезпечує коректну та безперебійну роботу системи. Особливо це актуально для веб-додатків, що використовують смарт контракти, де необхідно враховувати як інтерактивність інтерфейсу, так і складність внутрішніх процесів. У цьому розділі розглядається програмна реалізація ключових компонентів веб-додатку для роботи зі смарт контрактами.

Метою даного розділу є опис та аналіз призначення програмних модулів системи, а також детальний розгляд процесу їх розробки. Основні аспекти, що будуть висвітлені в цьому розділі, включають:

1. Призначення програмних модулів – визначення функцій та ролей кожного програмного модуля в системі.
2. Програмна реалізація header і footer для веб-сторінок – створення універсальних компонентів для верхнього та нижнього колонтитулів, що забезпечують навігацію та загальний стиль сайту.
3. Програмна реалізація інтерфейсу головної сторінки – розробка головної сторінки веб-додатку, що забезпечує доступ до основних функцій системи.
4. Програмна реалізація інтерфейсу взаємодії зі смарт контрактом – створення інтерфейсу для перегляду, підписання та управління смарт контрактами.
5. Програмна реалізація логіки зміни статусу смарт контракту – розробка алгоритмів, що дозволяють змінювати статус смарт контрактів залежно від дій користувачів.
6. Програмна реалізація інтерфейсу створення нового смарт контракту – створення зручного інтерфейсу для генерування нових смарт контрактів.
7. Програмна реалізація інтерфейсу написання відгуків щодо використання системи – розробка форми для залишення зворотного зв'язку користувачами.

8. Програмна реалізація логіки реєстрації у системі – реалізація процесу реєстрації нових користувачів у системі.

9. Програмна реалізація сторінки авторизації – створення сторінки для входу зареєстрованих користувачів у систему.

Цей розділ забезпечує повне уявлення про те, як різні компоненти веб-додатку взаємодіють між собою і з користувачем, а також описує методи та інструменти, які використовуються для їх реалізації. Завдяки цьому буде сформовано чітке бачення структури та функціональності системи, що дозволить ефективно виконати її розробку та впровадження.

3.1 Призначення програмних модулів

Проектована інформаційна система складається з набору програмних модулів, кожний з яких реалізує логіку тих чи інших процесів. У таблиці 2.6 наведено призначення програмних модулів.

Таблиця 3.1 - Призначення програмних модулів

№	Позначення	Призначення
1.	Index	Модуль головної сторінки інформаційної системи, відповідає за навігацію по системі, відображення інформації щодо поточних смарт - контрактів, що наразі актуальні у базі даних системи та інформаційного блоку щодо використання системи.
2	Addcontract	Модуль для реалізації можливості створення нового смарт - контракту. Реалізує інтерфейс користувача та логіку взаємодії з базою даних.
3	Contract	Модуль для реалізації логіки відображення інформації щодо обраного смарт – контракту та можливості взаємодії з ним.
4	Modify_signature	Модуль для реалізації логіки підпису смарт контракту за допомогою цифрового коду.

5	Scripts	Модуль скриптів для реалізації роботи динамічної частини інтерфейсу користувача.
6	Contact-us	Модуль для реалізації логіки залишення користувачем відгуку щодо використання послуг системи.
7	Auth	Модуль для реалізації сторінки та логіки авторизації користувача у системі. Авторизація обов'язкова для використання системи.
8	Registration	Модуль для реалізації сторінки та логіки реєстрації користувача у системі. Реєстрація обов'язкова для використання системи.
9	Logout	Модуль для реалізації логіки завершення сесії користувача.

3.2 Програмна реалізація header и footer для веб-сторінок

Header - це блок у верхній частині сторінки сайту, який видно на всіх сторінках сайту.

Footer - блок у нижній частині сторінки. Містить корисну, але не першорядну інформацію.

Header і Footer використовуються для того, щоб не писати по багато разів один і той же повторюваний на багатьох веб сторінках код. Ця практика неймовірно зручно і значно економить час, а також робить код набагато читальнішим.

Для створюваного додатку було вирішено помістити в header горизонтальне меню для навігації сторінками додатку. Програмна реалізація наведена далі:

```
<div class="navbar-brand">
  <a href="index.php">
    <h1 class="navbar-heading">Smart Contracts</h1>
  </a>
</div>
<div class="navbar-container">
  <nav class="navbar">
    <ul class="navbar-menu">
      <li><a href="index.php">Головна</a></li>
      <li><a href="addcontract.php">Створення контракту</a></li>
      <li><a href="contact-us.php">Відгук</a></li>
      <li><a href="logout.php">Вийти</a></li>
    </ul>
  </nav>
</div>
```

```

    </ul>
  </nav>
</div>

```

У footer було вирішено помістити посилання на соціальні мережі, а також контактну інформацію про інформаційну систему та загальну інформацію про неї.

Програмна реалізація наведена далі:

```

<div class="footer-section footer-section1">
  <h2><i class="fas fa-user-alt"></i> Соціальні Мережі</h2>
  <div class="footer-section-inner-container">
    <span><a href="https://ru-ru.facebook.com/elonreevesmusk"><i class="fab fa-lg fa-facebook-square"></i>
Facebook</a></span>
    <span><a href="https://twitter.com/elonmusk?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Eauthor"><i
class="fab fa-lg fa-twitter-square"></i> Twitter</a></span>
    <span><a href="https://www.instagram.com/tina_baeva_"><i> Instagram</a></span>
  </div>
</div>

<div class="footer-section footer-section2">
  <h2><i class="fas fa-map-marked"></i> Зв'яжіться з нами </h2>
  <div class="footer-section-inner-container">
    <p>Зв'яжіться з нами , щоб зробити наш сервіс кращим, будь-який відгук буде враховано.</p>
    <p>Телефон: +38 099 999 99 99</p>
    <p>E-mail:SmartContracts@gmail.com</p>
  </div>
</div>

<div class="footer-section footer-section3">
  <p>© 2024 Smart Contracts. Created by Myself. </p>
  <a href="logout.php">Вийти</a>
  <a>Advertising</a>
  <a>Privacy Policy</a>
  <a>Contact </a>
</div>

```

3.3 Програмна реалізація інтерфейсу головної сторінки

На головній сторінці були розміщені блоки з смарт - контрактами, які на даний момент актуальні для поточного користувача, та з якими він може взаємодіяти. Дані щодо них знаходяться у базі даних та підвантажуються на сторінку за запитом користувача. Також на сторінці представлено меню з описом переваг взаємодії з додатком та відеоогляди переваг смарт – контрактів над аналогами.

```

<?php
session_start();
if($_SESSION['itlogin'] == ""){
    header('Location: /auth.php');
}else{
    $login = $_SESSION['itlogin'];
}
?>

```

У наведеній частині коду реалізовано ініціювання перевірки сесії користувача. Якщо користувача авторизована та токен його сесії існує – на інтерфейс буде відображено головну сторінку додатку. В зворотньому випадку користувача буде перенаправлено на сторінку авторизації.

```
<?php
$link = mysqli_connect("localhost", "root", "", "contracts");
$sql = "SELECT * FROM contracts";
?>
<header></header>
<div id="home-section-1" class="movie-show-container">
    <h1>Ваші поточні смарт - контракти</h1>
    <h3>Будьласка, оберіть смарт - контракт для перегляду інформації щодо нього</h3>

    <div class="movies-container">
        <?php
            if($result = mysqli_query($link, $sql)){
                $lines = mysqli_num_rows($result);
                if(mysqli_num_rows($result) > 0){
                    for ($i = 0; $i <= $lines - 1; $i++){
                        $row = mysqli_fetch_array($result);
                        echo '<div class="movie-box">';
                        echo '';
                        echo '<div class="movie-info ">';
                        echo '<h3>'. $row['title'] . '</h3>';
                        echo '<a href="contract.php?id='.$row['id'].'"><i class="fas fa-ticket-alt"></i> Перейти </a>';
                        echo '</div>';
                        echo '</div>';
                    }
                    mysqli_free_result($result);
                } else{
                    echo '<h4 class="no-annot">На даний момент немає актуальних смарт - контрактів</h4>';
                }
            } else{
                echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
            }

            // Close connection
            mysqli_close($link);
        ?>
    </div>
</div>
```

У наведеній частині коду реалізується підключення до бази даних за допомогою SQL запиту та побудова списку смарт - контрактів, які на даний момент актуальні у базі даних для поточного користувача. При натисканні на обраний смарт - контракт користувача буде перенесено на сторінку взаємодії з ним.

```
<div id="home-section-2" class="services-section">
    <h1>Чому ми ?</h1>
    <h3>Декілька плюсів використання системи !</h3>

    <div class="services-container">
        <div class="service-item">
```

```

<div class="service-item-icon">
  <i class="fas fa-4x fa-video"></i>
</div>
<h2>1. Автоматизація</h2>
<p>Смарт-контракти дозволяють автоматизувати виконання угодних умов без потреби посередників або
третіх сторін</p>
</div>
<div class="service-item">
  <div class="service-item-icon">
    <i class="fas fa-4x fa-credit-card"></i>
  </div>
  <h2>2. Безкоштовність</h2>
  <p>Використання нашого сервісу повністю безкоштовне !</p>
</div>
<div class="service-item">
  <div class="service-item-icon">
    <i class="fas fa-4x fa-theater-masks"></i>
  </div>
  <h2>3. Час та витрати</h2>
  <p>Використання смарт-контрактів може значно скоротити час та витрати, пов'язані з укладенням та
виконанням угод</p>
</div>
<div class="service-item"></div>
</div>
</div>

```

У наведеній частині коду реалізується меню з описом переваг процесу взаємодії з системою. Переваги представлені у вигляді динамічних іконок, що підсвічуються при наведенні.

```

<div id="home-section-3" class="trailers-section">
  <h1 class="section-title"> Не знаєте які переваги надасть вам використання смарт - контрактів ?</h1>
  <h3>Будьласка, ознайомтеся з наведеними описами переваг використання !</h3>
  <div class="trailers-grid">
    <div class="trailers-grid-item">
      
      <div class="trailer-item-info" data-video="ZE2HxTmxfrI?si=N-iXuaoCp3dQCK--">
        <h3>Smart Contracts</h3>
        <i class="far fa-3x fa-play-circle"></i>
      </div>
    </div>
    <div class="trailers-grid-item">
      
      <div class="trailer-item-info" data-video="pyalppMhuic?si=5MYeDu1Yz-kvF89w">
        <h3>Smart Contracts in Crypto</h3>
        <i class="far fa-3x fa-play-circle"></i>
      </div>
    </div>
    <div class="trailers-grid-item">
      
      <div class="trailer-item-info" data-video="kekx8QbEYHE?si=62Nky0xghmTLm319">
        <h3>Smart Contracts All About</h3>
        <i class="far fa-3x fa-play-circle"></i>
      </div>
    </div>
    <div class="trailers-grid-item">
      
    </div>
  </div>

```

```

    <div class="trailer-item-info" data-video="9PSjgELvQTo?si=TSQAeffs2L4pDVxp">
      <h3>Smart Contracts</h3>
      <i class="far fa-3x fa-play-circle"></i>
    </div>
  </div>
  <div class="trailers-grid-item">
    
    <div class="trailer-item-info" data-video="KO9mdk8CSpo?si=iUd4YvTpxEnfLv1Z">
      <h3>What is Smart Contracts</h3>
      <i class="far fa-3x fa-play-circle"></i>
    </div>
  </div>
  <div class="trailers-grid-item">
    
    <div class="trailer-item-info" data-video="4StueSP_2sw?si=QGajDS5ByBQhsxyk">
      <h3>How to Clone a Smart Contract</h3>
      <i class="far fa-3x fa-play-circle"></i>
    </div>
  </div>
</div>

```

У наведеній частині коду реалізований блок з відеоогляди переваг смарт – контрактів над аналогами. При натисканні на відео розпочнеться відтворення. Інтерфейс сторінки наведено на рисунку 3.1.

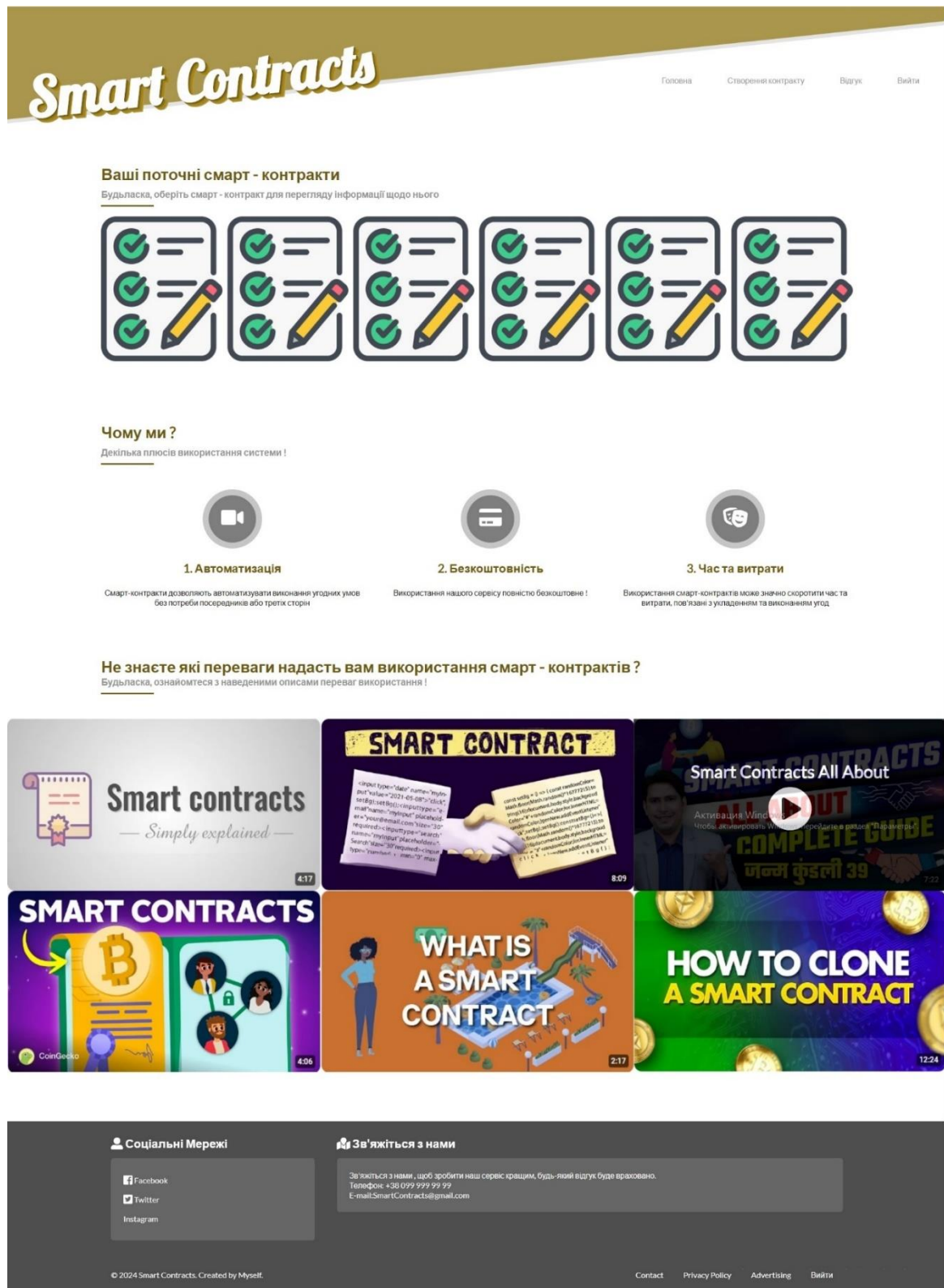


Рисунок 3.1 - Інтерфейс головної сторінки реалізованої системи

3.4 Програмна реалізація інтерфейсу взаємодії зі смарт - контрактом

На сторінці взаємодії зі смарт - контрактом було розміщено загальну інформацію про контракт, який було обрано користувачем на головній сторінці та

безпосередньо блоки, що реалізують інтерфейс підписування поточного контракту. Інформація про контракт та взаємодію користувачів з ним знаходиться у базі даних та витягується звідти за запитом користувача.

```
session_start();
$userlogin = $_SESSION['itlogin'];
$id = $_GET['id'];
$link = mysqli_connect("localhost", "root", "", "contracts");

$query = "SELECT * FROM contracts WHERE id = $id";
$imageById = mysqli_query($link,$query);
$row = mysqli_fetch_array($imageById);
$name = $row['title'];

<div class="booking-panel-section booking-panel-section4" style="width: 96%; margin-left: 2%; margin-bottom: 50px; height: 2000px; ">
    <button class="btn btn-warning" onclick="window.location = 'index.php';"><span> До вибору контракту
</span></button>
    <div class="movie-box" style="width: 20%; margin-left: 40%;">
        <?php
            echo '';
        ?>
    </div>
    <div class="title" style="text-align: center;">Контракт: <?php echo $row['title']; ?></div>

    <div style="margin-left: 5%;">
        <h4 style="text-align: center; margin-left: 10%; width: 80%; line-height: 1.5;"><?php echo $row['descr']; ?></h4>
    </div>
    <div class="title" style="text-align: center;">Подія при активації контракту:</div>
        <h4 style="text-align: center; margin-left: 10%; width: 80%; line-height: 1.5;"><?php echo $row['activation']; ?></h4>
    <div class="title" style="text-align: center;">Власник контракту:</div>
        <h4 style="text-align: center; margin-left: 10%; width: 80%; line-height: 1.5;"><?php echo $row['creator']; ?></h4>
    <div class="title" style="text-align: center;">Статус контракту:</div>
```

У наведеній частині коду реалізовано підключення до бази даних, з бази даних витягується інформація про поточний смарт - контракт за його айді, яке передається методом GET разом із завантаженням сторінки. Далі за допомогою цих даних формується інформаційний блок з описом обраного контракту.

```
<?php
// Отримання рядка з бази даних, де id = $id
$sql = "SELECT * FROM contracts WHERE id = $id";
$result = mysqli_query($link, $sql);
$row = mysqli_fetch_assoc($result);
$user_string = $row['users'];
$contract_status = $row['status'];

if ($contract_status == 'activated') {
    // Якщо контракт активовано, виводимо зелений блок
    echo '<div class="status-block activated">';
    echo ' <span>Активовано</span>';
    echo '</div>';
} else {
    // Якщо контракт не активовано, виводимо жовтий блок
    echo '<div class="status-block waiting">';
```



```

echo ' <span>В очікуванні активації</span>';
echo '</div>';
}

```

```
?>
```

Наведена частина коду реалізує визначення поточного статусу смарт – контракту на основі актуальних підписів його учасників. Якщо смарт контракт був підписаний всіма користувачами, що приймають участь у його формуванні – він отримає статус «Активовано». Якщо хочаб одного підписа не вистачає – контракт отримає статус «В очікуванні активації».

```

<div class="title" style="text-align: center;">Статус підписів учасників контракту:</div>
<div id="contractUsersListContainer">
  <!-- Список користувачів контракту -->
  <ul id="contractUsersList" class="contract-users-list">
    <?php
    // Виведення блоку з відповідним статусом

    // Розділення рядка на логіни та статуси
    $users_array = explode(',', $user_string);

    // Виведення списку користувачів контракту
    foreach ($users_array as $user) {
      // Розділення логіну та статусу
      list($login, $status) = explode('-', $user);

      echo '<li class="contract-users-item">';
      echo ' <span>' . $login . '</span>';

      // Визначення посилання на зображення відповідно до статусу
      $image_src = ($status == 'signed') ? 'green.png' : 'red.png';
      // Виведення картинки з відповідним посиланням
      echo ' <img style="width: 4%;" src="" . $image_src . "" alt="" . $status . "">';
      echo '</li>';
    }
  </ul>
  ?>
</div>

```

Наведена частина коду реалізує інтерфейс виведення статусу поточних підписів учасників смарт - контракту.

```

<div id="contractSigningContainer" style="margin-top: 4%;">
  <?php
  // Перевірка, чи є поточний користувач учасником контракту та його статус
  $sql = "SELECT users FROM contracts WHERE id = $id";
  $result = mysqli_query($link, $sql);
  $row = mysqli_fetch_assoc($result);
  $users_string = $row['users'];
  $check = $userlogin . '-signed';
  // Перевірка, чи поточний користувач підписав контракт
  $user_signed_contract = (strpos($users_string, "$check") !== false) ? 1 : 0;
  // Виведення відповідного блоку HTML
  if ($user_signed_contract) {
    // Якщо користувач підписав контракт
    echo '<div class="contract-signed">';
  }

```

```

echo ' <h3 style="text-align: center;">Ви вже підписали цей контракт</h3>';
echo ' <form action="modify_signature.php" method="post">';
echo ' <input type="text" name="digital_key" style="width: 40%; margin-left: 30%; margin-top: 2%; margin-bottom:
2%;" placeholder="Введіть ваш цифровий ключ">';
echo ' <input type="hidden" name="contract_id" value="" . $id . ">';
echo ' <br>';
echo ' <button class="btn btn-warning" style="width: 40%; margin-left: 30%;" type="submit"
name="revoke_signature">Відкликати підпис</button>';
echo ' </form>';
echo ' </div>';
} else {
    // Якщо користувач не підписав контракт
    echo ' <div class="contract-not-signed">';
    echo ' <h3 style="text-align: center;">Підпишіть цей контракт, якщо згодні з його умовами</h3>';
    echo ' <form action="modify_signature.php" method="post">';
    echo ' <input type="text" name="digital_key" style="width: 40%; margin-left: 30%; margin-top: 2%; margin-bottom:
2%;" placeholder="Введіть ваш цифровий ключ">';
    echo ' <input type="hidden" name="contract_id" value="" . $id . ">';
    echo ' <br>'; // Додаємо роздільну лінію
    echo ' <button class="btn btn-warning" style="width: 40%; margin-left: 30%;" type="submit"
name="sign_contract">Підписати контракт</button>';
    echo ' </form>';
    echo ' </div>';
}

```

У наведеній частині коду реалізовано безпосередньо логіку взаємодії зі смарт - контрактом. Система автоматично визначає, чи є користувач учасником визначеного контракту та поточний статус його взаємодії з ним. Якщо користувач ще не підписав поточний контракт – йому буде відображено інтерфейс підписання за допомогою цифрового коду, що було надано йому при реєстрації. Якщо користувач вже підписав визначений контракт – система надасть йому інтерфейс для відкликання підпису за необхідністю. Інтерфейс взаємодії зі смарт - контрактом наведено на рисунку 3.2.

До вибору контракту

Контракт: Розподіл прибутків від спільного інвестування

Цей смарт-контракт реалізує механізм автоматичного розподілу прибутків між учасниками спільного інвестування. Контракт збирає внески від усіх інвесторів і автоматично розподіляє отриманий прибуток на основі внесених сум.

Подія при активації контракту:

Активация відбувається при підписанні всіма учасниками контракту, що означає готовність до спільного інвестування. При активації контракту, система автоматично починає приймати інвестиції, і коли сума інвестицій досягає певного рівня, починається розподіл прибутку.

Власник контракту:

user1

Статус контракту:

В очікуванні активації

Статус підписів учасників контракту:

user1	✗
user2	✓
user3	✓
user4	✗
user5	✓

Підпишіть цей контракт, якщо згодні з його умовами

Введіть ваш цифровий ключ

Підписати контракт

Рисунок 3.2 - Інтерфейс взаємодії зі смарт - контрактом

3.5 Програмна реалізація логіки зміни статусу смарт - контракта

Основним принципом роботи смарт контракту є можливість автоматизованого ініціювання функції активації при умові підпису контракту всіма користувачами, що було вказано при його створенні. Для цього реалізуємо необхідний функціонал, що буде взаємодіяти з підписами користувачів і викликати функцію активації при наявності повного їх списку на поточному смарт – контракті.

```
<?php
session_start();
// Перевірка авторизації
if (empty($_SESSION['itlogin'])) {
    header("Location: /auth.php");
    exit();
} else {
```

```

    $login = $_SESSION['itlogin'];
}

$link = mysqli_connect("localhost", "root", "", "contracts");

// Перевірка з'єднання
if ($link === false) {
    die("Помилка: неможливо підключитись. " . mysqli_connect_error());
}
// Перевірка, чи була відправлена форма
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Отримання значень з форми
    $contract_id = $_POST['contract_id'];
    $digital_key = $_POST['digital_key'];

    // Отримуємо логін користувача з сесії
    $login = $_SESSION['itlogin'];

    // Отримуємо значення ключа користувача з бази даних
    $sql = "SELECT sign FROM users WHERE login = '$login'";
    $result = mysqli_query($link, $sql);
    $row = mysqli_fetch_assoc($result);
    $user_sign = $row['sign'];

```

У наведеній частині коду реалізовано логіку ініціювання сесії користувача та отримання вхідних даних щодо взаємодії за допомогою методу POST. Система отримує id поточного контракту, цифровий ключ активації підпису, що було введено користувачем, інформацію щодо поточного користувача і смарт контракту.

```

if ($user_sign == $digital_key) {
    // Значення ключа співпадає - змінюємо статус підпису в базі даних
    $sql = "SELECT users FROM contracts WHERE id = '$contract_id'";
    $result = mysqli_query($link, $sql);
    $row = mysqli_fetch_assoc($result);
    $users_data = $row['users'];
    $users_array = explode(',', $users_data);
    $user_signed_contract = false;
    foreach ($users_array as $user) {
        // Розділення логіну та статусу
        list($user_login, $status) = explode('-', $user);
        // Перевірка, чи поточний користувач підписав контракт
        if ($user_login == $login && $status == 'signed') {
            $users_data = str_replace("$login-signed", "$login-nope", $users_data);
            break;
        } else {
            $users_data = str_replace("$login-nope", "$login-signed", $users_data);
        }
    }
}

```

У наведеній частині коду система реалізує логіку перевірки валідності цифрового ключа після введення його користувачем для підпису контракту. Якщо система визначила відповідність ключа поточному користувачу – ініціюється перевірка поточного статусу взаємодії цього користувача з контрактом.

```

$users_array = explode(',', $users_data);
$user_signed_contract = false;

```

```

foreach ($users_array as $user) {
    // Розділення логіну та статусу
    list($user_login, $status) = explode('-', $user);

    // Перевірка, чи поточний користувач підписав контракт
    if ($user_login == $login && $status == 'signed') {
        $users_data = str_replace("$login-signed", "$login-nope", $users_data);
        break;
    }else{
        $users_data = str_replace("$login-nope", "$login-signed", $users_data);
    }
}
// Оновлюємо дані в таблиці "contracts"
$update_sql = "UPDATE contracts SET users = '$users_data' WHERE id = '$contract_id'";
mysqli_query($link, $update_sql);

```

```

$sql = "SELECT users FROM contracts WHERE id = $contract_id";
$result = mysqli_query($link, $sql);
$row = mysqli_fetch_assoc($result);
$users_string = $row['users'];
$all_signed = true;
$users_array = explode(',', $users_string);
foreach ($users_array as $user) {
    list($user_login, $status) = explode('-', $user);
    if ($status !== 'signed') {
        $all_signed = false;
        break;
    }
}

```

У наведеній частині коду система виконує взаємодію, що було визначено користувачем щодо поточного контракту у випадку співпадання ключа підпису.

```

if ($all_signed) {
    // Змінюємо значення стовбця status на 'activated'
    $update_contract_sql = "UPDATE contracts SET status = 'activated' WHERE id = $contract_id";
    mysqli_query($link, $update_contract_sql);
    $link_sql = "SELECT link FROM contracts WHERE id = $contract_id";
    $result = mysqli_query($link, $link_sql);
    $row = mysqli_fetch_assoc($result);
    $redirect_link = $row['link'];

    // активація контракту
    header("Location: $redirect_link");
    exit;
}
$_SESSION['message'] = "Статус підпису змінено успішно.";
} else {
    // Повідомлення про невірний ключ
    $_SESSION['message'] = "Невірний цифровий ключ. Будь ласка, перевірте його та спробуйте знову.";
}
header("Location: " . $_SERVER["HTTP_REFERER"]);
exit();
}

```

У наведеній частині коду система виконує перевірку наявності підписів всіх користувачів поточного смарт – контракту після кожної взаємодії з ним. Якщо система визначила, що смарт – контракт було підписано всіма користувачами –

вона автоматично ініціює функцію активації, що було вказано для поточного смарт – контракту при його створенні.

3.6 Програмна реалізація інтерфейсу створення нового смарт - контракту

Для того, щоб мати можливість взаємодіяти зі смарт контрактами – користувач повинен мати можливість створювати їх та визначати групи зацікавлених сторін, що будуть підписувати його, функцію активації, посилання активації тощо.

```
<h1 style="text-align: center; margin-bottom: 2%;">Створення нового smart - контракту</h1>
<form action="addcontract.php" method="post">
  <label for="name">Назва контракту:</label>
  <input type="text" name="name" required><br>
  <label for="description">Опис контракту:</label>
  <textarea name="description" rows="4" cols="50" required></textarea><br>
  <label for="activation_link">Посилання для активації:</label>
  <input type="text" name="activation_link" required><br>
  <label for="activation_function">Функція активації:</label>
  <textarea name="activation_function" rows="4" cols="50" required></textarea><br>
  <label for="participants[]">Учасники контракту:</label><br>
  <select name="participants[]" multiple required>
    <?php
      $sql = "SELECT login FROM users";
      $result = mysqli_query($link, $sql);
      while ($row = mysqli_fetch_assoc($result)) {
        echo '<option value="" . $row['login'] . "">' . $row['login'] . '</option>';
      }
    ?>
```

Наведена частина коду реалізує інтерфейс для надання можливості користувачу створити новий смарт – контракт.

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $name = $_POST['name'];
  $description = $_POST['description'];
  $activation_link = $_POST['activation_link'];
  $activation_function = $_POST['activation_function'];
  $creator = $login;
  $participants = implode(',', $_POST['participants']);
  $users = "";
  foreach ($_POST['participants'] as $participant) {
    $users .= $participant . '-nope,'; // Додаємо кожного користувача
  }
  $users = rtrim($users, ','); // Видаляємо останню кому
  $status = 'waiting';
  $sql = "INSERT INTO contracts (img, title, descr, link, activation, creator, users, status) VALUES ('img/unknown.jpg',
'$name', '$description', '$activation_link', '$activation_function', '$creator', '$users', '$status')";
  if (mysqli_query($link, $sql)) {
    $_SESSION['message'] = "Контракт успішно створено.";
    header("Location: index.php");
    exit();
  } else {
    echo "Помилка: " . mysqli_error($link);
  }
  mysqli_close($link);
```

У наведеній частині коду реалізовано безпосередньо логіку обробки даних користувача щодо створення нового смарт – контракту. Система виконує перевірку даних, та якщо вони коректні – створює новий смарт – контракт та заносить інформацію щодо нього до бази даних. Інтерфейс створення нового смарт – контракту наведено на рисунку 3.3.

The screenshot displays the 'Smart Contracts' web application. At the top, a gold banner features the 'Smart Contracts' logo in a stylized, 3D font. To the right of the logo are navigation links: 'Головна', 'Створення контракту', 'Відгук', and 'Вийти'. Below the banner, the main heading is 'Створення нового smart - контракту'. The central form contains several input fields: 'Назва контракту:', 'Опис контракту:', 'Посилання для активації:', and 'Функція активації:'. Below these is a dropdown menu for 'Учасники контракту:' with options 'user1', 'user11', and 'user12'. A green 'Створити контракт' button is at the bottom of the form. The footer is dark gray and includes social media links (Facebook, Twitter, Instagram) under 'Соціальні Мережі', contact information (phone and email) under 'Зв'яжіться з нами', and a copyright notice '© 2024 Smart Contracts. Created by Myself.' on the left. On the right, there are links for 'Contact', 'Privacy Policy', 'Advertising', and 'Вийти'.

Рисунок 3.3 – Інтерфейс створення нового смарт – контракту

3.7 Програмна реалізація інтерфейсу написання відгуків щодо використання системи

На сторінці написання відгуків щодо використання системи було розміщено основну інформацію про систему, адресу умовного розташування офісу, телефони та Email, а також була вбудована картка з позначкою розташування. Нижче була розташована форма, за допомогою якої користувач може залишити свій відгук про використання послуг системи.

```
<?php
$link = mysqli_connect("localhost", "root", "", "contracts");
?>
```

```

<header></header>
<div class="gmap_canvas"><iframe id="gmap_canvas"
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3262.5815823728194!2d-
80.73889560000869!3d35.14211377150105!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x885421a2eb1e7e
a9%3A0x338011bddfd77820!2sWeb%20Development%20Company%20Charlotte!5e0!3m2!1sru!2sua!4v1654828127938
!5m2!1sru!2sua" width="600" height="450" style="border:0;" allowfullscreen="" loading="lazy" referrerpolicy="no-
referrer-when-downgrade"></iframe>
</div>
<div class="contact-us-container">
<div class="contact-us-section contact-us-section1">
<h1>Залиште ваш відгук</h1>
<p>Сприяйте кращій роботі сервісу. Будь ласка, залиште ваш відгук про нашу роботу.</p>
<form action="" method="POST">
<input placeholder="Ваше ім'я" name="fName" required><br>
<input placeholder="Тема відгуку" name="lName" ><br>
<input placeholder="Ваш E-mail адрес" name="email" required><br>
<textarea placeholder="Ваш відгук !" name="feedback" rows="10" cols="30" required></textarea><br>
<button type="submit" name="submit" value="submit">Відправити відгук</button>
<?php
if(isset($_POST['submit'])){
$insert_query = "INSERT INTO
feedbackTable ( senderfName,
senderlName,
sendereMail,
senderfeedback)
VALUES (
".$_POST["fName"].",
".$_POST["lName"].",
".$_POST["eMail"].",
".$_POST["feedback"].")";
mysqli_query($link,$insert_query);
}
?>
</form>
</div>
<div class="contact-us-section contact-us-section2">
<h1>Адреса & Інформація</h1>
<h3>Наші номери</h3>
<p><a href="tel:01011391148">+38 099 999 99 99</a><br>
<a href="tel:01011391148">+38 099 999 99 99</a></p>
<h3>Адрес</h3>
<p>El Sherouk City , Suez Desert Road , Cairo 11837 - P.O. Box 43</p>
<h3>E-mail</h3>
<p><a href="#">smartcontracts@gmail.com</a></p>

```

Наведена частина коду реалізує підключення до бази даних відгуків, а також відображення та функціонування форми для надсилання відгуку. Для надсилання користувач повинен заповнити поля форми і за натисканням кнопки надіслати дані якщо вони валідні будуть відправлені до бази даних відгуків для подальшої обробки.

The screenshot displays the 'Smart Contracts' website. At the top, there's a navigation bar with links: 'Головна', 'Створення контракту', 'Відгук', and 'Вийти'. Below the navigation bar is a map of a city area with various location markers. Under the map, there are two main sections: 'Залиште ваш відгук' (Leave your review) and 'Адреса & Інформація' (Address & Information).

Залиште ваш відгук

Спробуйте кращі роботи сервісу. Будь ласка, залиште ваш відгук про нашу роботу.

Ваше ім'я:

Тема відгуку:

Ваш e-mail адрес:

Ваш відгук:

Адреса & Інформація

Наші номери

+38 099 999 99 99
+38 099 999 99 99

Адрес

El Sherouk City, Suez Desert Road, Cairo 11837 - P.O. Box 43

E-mail

smartcontracts@gmail.com

Соціальні Мережі

Facebook
Twitter
Instagram

З'єднати з нами

З'єднати з нами, щоб зробити наш сервіс кращим, будь-який відгук буде кращим.
Телефон: +38 099 999 99 99
E-mail: SmartContracts@gmail.com

© 2024 Smart Contracts. Created by Myself.

Contact Privacy Policy Advertising Вийти

Рисунок 3.4 – Інтерфейс функціоналу написання відгуків

3.8 Програмна реалізація логіки реєстрації у системі

На даній сторінці було розміщено форму для реєстрації користувача у системі. Реєстрація обов'язкова для подальшої взаємодії зі смарт - контрактами. На сторінці реалізується форма реєстрації з полями логіну, паролю, ім'я, прізвища та країни користувача.

Логіка сторінки відповідає за перевірку даних з реєстраційної форми на валідність та якщо вони валідні – відправлення даних до бази даних користувачів та формування цифрового коду підпису для поточного користувача. Програмну реалізацію зазначеної логіки наведено далі:

```
<?php
$link = mysqli_connect("localhost", "root", "", "contracts");
?>
<div class="navbar-brand">
  <a href="index.php">
    <h1 class="navbar-heading">Smart Contracts</h1>
  </a>
</div>
<div>
  <div mt="150px" class="container">
```

```

<div class="row">
  <div class="col-3"></div>
  <div class="col-6">
    <h1><center>Будьласка, заповніть форму для реєстрації у системі</center></h1><br>

    <form action="" method="POST" >
      <div class="box-body">

        <div class="step_1">
          <div class="form-group">
            <label for="First_name">Лорін:</label>
            <input type="text" class="form-control first_name " name="name" required>
          </div><br>

          <div class="form-group">
            <label for="Second_name">Ім'я:</label>
            <input type="text" class="form-control second_name" name="sname" required>
          </div><br>

          <div class="form-group">
            <label for="birthday">Прізвище:</label>
            <input class="form-control birthday" required type="text" name="company" >
          </div><br>

          <div class="form-group">
            <label for="subject">Пароль:</label>
            <input class="form-control subject" required type="text" name="position" >
          </div><br>

          <div class="form-group">
            <label for="country">Країна:</label>
            <select class="form-control country" name="country" id="country" >
              <?php
                $arr = array("Україна", "Польша", "Казахстан", "Інші");
                foreach ($arr as &$value) {
                  echo "<option>".$value."</option>";
                }?>
            </select>
          </div><br>

          <div class="box-footer">
            <button type="submit" style="width: 100%;" name="submit" value="submit" class="btn btn-
danger">Зареєструватися</button>
          </div>
          <div class="auth_button">
            <h4 style="margin-top: 5%;">Вже зареєстровані?</h4>
            <a href="auth.php">Увійти</a>
          </div>
        </div>
      </div>
    <?php

    if(isset($_POST['submit'])){
      $tyy = $_POST['name'];
      $sql = "SELECT id FROM users WHERE login = '$tyy'";
      $result = mysqli_query($link,$sql);
      if(mysqli_num_rows($result)==0){
        $random_key = substr(md5(uniqid(mt_rand(), true)), 0, 20); // Генеруємо випадковий ключ
        $insert_query = "INSERT INTO

```

```

users ( login,
        name,
        lname,
        password,
        country,
        sign
      )
VALUES (
    '$_POST["name"].',
    '$_POST["sname"].',
    '$_POST["company"].',
    '$_POST["position"].',
    '$_POST["country"].',
    '$_random_key.'
);
mysqli_query($link,$insert_query);
$_SESSION['itlogin'] = $tyy;
echo "<script>
var generatedKey = '$_random_key';
var confirmation = confirm('Успішна реєстрація! Ваш ключ для цифрового підпису контрактів: ' +
generatedKey + '. Натисніть OK, щоб перейти на сторінку або відмініть для залишення на цій сторінці.');"
if (confirmation) {
    window.location.href = 'index.php';
}
</script>";
}else{
    echo "<script>alert(\"Цей логін вже зайнятий!\");</script>";
}
}
?>

```

Наведена логіка реалізує інтерфейс реєстрації користувача у системі у вигляді форми для введення особистих даних. Після введення дані буде перевірено на наявність у базі даних. Якщо такого користувача ще не існує – його дані буде занесено до бази даних для подальшої авторизації у системі, а також згенеровано та надано індивідуальний цифровий код для взаємодії зі смарт – контрактами.

Smart Contracts

Будьласка, заповніть форму для реєстрації у системі

Логін:

Емел:

Провіда:

Пароль:

Країна:
Україна

Зареєструватися

Вже зареєстровані?
[Увійти](#)

Соціальні Мережі

Зв'яжіться з нами

Зв'яжіться з нами, щоб зробити наш сервіс кращим, будь-який відгук буде враховано.
Телефон: +38 099 999 99 99
E-mail: SmartContracts@gmail.com

© 2024 Smart Contracts. Created by Myself.

Contact Privacy Policy Advertising

Рисунок 3.5 – Інтерфейс реєстрації користувача у системі

3.9 Програмна реалізація сторінки авторизації

Для авторизації користувача у системі було реалізовано відповідний інтерфейс, на якому було розміщено форму авторизації. Введені у форму дані будуть перевірені на співпадання з базою даних користувачів, та якщо вони співпадають, користувача буде переміщено до головної сторінки системи.

На сторінці реалізується підключення до бази даних, створення форми з полями імені та паролю та перевірка їх наявності у базі даних. Програмну реалізацію зазначеного функціоналу наведено далі:

```
<?php
$link = mysqli_connect("localhost", "root", "", "contracts");
?>
<div class="navbar-brand">
  <a href="index.php">
    <h1 class="navbar-heading">Smart Contracts</h1>
  </a>
</div>
<div>
<div mt="150px" class="container">
  <div class="row" style="margin-bottom: 20%;">
    <div class="col-3"></div>
```

```

<div class="col-6">
  <h1><center>Будь ласка, введіть свій логін та пароль</center></h1><br>

  <form action="" method="POST" >
    <div class="box-body">

      <div class="step_1">
        <div class="form-group">
          <label for="First_name">Логін:</label>
          <input type="text" class="form-control first_name " name="name" required>
        </div><br>

        <div class="form-group">
          <label for="subject">Пароль:</label>
          <input class="form-control subject" required type="text" name="position" >
        </div><br>

        <div class="box-footer">
          <button type="submit" style="width: 100%;" name="submit" value="submit" class="btn btn-
danger">Увійти</button>
          <button type="button" style="width: 100%; margin-top: 2%; background-color: #1b9b4c;"
onclick="window.location.href = 'registration.php';" class="btn btn-danger">Зареєструватися</button>
        </div>
      </div>
    </div>
  <?php

  if(isset($_POST['submit'])){
    $tyy = $_POST['name'];
    $pass = $_POST['position'];
    $sql = "SELECT id FROM users WHERE login = '$tyy'";
    $check = "SELECT id FROM users WHERE login = '$tyy' and password = '$pass'";
    $result = mysqli_query($link,$sql);
    $res = mysqli_query($link,$check);
    if(mysqli_num_rows($result)==0) {
      echo "Такого користувача не існує, будь ласка, зареєструйтесь!";
    }elseif(mysqli_num_rows($res)==0){
      echo "Введений пароль невірний, спробуйте знову";
    }else{
      $_SESSION['itlogin'] = $tyy;
      header ("location: /index.php");
    }
  }
  ?>
</form>
</div>

```

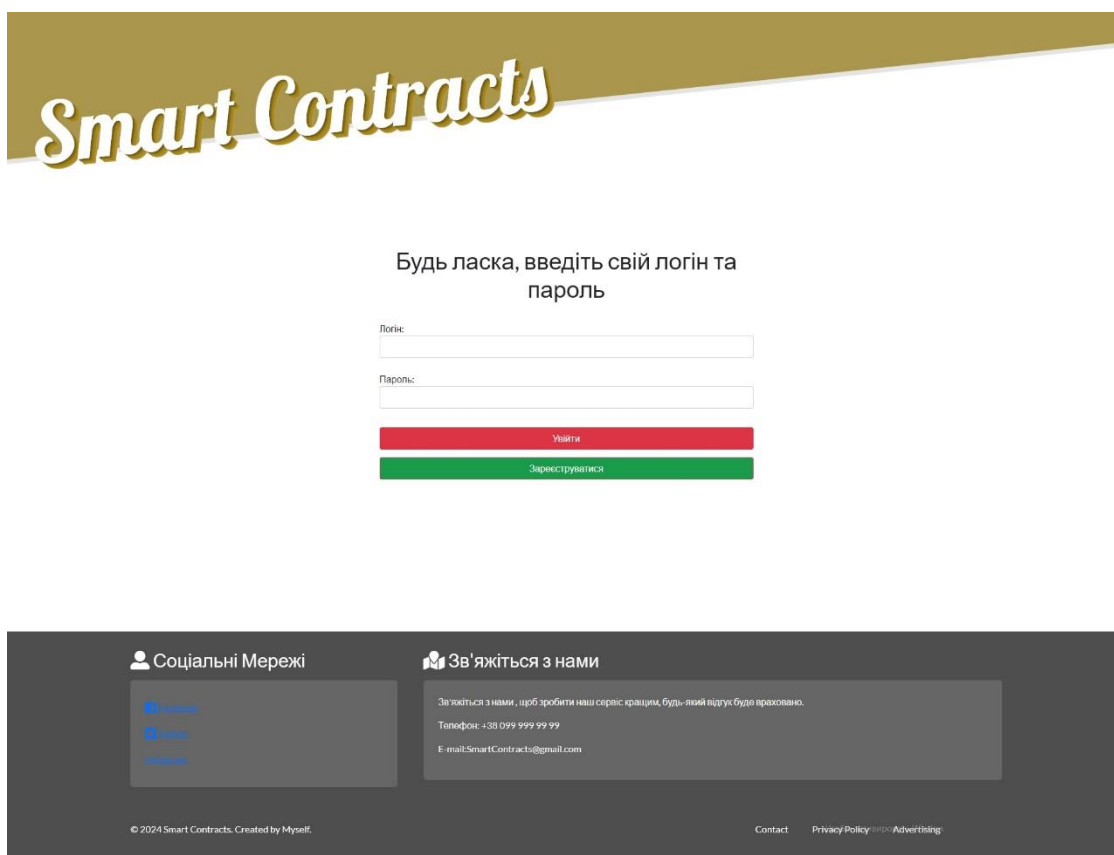


Рисунок 3.6 - Інтерфейс авторизації користувача у системі

3.10 Висновок до розділу 3

У цьому розділі було детально розглянуто програмну реалізацію ключових компонентів веб-додатку для роботи зі смарт-контрактами, з особливою увагою до створення зручних та функціональних інтерфейсів користувача і забезпечення коректної програмної логіки системи.

Перш за все, було визначено призначення кожного програмного модуля, що дало змогу чітко зрозуміти їх функції та ролі у системі. Це дозволило забезпечити ефективну інтеграцію модулів і їх гармонійну взаємодію, що є основою для безперебійної роботи веб-додатку.

Наступним важливим аспектом була розробка універсальних компонентів для верхнього та нижнього колонтитулів (header і footer), які забезпечують навігацію та єдину стилістичну концепцію всього сайту. Це сприяло створенню інтуїтивно зрозумілого та естетично привабливого інтерфейсу для користувачів.

Особливу увагу було приділено розробці головної сторінки веб-додатку, яка є ключовим елементом, що забезпечує доступ до основних функцій системи. Це

включало інтерактивний дизайн та логіку, які гарантують зручність користування та ефективність роботи.

Також було реалізовано інтерфейс взаємодії зі смарт-контрактами, що дозволяє користувачам переглядати, підписувати та управляти смарт-контрактами. Це забезпечує високу функціональність та безпеку при роботі з блокчейн-технологіями.

Розробка алгоритмів зміни статусу смарт-контрактів була важливим кроком для забезпечення динамічного управління контрактами залежно від дій користувачів. Це забезпечує гнучкість і надійність роботи системи.

Створення інтерфейсу для генерування нових смарт-контрактів було реалізовано з метою забезпечення простоти та зручності цього процесу для користувачів. Це дозволяє швидко та ефективно створювати нові контракти без необхідності в глибоких технічних знаннях.

Також було розроблено форму для залишення зворотного зв'язку користувачами, що є важливим елементом для постійного вдосконалення системи на основі реальних відгуків та побажань користувачів.

Окрім цього, було реалізовано процес реєстрації нових користувачів та створено сторінку для входу зареєстрованих користувачів у систему. Це забезпечує безпеку та контроль доступу до системи, а також зручність для користувачів при вході в систему.

ВИСНОВОК

Під час виконання дипломної роботи була проведена робота з вивчення технологій розробки веб – орієнтованих систем з використанням технологій PHP, MySql, HTML, CSS. За підсумками виконання роботи було розроблено інформаційну систему для автоматизації процесів групового цифрового підписування смарт контрактів.

Реалізовано функції користувача, що забезпечить можливостями перегляду основної інформації щодо смарт - контрактів, що актуальні для поточного користувача, взаємодії з поточними контрактами, цифрового підпису обраного контракту, створення нових смарт – контрактів, написання відгуків, надсилання заявок на співпрацю.

Створена система дозволяє користувачу використовувати функціонал додатку за допомогою інтерфейсу користувача, та керувати вмістом додатку та бази даних, змінювати, видаляти та додавати інформацію щодо поточних смарт - контрактів. При створенні системи були враховані всі плюси та мінуси систем-аналогів, вибраних для порівняння.

Проведено всі необхідні аналітичні процеси для проектування та програмування системи та бази даних. Після реалізації системи обраними мовами програмування та середовищах, було проведено випробування на працездатність, яке засвідчило, що система виконує усі поставлені завданням функції.

Оскільки порушень не виявлено, всі функції відповідають вимогам, тому можна визначити, що реалізована система готова для впровадження, та може бути використана для автоматизації процесів групового цифрового підписування смарт контрактів.

СПИСОК ЛІТЕРАТУРИ

1. Docusign. URL: <https://www.docusign.com/>.
2. Adobe. URL: <https://www.adobe.com/ua/sign.html>.
3. Hellosign. URL: <https://app.hellosign.com/account/login>.
4. GDPR (Загальний регламент з захисту персональних даних): Європейський парламент та Рада, 2016. - URL: <https://eur-lex.europa.eu/eli/reg/2016/679/oj>.
5. Закон України "Про захист персональних даних": прийнятий Верховною Радою України 01.06.2010. - URL: <https://zakon.rada.gov.ua/laws/show/2297-17#Text>.
6. Закон України "Про рекламу": прийнятий Верховною Радою України 03.04.1997. - URL: <https://zakon.rada.gov.ua/laws/show/270/96-вр#Text>.
7. Закон України "Про захист прав споживачів": прийнятий Верховною Радою України 12.05.1991. - URL: <https://zakon.rada.gov.ua/laws/show/1023-12#Text>.
8. Закон України "Про електронну комерцію": прийнятий Верховною Радою України 03.09.2015. - URL: <https://zakon.rada.gov.ua/laws/show/675-19>.
9. Marshall S. Social Media Analytics: Effective Tools for Building, Interpreting, and Using Metrics / McGraw Hill LLC, 2011. 320 с.
10. Reza Z., Mohammad A., Abbasi H. Social Media Mining: An Introduction / Cambridge University Press, 2014. 382 с.
11. Gabor S., Gungor P. Social Media Analytics and Data Mining / John Wiley & Sons, 2018. 352 с.
12. Michael B. Practical Web Analytics for User Experience: How Analytics Can Help You Understand Your Users / Elsevier Science, 2013. 256 с.
13. Matthew G., Avinash K. Social Media Analytics: Techniques and Insights for Extracting Business Value Out of Social Media / Pearson Education, 2015, 304 с.
14. Foster P., Tom F. Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking / O'Reilly Media, 2013. 414 с.
15. Nick H. Social Media Mining: Social Media Branding and Marketing Strategies / IGI Global, 2015. 438с.
16. Duckett, J. "HTML & CSS: Design and Build Websites." Wiley, 2014. 490 p.

17. Ducket, J. "JavaScript and jQuery: Interactive Front-End Web Development." Wiley, 2015. 640 p.
18. Flanagan, D. "JavaScript: The Good Parts." O'Reilly Media, 2006. 176 p.
19. Freeman, E., & Robson, E. "Head First JavaScript Programming: A Brain-Friendly Guide." O'Reilly Media, 2017. 704 p
20. Vacca, J. R. "Computer and Information Security Handbook," 2nd Edition. New York: McGraw-Hill, 2013. 1104 p
21. Smith, J. "Web Development with PHP, HTML, and JavaScript: A Step by Step Guide to Creating Dynamic Websites." Packt Publishing, 2020. 350 p.

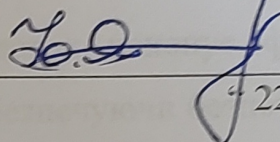
ДОДАТКИ

ДОДАТОК А. Технічне завдання

Вінницький національний технічний університет
Факультет менеджменту та інформаційної безпеки
Кафедра менеджменту та безпеки інформаційних систем

ЗАТВЕРДЖУЮ

Голова секції “Управління інформаційною
безпекою” кафедри МБІС
д.т.н., професор

 Юрій ЯРЕМЧУК
“ 22 ” березня 2024 р.

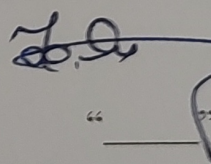
ТЕХНІЧНЕ ЗАВДАННЯ

до бакалаврської дипломної роботи на тему:

Розробка програми групового цифрового підписування смарт контрактів

08-72.БДР.007.00.104.ТЗ

Керівник бакалаврської дипломної роботи

 д.т.н., професор
Яремчук Ю.Є.

1. Найменування та область застосування

Веб-додаток для групового цифрового підписування смарт контрактів. Область застосування: автоматизація та безпека процесу підписання смарт-контрактів у багатосторонніх угодах.

2. Підстава для розробки

Розробка виконується на основі наказу ректора ВНТУ № 80 від 11.03.2024 р.

3. Мета та призначення розробки

3.1 Мета розробки: розробка ефективного веб-додатку для групового цифрового підписування смарт-контрактів, що забезпечує автоматизацію та безпеку багатосторонніх угод.

3.2 Призначення: розроблений веб-додаток виконує функції групового цифрового підписування смарт-контрактів, забезпечуючи безпечний і ефективний процес укладання угод між багатьма учасниками.

4. Джерела розробки

4.1. Duckett, J. "HTML & CSS: Design and Build Websites." Wiley, 2014. 490 p.

4.2. Duckett, J. "JavaScript and jQuery: Interactive Front-End Web Development." Wiley, 2015. 640 p.

4.3. Smith, J. "Web Development with PHP, HTML, and JavaScript: A Step by Step Guide to Creating Dynamic Websites." Packt Publishing, 2020. 350 p.

4.4. Gabor S., Gungor P. Social Media Analytics and Data Mining / John Wiley & Sons, 2018. 352 c.

5. Вимоги до програми

5.1 Вимоги до функціональних характеристик:

5.1.1 Програмний засіб повинен мати зручний, легкий у використанні інтерфейс користувача;

5.1.2 Реалізація методу не повинна вимагати спеціальних ліцензійних програмних додатків;

5.1.3 Програмний засіб повинен виконувати процес автентифікації користувачів у системі.

5.2 Вимоги до надійності:

5.2.1 Програмний засіб повинен працювати без помилок, у випадку виникнення критичних ситуацій необхідно передбачити виведення відповідних повідомлень;

5.2.2 Програмний засіб повинен виконувати свої функції.

5.3 Вимоги до складу і параметрів технічних засобів:

- Даний проект реалізований мовою програмування PHP, для поведінки сторінок використана мова JavaScript, для роботи з базою даних СУБД MySQL, а також мови розмітки HTML та CSS.

- Оперативна пам'ять 512Mb;
- Процесор із частотою не менше 2.50GHz;
- Дисківне місце на стороні сервера 5-10Gb.

6. Вимоги до програмної документації

6.1 Обов'язкова інструкція функціоналу для майбутніх користувачів, наведена у пункті 2.1

7. Вимоги до технічного захисту інформації

7.1 Необхідно забезпечити захист розроблюваного програмного засобу від несанкціонованого використання.

7.2 Неможливість отримання доступу незареєстрованих користувачів до інформаційних ресурсів.

8. Техніко-економічні показники

8.1 Цінність результатів використання даного проекту повинна перевищувати витрати на його реалізацію.

8.2 Має бути реалізований таким чином, щоб підходити для використання широкого загалу.

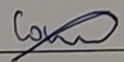
9. Стадії та етапи розробки

№ з/п	Назва етапів бакалаврської дипломної роботи	Початок	Закінчення
1.	Визначення напрямку бакалаврської роботи, формулювання теми	21.03.2024	22.03.2024
2.	Аналіз предметної області обраної теми	23.03.2024	21.04.2024
3.	Розробка алгоритму роботи	10.05.2024	18.05.2024
4.	Написання бакалаврської роботи на основі розробленої теми	18.05.2024	25.05.2024
5.	Передзахист бакалаврської дипломної роботи	26.05.2024	27.05.2024
6.	Виправлення, уточнення, корегування бакалаврської дипломної роботи	28.05.2024	6.06.2024
7.	Захист бакалаврської дипломної роботи	12.06.2024	17.06.2024

10. Порядок контролю та прийому

10.1 До приймання бакалаврської дипломної роботи надається:

- ПЗ до бакалаврської дипломної роботи;
- демонстрація результату бакалаврської дипломної роботи;
- презентація;
- відзив керівника роботи;
- відзив рецензента

Технічне завдання до виконання прийняв  Гончарук О.М.

ДОДАТОК Б. Лістинг бази даних

```
-- phpMyAdmin SQL Dump
-- version 5.2.0
-- https://www.phpmyadmin.net/
--
-- Хост: 127.0.0.1:3306
-- Час створення: Квт 24 2024 р., 08:55
-- Версія сервера: 8.0.30
-- Версія PHP: 8.1.9

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- База даних: `contracts`
--

--
-- Структура таблиці `contracts`
--

CREATE TABLE `contracts` (
  `id` int NOT NULL,
  `img` varchar(1500) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
  `title` varchar(1000) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
  `descr` varchar(500) COLLATE utf8mb4_general_ci NOT NULL,
  `link` varchar(5000) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
  `activation` varchar(500) COLLATE utf8mb4_general_ci NOT NULL,
  `creator` varchar(500) COLLATE utf8mb4_general_ci NOT NULL,
  `users` varchar(500) COLLATE utf8mb4_general_ci NOT NULL,
  `status` varchar(500) COLLATE utf8mb4_general_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Дамп даних таблиці `contracts`
--

INSERT INTO `contracts` (`id`, `img`, `title`, `descr`, `link`, `activation`, `creator`, `users`, `status`) VALUES
(1, 'img/unknown.jpg', 'Розподіл прибутків від спільного інвестування', 'Цей смарт-контракт реалізує механізм автоматичного розподілу прибутків між учасниками спільного інвестування. Контракт збирає внески від усіх інвесторів і автоматично розподіляє отриманий прибуток на основі внесених сум.', 'activationfunction.php', 'Активация відбувається при підписанні всіма учасниками контракту, що означає готовність до спільного інвестування. При активації контракту, система автоматично починає приймати інвестиції, і коли сума інвестицій досягає певного рівня, починається розподіл прибутку.', 'user1', 'user1-nope, user2-signed, user3-signed, user4-nope, user5-signed', 'waiting'),
(2, 'img/unknown.jpg', 'Організація масштабних заходів', 'Цей смарт-контракт призначений для автоматизації організації та управління масштабними заходами, такими як конференції, фестивалі, тощо. Контракт забезпечує автоматичне управління бюджетом заходу, реєстрацію учасників, оплату послуг і постачальників.', 'activationfunction.php\n', 'Активация відбувається при підписанні всіма учасниками контракту. При активації контракту, система автоматично запускає процеси організації заходу, включаючи реєстрацію учасників, відбір постачальників, укладання договорів і оплату послуг.', 'user1', '', ''),
(3, 'img/unknown.jpg', 'Розподіл спадщини', 'Цей смарт-контракт дозволяє автоматично розподілити спадщину між
```


спадкоємцями відповідно до встановлених правил та умов. Контракт збирає інформацію про спадок та права спадкоємців і автоматично розподіляє майно згідно з вказаними умовами.', 'activationfunction.php\n', 'Активация відбувається при підписанні всіма спадкоємцями контракту. Після активації, система автоматично визначає розподіл майна між спадкоємцями згідно з встановленими правилами.', 'user1', "", ""),

(4, 'img/unknown.jpg', 'Розподіл прибутків від колективної творчості', 'Цей смарт-контракт створений для автоматичного розподілу прибутків від колективної творчості, наприклад, створення музичних альбомів або фільмів. Контракт збирає внески та визначає частки прибутку кожного учасника на основі їхнього внеску.', 'activationfunction.php\n', 'Активация відбувається при підписанні всіма учасниками контракту. Після активації, система автоматично визначає розподіл прибутку від продажу твору серед учасників згідно з установленими правилами.', 'user1', "", ""),

(5, 'img/unknown.jpg', 'Децентралізований фонд благодійності', 'Цей смарт-контракт створює децентралізований фонд благодійності, який автоматично збирає кошти від добровільних внесків та надає фінансову допомогу потребуючим. Контракт забезпечує прозорий і ефективний механізм благодійної допомоги.', 'activationfunction.php\n', 'Активация відбувається при підписанні всіма учасниками контракту. Після активації, система автоматично приймає благодійні внески та надає фінансову допомогу потребуючим згідно з встановленими правилами.', 'user1', "", ""),

(6, 'img/unknown.jpg', 'Автоматична розподіл ресурсів у спільному господарстві', 'Цей смарт-контракт забезпечує автоматичний розподіл ресурсів, таких як електроенергія або вода, у спільному господарстві між учасниками на основі їхніх потреб та внесків. Контракт забезпечує ефективне використання ресурсів та уникнення конфліктів.', 'activationfunction.php\n', 'Активация відбувається при підписанні всіма учасниками контракту. Після активації, система автоматично визначає розподіл ресурсів між учасниками згідно з їхніми потребами та внесками.', 'user1', "", "");

-- -----

--

-- Структура таблиці `feedbacktable`

--

```
CREATE TABLE `feedbacktable` (
  `id` int NOT NULL,
  `senderfName` varchar(50) NOT NULL,
  `senderlName` varchar(50) DEFAULT NULL,
  `sendereMail` varchar(100) NOT NULL,
  `senderfeedback` varchar(500) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

--

-- Дамп даних таблиці `feedbacktable`

--

```
INSERT INTO `feedbacktable` (`id`, `senderfName`, `senderlName`, `sendereMail`, `senderfeedback`) VALUES
(5, 'Alex', 'Dikkens', '', 'sdggdg');
```

-- -----

--

-- Структура таблиці `users`

--

```
CREATE TABLE `users` (
  `id` int NOT NULL,
  `login` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
  `name` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
  `lname` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
  `password` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
  `country` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
  `sign` varchar(500) COLLATE utf8mb4_general_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```

--
-- Дамп даних таблиці `users`
--

INSERT INTO `users` (`id`, `login`, `name`, `lname`, `password`, `country`, `sign`) VALUES
(15, 'user1', 'user', 'sgds', 'user1', 'Україна', '9894fb480903000f7fc5'),
(19, 'user11', 'user11', 'user11', 'user11', 'Україна', '9894fb480903000f7fc5'),
(20, 'user12', 'user12', 'user12', 'user12', 'Україна', 'cb694f345e3b33543ba5');

--
-- Індокси збережених таблиць
--

--
-- Індокси таблиці `contracts`
--
ALTER TABLE `contracts`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `movieID` (`id`);

--
-- Індокси таблиці `feedbacktable`
--
ALTER TABLE `feedbacktable`
  ADD PRIMARY KEY (`id`),
  ADD UNIQUE KEY `msgID` (`id`);

--
-- Індокси таблиці `users`
--
ALTER TABLE `users`
  ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT для збережених таблиць
--
--
-- AUTO_INCREMENT для таблиці `contracts`
--
ALTER TABLE `contracts`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=22;

--
-- AUTO_INCREMENT для таблиці `feedbacktable`
--
ALTER TABLE `feedbacktable`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

--
-- AUTO_INCREMENT для таблиці `users`
--
ALTER TABLE `users`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */

```

ДОДАТОК Б. Програмний код

Addcontract.php

```
<?php
session_start();
if($_SESSION['itlogin'] == ""){
    header('Location: /auth.php');
}else{
    $login = $_SESSION['itlogin'];
}
$link = mysqli_connect("localhost", "root", "", "contracts");

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="icon" type="image/png" href="img/logo.png">
    <link rel="stylesheet" href="style/styles.css">
    <title>Додавання нового контракту</title>
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384-
    fnmOCqbTIWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous">
    <link rel="stylesheet" href="style.css" />
    <link
        rel="stylesheet"
        href="https://cdnjs.cloudflare.com/ajax/libs/odometer.js/0.4.8/themes/odometer-theme-default.min.css"
    />
</head>

<header></header>
<body>
<?

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST['name'];
    $description = $_POST['description'];
    $activation_link = $_POST['activation_link'];
    $activation_function = $_POST['activation_function'];
    $creator = $login;
    $participants = implode(',', $_POST['participants']); // Перетворюємо масив в рядок
    $users = "";
    foreach ($_POST['participants'] as $participant) {
        $users .= $participant . '-nore,'; // Додаємо кожного користувача зі статусом "nore"
    }
    $users = rtrim($users, ','); // Видаляємо останню кому
    $status = 'waiting';

    // Вставка даних в базу даних
```

```

    $sql = "INSERT INTO contracts (img, title, descr, link, activation, creator, users, status) VALUES ('img/unknown.jpg',
'name', '$description', '$activation_link', '$activation_function', '$creator', '$users', '$status')";
    if (mysqli_query($link, $sql)) {
        $_SESSION['message'] = "Контракт успішно створено.";
        header("Location: index.php");
        exit();
    } else {
        echo "Помилка: " . mysqli_error($link);
    }

    // Закриття з'єднання з базою даних
    mysqli_close($link);
}

?>

```

<h1 style="text-align: center; margin-bottom: 2%;">Створення нового smart - контракту</h1>

```

<form action="addcontract.php" method="post">
    <label for="name">Назва контракту:</label>
    <input type="text" name="name" required><br>

    <label for="description">Опис контракту:</label>
    <textarea name="description" rows="4" cols="50" required></textarea><br>

    <label for="activation_link">Посилання для активації:</label>
    <input type="text" name="activation_link" required><br>

    <label for="activation_function">Функція активації:</label>
    <textarea name="activation_function" rows="4" cols="50" required></textarea><br>

    <label for="participants[]">Учасники контракту:</label><br>
    <select name="participants[]" multiple required>
        <?php
            $sql = "SELECT login FROM users";
            $result = mysqli_query($link, $sql);
            while ($row = mysqli_fetch_assoc($result)) {
                echo '<option value="' . $row['login'] . '">' . $row['login'] . '</option>';
            }
        ?>
    </select>

    <input type="submit" value="Створити контракт">
</form>
<footer></footer>
<script src="scripts/jquery-3.3.1.min.js "></script>
<script src="scripts/owl.carousel.min.js "></script>
<script src="scripts/script.js "></script>
</body>
</html>

```

Auth.php

```

<?php
session_start();
?>

```

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="style/styles.css">
  <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" integrity="sha384-
fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr" crossorigin="anonymous">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
  integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOMLASjC"
  crossorigin="anonymous">
  <link href="https://cdn.jsdelivr.net/npm/select2@4.1.0-rc.0/dist/css/select2.min.css" rel="stylesheet" />
  <title>Авторизація</title>
  <link rel="icon" type="image/png" href="img/logo.png">
</head>

<body>
<?php
  $link = mysqli_connect("localhost", "root", "", "contracts");
?>
<div class="navbar-brand">
  <a href="index.php">
    <h1 class="navbar-heading">Smart Contracts</h1>
  </a>
</div>
<div>
<div mt="150px" class="container">
  <div class="row" style="margin-bottom: 20%;">
    <div class="col-3"></div>
    <div class="col-6">
      <h1><center>Будь ласка, введіть свій логін та пароль</center></h1><br>

      <form action="" method="POST" >
        <div class="box-body">

          <div class="step_1">
            <div class="form-group">
              <label for="First_name">Логін:</label>
              <input type="text" class="form-control first_name" name="name" required>
            </div><br>

            <div class="form-group">
              <label for="subject">Пароль:</label>
              <input class="form-control subject" required type="text" name="position" >
            </div><br>

            <div class="box-footer">
              <button type="submit" style="width: 100%;" name="submit" value="submit" class="btn btn-
danger">Увійти</button>
              <button type="button" style="width: 100%; margin-top: 2%; background-color: #1b9b4c;"
onclick="window.location.href = 'registration.php';" class="btn btn-danger">Зареєструватися</button>
            </div>
          </div>
        </div>
<?php

```



```

// Розділення логіну та статусу
list($login, $status) = explode('-', $user);

echo '<li class="contract-users-item">';
echo ' <span>' . $login . '</span>';
$image_src = ($status == 'signed') ? 'green.png' : 'red.png';
// Виведення картинки з відповідним посиланням
echo ' <img style="width: 4%;" src="" . $image_src . "" alt="" . $status . "">';
echo '</li>';
}
?>
</ul>
</div>

<div id="contractSigningContainer" style="margin-top: 4%;">
<?php
// Перевірка, чи є поточний користувач учасником контракту та його статус
$sql = "SELECT users FROM contracts WHERE id = $id";
$result = mysqli_query($link, $sql);
$row = mysqli_fetch_assoc($result);
$users_string = $row['users'];
$check = $userlogin . '-signed';
// Перевірка, чи поточний користувач підписав контракт
$user_signed_contract = (strpos($users_string, "$check") !== false) ? 1 : 0;
// Виведення відповідного блоку HTML
if ($user_signed_contract) {
    // Якщо користувач підписав контракт
    echo '<div class="contract-signed">';
    echo ' <h3 style="text-align: center;">Ви вже підписали цей контракт</h3>';
    echo ' <form action="modify_signature.php" method="post">';
    echo ' <input type="text" name="digital_key" style="width: 40%; margin-left: 30%; margin-top: 2%; margin-
bottom: 2%;" placeholder="Введіть ваш цифровий ключ">';
    echo ' <input type="hidden" name="contract_id" value="" . $id . "">';
    echo ' <br>';
    echo ' <button class="btn btn-warning" style="width: 40%; margin-left: 30%;" type="submit"
name="revoke_signature">Відкликати підпис</button>';
    echo ' </form>';
    echo '</div>';
} else {
    // Якщо користувач не підписав контракт
    echo '<div class="contract-not-signed">';
    echo ' <h3 style="text-align: center;">Підпишіть цей контракт, якщо згодні з його умовами</h3>';
    echo ' <form action="modify_signature.php" method="post">';
    echo ' <input type="text" name="digital_key" style="width: 40%; margin-left: 30%; margin-top: 2%; margin-
bottom: 2%;" placeholder="Введіть ваш цифровий ключ">';
    echo ' <input type="hidden" name="contract_id" value="" . $id . "">';
    echo ' <br>'; // Додаємо роздільну лінію
    echo ' <button class="btn btn-warning" style="width: 40%; margin-left: 30%;" type="submit"
name="sign_contract">Підписати контракт</button>';
    echo ' </form>';
    echo '</div>';
}
?>
</div>
</div>
</div>

<script src="scripts/jquery-3.3.1.min.js "></script>
<script src="scripts/script.js "></script>

```



```

</body>

</html>
<script>
    // Перевіряємо, чи є повідомлення у сесії
    var message = "<?php echo isset($_SESSION['message']) ? $_SESSION['message'] : ''; ?>";
    if (message !== '') {
        // Виводимо повідомлення в спливаючому вікні
        alert(message);
        // Очищуємо повідомлення з сесії
        <?php unset($_SESSION['message']); ?>
    }
</script>

```

Index.php

```

<?php
session_start();
if($_SESSION['itlogin'] == ""){
    header('Location: /auth.php');
}else{
    $login = $_SESSION['itlogin'];
}

?>

<!DOCTYPE html>
<html lang="en">
<?php
ini_set('display_errors', 0);
ini_set('display_startup_errors', 0);
error_reporting(E_ALL);
?>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="style/styles.css">
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css"
        integrity="sha384-fnmOCqbTlWIlj8LyTjo7mOUStjsKC4pOpQbqyi7RrhN7udi9RwhKkMHpvLbHG9Sr"
        crossorigin="anonymous">
    <title>Smart Contracts</title>
    <link rel="icon" type="image/png" href="img/logo.png">
</head>

<body>
    <?php
    $link = mysqli_connect("localhost", "root", "", "contracts");
    $sql = "SELECT * FROM contracts";
    ?>
    <header></header>
    <div id="home-section-1" class="movie-show-container">
        <h1>Ваші поточні смарт - контракти</h1>
        <h3>Будьласка, оберіть смарт - контракт для перегляду інформації щодо нього</h3>

        <div class="movies-container">
            <?php
            if($result = mysqli_query($link, $sql)){
                $lines = mysqli_num_rows($result);

```

```

        if(mysqli_num_rows($result) > 0){
            for ($i = 0; $i <= $lines - 1; $i++){
                $row = mysqli_fetch_array($result);
                echo '<div class="movie-box">';
                echo '';
                echo '<div class="movie-info ">';
                echo '<h3>'. $row['title'] .'/h3>';
                echo '<a href="contract.php?id='.$row['id'].'"><i class="fas fa-ticket-alt"></i> Перейти </a>';
                echo '</div>';
                echo '</div>';
            }
            mysqli_free_result($result);
        } else{
            echo '<h4 class="no-annot">На даний момент немає актуальних смарт - контрактів</h4>';
        }
    } else{
        echo "ERROR: Could not able to execute $sql. " . mysqli_error($link);
    }

    // Close connection
    mysqli_close($link);
    ?>

</div>
</div>
<div id="home-section-2" class="services-section">
    <h1>Чому ми ?</h1>
    <h3>Деякі плюси використання системи !</h3>

    <div class="services-container">
        <div class="service-item">
            <div class="service-item-icon">
                <i class="fas fa-4x fa-video"></i>
            </div>
            <h2>1. Автоматизація</h2>
            <p>Смарт-контракти дозволяють автоматизувати виконання угодних умов без потреби посередників або третіх сторін</p>
        </div>
        <div class="service-item">
            <div class="service-item-icon">
                <i class="fas fa-4x fa-credit-card"></i>
            </div>
            <h2>2. Безкоштовність</h2>
            <p>Використання нашого сервісу повністю безкоштовне !</p>
        </div>
        <div class="service-item">
            <div class="service-item-icon">
                <i class="fas fa-4x fa-theater-masks"></i>
            </div>
            <h2>3. Час та витрати</h2>
            <p>Використання смарт-контрактів може значно скоротити час та витрати, пов'язані з укладенням та виконанням угод</p>
        </div>
        <div class="service-item"></div>

    </div>
</div>
<div id="home-section-3" class="trailers-section">
    <h1 class="section-title"> Не знаєте які переваги надасть вам використання смарт - контрактів ?</h1>
    <h3>Будьласка, ознайомтеся з наведеними описами переваг використання !</h3>

```

```
<div class="trailers-grid">
    <div class="trailers-grid-item">
        
        <div class="trailer-item-info" data-video="ZE2HxTmxfrl?si=N-iXuaoCp3dQCK--">
            <h3>Smart Contracts</h3>
            <i class="far fa-3x fa-play-circle"></i>
        </div>
    </div>
    <div class="trailers-grid-item">
        
        <div class="trailer-item-info" data-video="pyalppMhuic?si=5MYeDu1Yz-kvF89w">
            <h3>Smart Contracts in Crypto</h3>
            <i class="far fa-3x fa-play-circle"></i>
        </div>
    </div>
    <div class="trailers-grid-item">
        
        <div class="trailer-item-info" data-video="kekx8QbEYHE?si=62Nky0xghmTLm319">
            <h3>Smart Contracts All About</h3>
            <i class="far fa-3x fa-play-circle"></i>
        </div>
    </div>
    <div class="trailers-grid-item">
        
        <div class="trailer-item-info" data-video="9PSjgELvQTo?si=TSQAefFs2L4pDVxp">
            <h3>Smart Contracts</h3>
            <i class="far fa-3x fa-play-circle"></i>
        </div>
    </div>
    <div class="trailers-grid-item">
        
        <div class="trailer-item-info" data-video="KO9mdk8CSpo?si=iUd4YvTpXEnfLv1Z">
            <h3>What is Smart Contracts</h3>
            <i class="far fa-3x fa-play-circle"></i>
        </div>
    </div>
    <div class="trailers-grid-item">
        
        <div class="trailer-item-info" data-video="4StueSP_2sw?si=QGajDS5ByBQhsxyk">
            <h3>How to Clone a Smart Contract</h3>
            <i class="far fa-3x fa-play-circle"></i>
        </div>
    </div>
</div>
</div>
<footer></footer>

<script src="scripts/jquery-3.3.1.min.js "></script>
<script src="scripts/script.js "></script>
</body>

</html>
```

```

    header('Location: /auth.php');
    exit();
} else {
    $login = $_SESSION['itlogin'];
}

$link = mysqli_connect("localhost", "root", "", "contracts");

// Перевірка з'єднання
if ($link === false) {
    die("Помилка: неможливо підключитись. " . mysqli_connect_error());
}

// Перевірка, чи була відправлена форма
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Отримання значень з форми
    $contract_id = $_POST['contract_id'];
    $digital_key = $_POST['digital_key'];

    // Отримуємо логін користувача з сесії
    $login = $_SESSION['itlogin'];

    // Отримуємо значення ключа користувача з бази даних
    $sql = "SELECT sign FROM users WHERE login = '$login'";
    $result = mysqli_query($link, $sql);
    $row = mysqli_fetch_assoc($result);
    $user_sign = $row['sign'];

    // Перевірка, чи ключі співпадають
    if ($user_sign == $digital_key) {
        // Значення ключа співпадає - змінюємо статус підпису в базі даних
        $sql = "SELECT users FROM contracts WHERE id = '$contract_id'";
        $result = mysqli_query($link, $sql);
        $row = mysqli_fetch_assoc($result);
        $users_data = $row['users'];

        // Розділяємо дані на масив користувачів та їх статусів
        $users_array = explode(',', $users_data);
        $user_signed_contract = false;
        foreach ($users_array as $user) {
            // Розділення логіну та статусу
            list($user_login, $status) = explode('-', $user);

            // Перевірка, чи поточний користувач підписав контракт
            if ($user_login == $login && $status == 'signed') {
                $users_data = str_replace("$login-signed", "$login-nope", $users_data);
                break;
            } else {
                $users_data = str_replace("$login-nope", "$login-signed", $users_data);
            }
        }
    }

    // Оновлюємо дані в таблиці "contracts"
    $update_sql = "UPDATE contracts SET users = '$users_data' WHERE id = '$contract_id'";
    mysqli_query($link, $update_sql);
}

```

```

$sql = "SELECT users FROM contracts WHERE id = $contract_id";
$result = mysqli_query($link, $sql);
$row = mysqli_fetch_assoc($result);
$users_string = $row['users'];

$all_signed = true;
$users_array = explode(',', $users_string);
foreach ($users_array as $user) {
    list($user_login, $status) = explode('-', $user);
    if ($status !== 'signed') {
        $all_signed = false;
        break;
    }
}

if ($all_signed) {
    // Змінюємо значення стовбця status на 'activated'
    $update_contract_sql = "UPDATE contracts SET status = 'activated' WHERE id = $contract_id";
    mysqli_query($link, $update_contract_sql);
    $link_sql = "SELECT link FROM contracts WHERE id = $contract_id";
    $result = mysqli_query($link, $link_sql);
    $row = mysqli_fetch_assoc($result);
    $redirect_link = $row['link'];

    // активація контракту
    header("Location: $redirect_link");
    exit;
}

$_SESSION['message'] = "Статус підпису змінено успішно.";
} else {
    // Повідомлення про невірний ключ
    $_SESSION['message'] = "Невірний цифровий ключ. Будь ласка, перевірте його та спробуйте знову.";
}

// Перенаправлення на попередню сторінку
header("Location: " . $_SERVER['HTTP_REFERER']);
exit();
}

// Закриття з'єднання з базою даних
mysqli_close($link);
?>

```

ДОДАТОК В. Ілюстративний матеріал

«Розробка програми групового цифрового підписування смарт контрактів»

Виконав
Гончарук О.М.
ІКІТС-20Б



Керівник
Яремчук Ю.Є.
д.т.н., професор

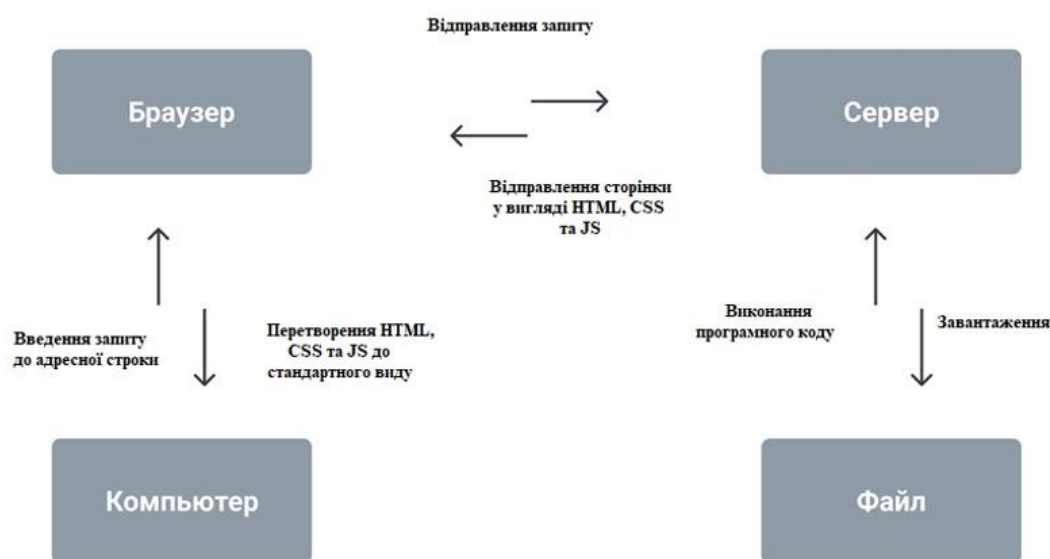
Мета та об'єкт дослідження

- **Метою роботи** є розробка інформаційної системи для автоматизації процесів групового цифрового підписування смарт контрактів.
- **Об'єктом дослідження** є інформаційна система групового цифрового підписування смарт контрактів.
- **Актуальність розробки** проектованої системи визначається великим попитом на подібні системи, оскільки автоматизація процесів робить процеси підписання контрактів більш гнучкими, зрозумілими, відкритими та стандартизованими, а також надає значні переваги : мобільність, безконтактність, ціна, масштабованість і т.ін.

Постановка задачі

- Провести аналіз предметної області для вибору найкращих інструментів для вирішення поставленого завдання
- Провести аналіз необхідних матеріалів для роботи з мовами програмування
- Провести аналіз необхідних матеріалів для роботи з базами даних
- Обрати архітектуру проектованою інформаційної системи
- Провести аналіз предметної області
- Виробити специфікацію вимог проектованої інформаційної системи
- Провести програмну реалізацію інтерфейсу користувача системи
- Провести програмну реалізацію бекенд частини системи для взаємодії з базою даних
- Провести тестування розробленої системи
- Провести аналіз результатів роботи

СХЕМА РОБОТИ ПРОЕКТОВАНОЇ СИСТЕМИ



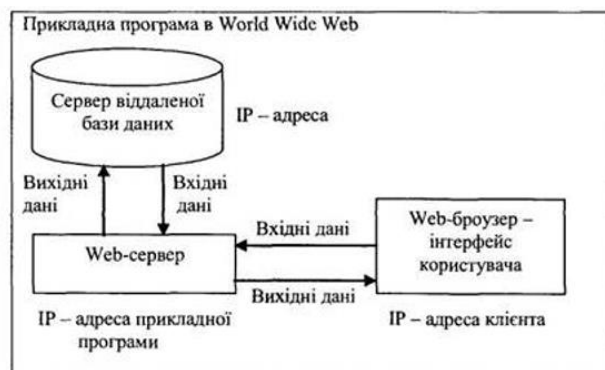
ВИКОРИСТАНІ ДЛЯ РОЗРОБКИ ПРОГРАМНІ ЗАСОБИ



ЗАСОБИ, ЩО ВИКОРИСТОВУЮТЬСЯ ДЛЯ РОБОТИ З БАЗОЮ ДАНИХ



СХЕМА ВЗАЄМОДІЇ ПРОЕКТОВАНОЇ СИСТЕМИ З БАЗОЮ ДАНИХ



РЕАЛІЗАЦІЯ ШИФРУВАННЯ ДАНИХ

```

function encrypt_decrypt($string, $action = 'encrypt')
{
    $encrypt_method = "AES-256-CBC";
    $secret_key = 'diplomaproject';
    $secret_iv = 'diplomaproject';
    $key = hash('sha256', $secret_key);
    $iv = substr(hash('sha256', $secret_iv), 0, 16); // алгоритм шифрування
    if ($action == 'encrypt') {
        $output = openssl_encrypt($string, $encrypt_method, $key, 0, $iv);
        $output = base64_encode($output);
    } else if ($action == 'decrypt') {
        $output = openssl_decrypt(base64_decode($string), $encrypt_method,
        $key, 0, $iv);
    }
    return $output;
}

```

СТРУКТУРА РЕАЛІЗОВАНОЇ БАЗИ ДАНИХ

Об'єкт	Власність	Тип	Розмір	Ідентифікатор
Таблиця смарт-контрактів	Ідентифікатор	Integer	100	Id
	Зображення	Varchar	100	Img
	Назва	Varchar	100	Title
	Опис	Varchar	100	Descr
	Посилання на активацію	Varchar	100	Link
	Функція активації	Varchar	100	Activation
	Власник контракту	Varchar	100	Creator
	Користувачі, що приймають участь у підписанні контракту	Varchar	100	Users
Таблиця відгуків	Статус контракту	Varchar	100	Status
	Ідентифікатор	Integer	100	msgID
	Ім'я відправника	Varchar	100	sendername
	Прізвище відправника	Varchar	100	senderlname
	Email відправника	Varchar	100	sendereMail
	Відгук	Varchar	500	Senderfeedback

Таблиця користувачів	Ідентифікатор	Integer	100	Id
	Логін	Varchar	100	Login
	Ім'я	Varchar	100	Name
	Прізвище	Varchar	100	Lname
	Пароль	Varchar	100	Password
	Країна	Varchar	100	Country
	Цифровий код	Varchar	100	Sign

ІНТЕРФЕЙС АВТОРИЗАЦІЇ У СИСТЕМІ

Smart Contracts

Будь ласка, введіть свій логін та пароль

Логін:

Пароль:

Соціальні Мережі



З'являйтесь з нами

З'являйтесь з нами, щоб отримати нові проекти, фінанси, курси, нові проекти, курси, проекти.

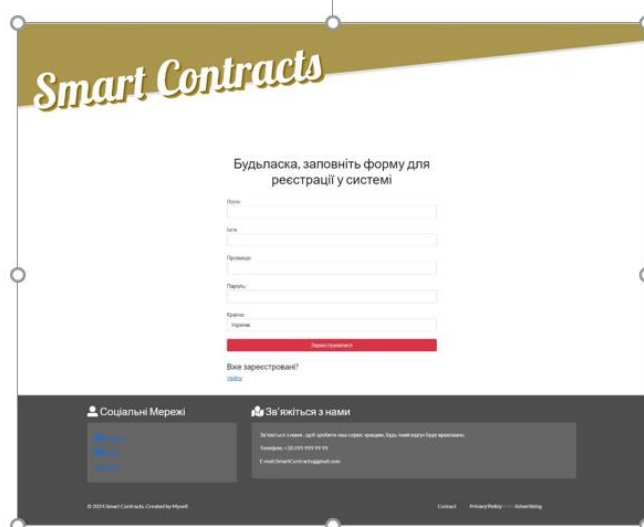
Телефон: +380 96 123 45 67 89

E-mail: smartcontracts@ukr.net

© 2017 Smart Contracts. Created by Hippi.

Contact Privacy Policy About Us

ІНТЕРФЕЙС РЕЄСТРАЦІЇ У СИСТЕМІ



Smart Contracts

Будьласка, заповніть форму для реєстрації у системі

Ім'я:

Пісь:

Прізвище:

Пароль:

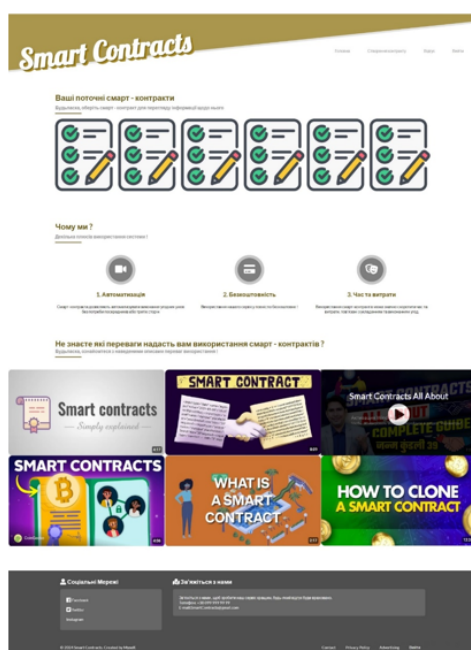
Електронна пошта:

Вже зареєстровані? [Вхід](#)

Соціальні Мережі: [Facebook](#), [Twitter](#), [LinkedIn](#)

З'яжіться з нами: [Telegram](#), [WhatsApp](#), [Viber](#)

© 2020 Smart Contracts. Created by Hubert



Smart Contracts

Ваш поточний смарт - контракти

Чому ми?

1. Автоматизація 2. Безпечність 3. Час та витрати

Не знаєте які переваги надасть вам використання смарт - контрактів?

Smart contracts - Simply explained

SMART CONTRACT

Smart Contracts All About

SMART CONTRACTS

WHAT IS A SMART CONTRACT

HOW TO CLONE A SMART CONTRACT

Соціальні Мережі: [Facebook](#), [Twitter](#), [LinkedIn](#)

З'яжіться з нами: [Telegram](#), [WhatsApp](#), [Viber](#)

© 2020 Smart Contracts. Created by Hubert

ІНТЕРФЕЙС ГОЛОВНОЇ СТОРІНКИ СИСТЕМИ

ІНТЕРФЕЙС ВЗАЄМОДІЇ ЗІ СМАРТ КОНТРАКТОМ

Контракт: Розподіл прибутків від спільного інвестування

Цей смарт контракт реалізує механізм автоматичного розподілу прибутків між учасниками спільного інвестування. Контракт збирає внески від усіх інвесторів і автоматично розподіляє отриманий прибуток на основі внесених сум.

Подія при активації контракту:

Активация відбувається при підписанні всіма учасниками контракту, що означає готовність до спільного інвестування. При активації контракту, система автоматично починає приймати інвестиції і коли сума інвестицій досягне певного рівня, починається розподіл прибутку.

Власник контракту:

user1

Статус контракту:

В очікуванні активації

Статус підписів учасників контракту:

user1	X
user1	●
user1	●
user1	X
user1	●

Поділіться цим контрактом, якщо згодні з його умовами

[Поділитися контрактом](#)

ІНТЕРФЕЙС СТВОРЕННЯ НОВОГО СМАРТ - КОНТРАКТУ

Smart Contracts

Головна | Створення контракту | Вартість | Баланс

Створення нового smart - контракту

Новий контракт:

Опис контракту:

Подія при активації:

Функція активації:

Учасники контракту:

user1
user11
user12

[Створити контракт](#)

Співпраця з Майданом

З'ясувати більше про нас: [Facebook](#), [Twitter](#), [Instagram](#)

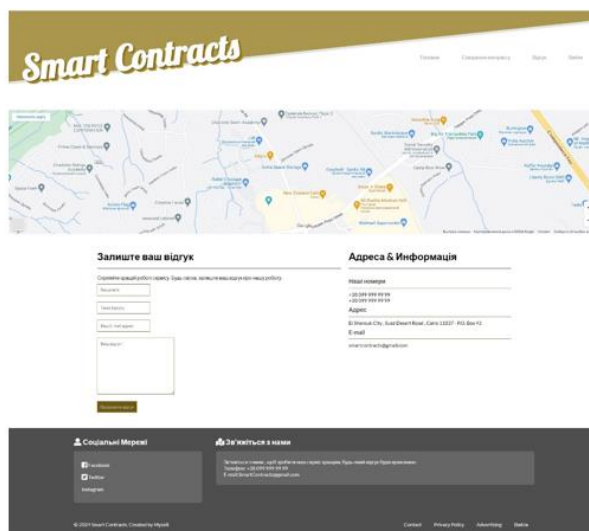
З'ясувати більше про нас: [Telegram](#), [YouTube](#), [Vimeo](#), [SoundCloud](#), [Spotify](#), [Bandcamp](#), [SoundCloud](#), [YouTube](#), [Vimeo](#), [SoundCloud](#), [Spotify](#), [Bandcamp](#)

Телефон: +380 999 999 99 99
E-mail: SmartContracts@gmail.com

© 2024 Smart Contracts. Created by Mynt.

[Contact](#) | [Privacy Policy](#) | [Advertising](#) | [Index](#)

ІНТЕРФЕЙС ВІДГУКІВ ЩОДО ВИКОРИСТАННЯ СИСТЕМИ



Висновки

- Під час виконання дипломної роботи була проведена робота з вивчення технологій розробки веб – орієнтованих систем з використанням технологій PHP, MySQL, HTML, CSS. За підсумками виконання роботи було розроблено інформаційну систему для автоматизації процесів групового цифрового підписування смарт контрактів.
- Реалізовано функції користувача, що забезпечить можливостями перегляду основної інформації щодо смарт - контрактів, що актуальні для поточного користувача, взаємодії з поточними контрактами, цифрового підпису обраного контракту, створення нових смарт – контрактів, написання відгуків, надсилання заявок на співпрацю.
- Створена система дозволяє користувачу використовувати функціонал додатку за допомогою інтерфейсу користувача, та керувати вмістом додатку та бази даних, змінювати, видаляти та додавати інформацію щодо поточних смарт - контрактів. При створенні системи були враховані всі плюси та мінуси систем-аналогів, вибраних для порівняння.
- Проведено всі необхідні аналітичні процеси для проектування та програмування системи та бази даних. Після реалізації системи обраними мовами програмування та середовищах, було проведено випробування на працездатність, яке засвідчило, що система виконує усі поставлені завданням функції.
- Оскільки порушень не виявлено, всі функції відповідають вимогам, тому можна визначити, що реалізована система готова для впровадження, та може бути використана для автоматизації процесів групового цифрового підписування смарт контрактів.

ДОДАТОК Г. Протокол перевірки на антиплагіат

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка програми групового цифрового підписування смарт контрактів

Тип роботи: бакалаврська дипломна робота
(БДР, МКР)

Підрозділ: Кафедра менеджменту та безпеки інформаційних систем
Факультет менеджменту та інформаційної безпеки
(кафедра, факультет)

Показники звіту подібності Unichesk

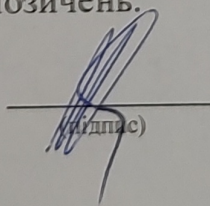
Оригінальність 76 %

Схожість 24 %

Аналіз звіту подібності (відмітити потрібне):

1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку

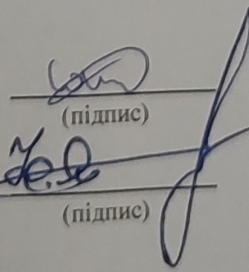


(підпис)

Коваль Н.П.
(прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unichesk щодо роботи.

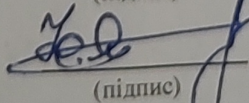
Автор роботи



(підпис)

Гончарук О.М.
(прізвище, ініціали)

Керівник роботи



(підпис)

Яремчук Ю.Є.
(прізвище, ініціали)