Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету )
Кафедра кібербезпеки
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(освітній рівень)

на тему: "Розробка децентралізованої системи електронного голосування основі блокчейну"

Виконав: студент (ка)

Спеціальності:
125 «Кібербезпека та захист інформації»
(шифр і назва напряму підготовки, спеціальності)
Ратишин Микола Олегович
підпис (прізвище та ініціали)

Керівник Загородна Н.В.
підпис (прізвище та ініціали)

Нормоконтроль Тимощук Д. І.
підпис (прізвище та ініціали)

Завідувач кафедри Загородна Н.В.
підпис (прізвище та ініціали)

Рецензент
підпис (прізвище та ініціали)

м. Тернопіль – 2024

Міністерство освіти і науки України
**Тернопільський національний технічний університет імені Івана Пулюя**

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2024 р.

# З А В Д А Н Н Я
# НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека та захист інформації
(шифр і назва спеціальності)

студенту Ратишину Миколі Олеговичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка децентралізованої системи електронного голосування на основі блокчейну

Керівник роботи Загородна Наталія Володимирівна, к.т.н.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «22» 11 2024 року № 4/7-1119

2. Термін подання студентом завершеної роботи 25.12.2024

3. Вихідні дані до роботи вимоги до функціоналу системи

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ. Розділ 1. Аналітичний огляд в області дослідження.
Розділ 2. Теоретична частина.
Розділ 3. Практична частина.
Розділ 4. Охорона праці та безпека в надзвичайних ситуаціях
Висновки. Перелік джерел. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
Титульна сторінка. Актуальність, мета, предмет дослідження. Задачі. Існуючі рішення та їх недоліки. Наукова новизна. Архітектура системи. Діаграми послідовності. Реалізація смарт-контрактів. Результати тестування. Висновки.

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| Охорона праці | Осухівська Г.М., к.т.н., доцент | | |
| Безпека в надзвичайних ситуаціях | Клепчик В.М., проректор з адміністрати господарської роботи та будівництва | | |

7. Дата видачі завдання 23.09.2024 р

# КАЛЕНДАРНИЙ ПЛАН

| | Назва етапів роботи | Термін виконання етапів роботи | Примітка |
|---|---|---|---|
| 1 | Ознайомлення з завданням до кваліфікаційної роботи | 23.09 – 25.09 | Виконано |
| 2 | Аналіз предметної області | 21.11 – 25.11 | Виконано |
| 3 | Огляд існуючих методів | 26.11 – 29.11 | Виконано |
| 4 | Основна частина | 30.11 – 03.12 | Виконано |
| 5 | Виконання завдання до підрозділу «Охорона праці та безпека в надзвичайних ситуаціях» | 04.12 – 08.12 | Виконано |
| 6 | Оформлення кваліфікаційної роботи | 01.12 – 08.12 | Виконано |
| 7 | Нормоконтроль | 09.12 – 11.12 | Виконано |
| 8 | Перевірка на плагіат | 12.12 – 15.12 | Виконано |
| 9 | Попередній захист кваліфікаційної роботи | 16.12 – 20.12 | Виконано |
| 10 | Захист кваліфікаційної роботи | 27.12.2024 | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Студент _____ Ратишин М.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Загородна Н.В.
(підпис) (прізвище та ініціали)

# АНОТАЦІЯ

Розробка децентралізованої системи електронного голосування на основі блокчейну // Кваліфікаційна робота освітнього рівня “Магістр” // Ратишин Микола Олегович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-62, 2024 // С. 64, рис. – 30, табл. – 2, додат. – 1, бібліогр. – 30.

Ключові слова: блокчейн, електронне голосування, смарт-контракти, децентралізація, кібербезпека, Ethereum, Web3j, система голосування.

Кваліфікаційна робота присвячена дослідженню, розробці та впровадженню децентралізованої системи електронного голосування на основі блокчейну.

У першому розділі роботи проведено аналітичний огляд сучасних рішень у сфері електронного голосування, обґрунтовано актуальність теми, визначено мету та задачі дослідження, об'єкт і предмет дослідження, а також розглянуто базові аспекти використання блокчейну для голосування.

У другому розділі висвітлено теоретичні аспекти, що включають дослідження криптографічних механізмів для забезпечення безпеки голосування, розробку архітектури системи, а також опис основних технологій і інструментів, які використовуються для реалізації децентралізованих додатків.

У третьому розділі розглянуто процес реалізації системи, включаючи розробку смарт-контрактів та клієнт-серверну архітектуру. Проведено тестування роботи системи, зокрема оцінено функціональні можливості додатку, ідентифіковано потенційні загрози та запропоновано методи їх мінімізації.

Об’єкт дослідження: Процес організації електронного голосування з використанням сучасних технологій.

Предмет дослідження: Методи та засоби реалізації децентралізованих систем голосування з використанням технології блокчейн.

# ABSTRACT

Development of a Decentralized Electronic Voting System Based on Blockchain // Master's Qualification Work // Ratyshyn Mykola Olehovych// Ternopil Ivan Puluj National Technical University, Faculty of Computer-Information Systems and Software Engineering, Department of Cybersecurity, Group SBm-62, 2024 // P. 64, fig. – 30, tables – 2, added. – 1, references – 30.

Keywords: blockchain, electronic voting, smart contracts, decentralization, cybersecurity, Ethereum, Web3j, voting system.

This master's qualification work is dedicated to the research, development, and implementation of a decentralized electronic voting system based on blockchain technology.

The first chapter provides an analytical review of modern solutions in the field of electronic voting, substantiates the relevance of the topic, defines the goals and objectives of the research, the object and subject of study, and considers the fundamental aspects of blockchain application for voting.

The second chapter highlights theoretical aspects, including the exploration of cryptographic mechanisms to ensure voting security, the development of the system architecture, and a description of the main technologies and tools used for the implementation of decentralized applications.

The third chapter examines the system's implementation process, including the development of smart contracts and the client-server architecture. The system's functionality is tested, potential threats are identified, and methods to mitigate them are proposed.

Object of the research: The process of organizing electronic voting using modern technologies.

Subject of the research: Methods and tools for implementing decentralized voting systems using blockchain technology

ЗМІСТ

# ВСТУП

Актуальність теми. Розробка децентралізованих систем електронного голосування на основі блокчейну є надзвичайно актуальною в умовах глобальної цифровізації суспільства, коли виборчі процеси потребують забезпечення прозорості, безпеки та довіри з боку громадян. Використання традиційних підходів до організації виборів стикається з проблемами, пов'язаними з маніпуляціями, недостатньою прозорістю процесів та обмеженнями щодо географічного охоплення. Технологія блокчейн відкриває нові можливості для модернізації виборчих процесів, забезпечуючи високий рівень безпеки, надійності та довіри завдяки своїй децентралізованій природі.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи є розробка децентралізованої системи електронного голосування на основі блокчейну, яка забезпечує прозорість, безпеку та довіру виборців до процесу голосування.

Завданнями дослідження є:

1. Аналіз існуючих підходів до побудови систем електронного голосування.
2. Дослідження можливостей технології блокчейн у контексті виборчих систем.
3. Розробка архітектури децентралізованої системи голосування.
4. Реалізація смарт-контрактів для забезпечення безпеки та прозорості виборчого процесу.
5. Розробка клієнт-серверної системи для взаємодії з блокчейн-мережею.
6. Тестування та верифікація функціональності розробленої системи.

Об’єкт дослідження. Процес організації електронного голосування з використанням сучасних технологій.

Предмет дослідження. Методи та засоби реалізації децентралізованих систем голосування з використанням технології блокчейн.

Наукова новизна одержаних результатів кваліфікаційної роботи. Новизна полягає у розробці архітектурного рішення для децентралізованої системи

електронного голосування, що вперше використовує можливості блокчейн-мережі Ethereum для інтеграції смарт-контрактів, які забезпечують прозорість, безпеку та анонімність виборчого процесу, перевершуючи існуючі рішення за рівнем децентралізації, гнучкості та масштабованості. Ethereum забезпечує значно вищий рівень децентралізації, прозорості та гнучкості у порівнянні з PoA-блокчейнами, такими як Electis чи швейцарські рішення, а також перевершує вузькоспеціалізовані блокчейни на кшталт Voatz завдяки відкритій архітектурі та потужній екосистемі смарт-контрактів. Це робить Ethereum оптимальним вибором для масштабованих і захищених голосувань.

Практичне значення одержаних результатів. Розроблена система може бути використана для проведення виборів різного рівня, забезпечуючи прозорість і доступність голосування для виборців. Запропоновані підходи можуть бути інтегровані в існуючі інформаційні системи виборчих комісій або використані для проведення внутрішніх голосувань у корпоративних середовищах.

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на: XII науковій-технічній конференції «Інформаційні моделі, системи та технології» (м. Тернопіль, Україна).

Публікації. Основні результати кваліфікаційної роботи опубліковано у працях конференції (див. Додаток А).

# РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД В ОБЛАСТІ ДОСЛІДЖЕНЬ

## 1.1 Сучасний стан електронного голосування

Електронне голосування, як інноваційний спосіб забезпечення демократичного волевиявлення громадян, стає дедалі популярнішим у сучасному світі. Цей метод є результатом цифровізації суспільства та відповідає потребам сучасних людей, які прагнуть більшої зручності у повсякденних справах. Багато країн стикаються з проблемами, пов’язаними із традиційними методами голосування: бюрократією, складністю організації виборчих процесів, людським фактором у підрахунку голосів, а також відсутністю можливостей голосування для громадян, які фізично не можуть бути присутніми на дільницях. Усі ці виклики стають причинами зростання інтересу до електронного голосування, яке, на відміну від традиційних паперових систем, пропонує нові можливості та значні покращення.

Одна з основних причин зростання популярності електронного голосування полягає у його здатності значно скорочувати витрати на організацію виборчого процесу. Традиційне голосування передбачає великий комплекс заходів, починаючи від друку бюлетенів і закінчуючи логістикою, що вимагає величезних ресурсів. Додатково, у більшості випадків вибори потребують залучення великої кількості виборчої комісії для підрахунку голосів, що також створює ризик людських помилок і підвищує витрати. Електронне голосування, у свою чергу, дозволяє уникнути цих проблем, знижуючи витрати як на фізичні ресурси, так і на людську працю, що є важливим аспектом у сучасному економічно нестабільному середовищі.

Ще одна ключова перевага електронного голосування полягає в його здатності прискорити підрахунок голосів. Завдяки використанню цифрових технологій процес підрахунку стає майже миттєвим, а результати можуть бути відомі відразу після закриття виборчих дільниць. Це суттєво скорочує час на підбиття підсумків та зменшує ризик маніпуляцій з результатами в процесі

передачі даних, що є важливою перевагою над традиційними паперовими бюлетенями.

У контексті доступності електронне голосування надає нові можливості тим громадянам, які мають фізичні або географічні обмеження. Наприклад, громадяни, які проживають за кордоном, або ті, хто має обмежені фізичні можливості, можуть скористатися можливістю голосування онлайн. Це означає, що участь у виборах стає можливою для ширшого кола виборців, що сприяє зростанню рівня демократичної участі. У світі, де все більше процесів відбувається онлайн, доступ до демократичних процедур через інтернет є логічним кроком уперед.

На сьогодні існує кілька основних підходів до електронного голосування. Одним із найпоширеніших є голосування через інтернет, яке дозволяє громадянам здійснити вибір зі своїх пристроїв з будь-якого місця, де є доступ до мережі. Цей тип голосування успішно впроваджений, наприклад, в Естонії, яка стала першою країною, що використала інтернет-голосування на національному рівні. Для багатьох людей, які звикли здійснювати щоденні операції онлайн, включаючи покупки, банківські операції та навчання, можливість віддати свій голос через інтернет є не лише зручністю, але й необхідністю.

Інший підхід, що отримав широке застосування, — це використання електронних машин для голосування на виборчих дільницях. Цей метод забезпечує швидкість підрахунку голосів без ризику людської помилки. Але при цьому виникають інші проблеми, зокрема, необхідність фінансування дорогого обладнання та навчання персоналу, що буде обслуговувати ці пристрої. Тим не менш, електронні машини дозволяють уникнути маніпуляцій із бюлетенями, а також забезпечують повну автоматизацію процесу, що зменшує ризики, пов’язані з людським фактором.

Незважаючи на всі переваги, електронне голосування стикається з низкою проблем, які потребують вирішення. Першою і найбільш значущою є проблема безпеки. З одного боку, використання цифрових технологій створює можливість забезпечення аутентифікації виборців та захисту результатів, але, з іншого боку, існує ризик кібератак та зловживання. Виборчі системи є привабливою мішенню

для хакерів, і у разі успішного нападу можна легко змінити результати, що підриває легітимність усього виборчого процесу. Питання забезпечення безпеки в електронному голосуванні вимагає використання складних криптографічних алгоритмів і постійного моніторингу системи на предмет вразливостей.

Ще одним важливим питанням є довіра з боку громадян. Багато людей бояться, що їхні голоси можуть бути підроблені або змінені. Довіра до електронного голосування є одним з основних викликів, і урядам необхідно розробляти програми, що сприяють підвищенню прозорості цього процесу. Публічні огляди коду систем голосування, незалежний аудит і можливість для кожного виборця перевірити свій голос у системі — ось кроки, які можуть допомогти підвищити довіру до електронного голосування [1].

Також не варто забувати про можливі технічні збої. Виборчі системи повинні бути стійкими до збоїв, оскільки кожна технічна помилка може вплинути на результат. Втрата даних, помилки в програмному забезпеченні, збої в обладнанні — усе це може призвести до недійсних результатів. Таким чином, розробка системи електронного голосування вимагає ретельного тестування, а також забезпечення планів на випадок непередбачуваних ситуацій.

Проблема доступності електронного голосування також не може бути ігнорованою. Не всі громадяни мають доступ до інтернету або можуть використовувати сучасні електронні засоби. Для деяких верств населення, зокрема людей похилого віку або осіб, які живуть у віддалених регіонах, електронне голосування може стати додатковою перешкодою, а не полегшенням. Тому урядам необхідно розробити механізми забезпечення рівного доступу для всіх громадян.

## 1.2 Існуючі рішення у галузі електронного голосування

Електронне голосування є ключовою сферою для забезпечення сучасних демократичних процесів, особливо у світі, де цифрові технології все більше проникають у всі аспекти нашого життя. Різноманітні країни та організації вже впроваджують електронні системи голосування або проводять експерименти з

ними, використовуючи різні технологічні рішення для забезпечення безпеки, прозорості та надійності цього процесу. Сучасні рішення у сфері електронного голосування пропонують різні моделі, але всіх їх об'єднує спільне завдання — забезпечити максимально прозорий, безпечний і зручний для користувача процес виборів.

Одним з найбільш відомих прикладів системи електронного голосування є швейцарська платформа для онлайн-голосування, яку використовували для проведення місцевих виборів у кількох кантонах. Ця система дозволяє громадянам голосувати з дому, використовуючи захищене з'єднання через інтернет. Громадянам потрібно підтвердити свою особу через електронну ID-картку, що дає можливість забезпечити автентифікацію виборців і запобігти можливим шахрайствам. Крім цього, система включає механізм, який дозволяє перевірити, чи був голос врахований і чи не був він змінений, що підвищує рівень довіри до її використання.

Інший цікавий приклад — система Voatz, яку використовували для голосування під час праймеріз у штаті Західна Вірджинія у США. Ця система базується на блокчейн-технології, що робить її прозорою і незмінною. Виборці могли використовувати свої мобільні пристрої для голосування, що робило процес зручним і доступним для тих, хто не міг з'явитися на виборчі дільниці особисто. Впровадження блокчейну забезпечило прозорість підрахунку голосів, оскільки кожна транзакція записується в незмінний реєстр, який доступний для публічного перегляду (див. рисунок 1.1).


Рисунок 1.1 – Логотип додатку Voatz

Крім того, Естонія — одна з перших країн, яка впровадила повномасштабне електронне голосування на державному рівні. В Естонії громадяни можуть голосувати онлайн вже протягом понад 15 років, і ця система отримала позитивні відгуки як на національному, так і на міжнародному рівні. Ключовими елементами естонської системи є використання цифрового підпису та смарт-карт для ідентифікації виборців. Ці інструменти забезпечують надійний рівень автентифікації і захисту від шахрайства. Крім цього, система дозволяє виборцю змінити свій голос протягом голосування — якщо виборець вважає, що його голос може бути скомпрометовано, він може просто проголосувати ще раз, і система враховує лише останній голос.

Ще одним цікавим прикладом є система FollowMyVote, яка також використовує блокчейн для реалізації електронного голосування. FollowMyVote — це децентралізована платформа, яка дає можливість кожному виборцю анонімно голосувати і перевірити свій голос у загальному реєстрі. Відмінною рисою цієї системи є її відкритий вихідний код, що дозволяє будь-якому охочому перевірити алгоритми і переконатися у їхній надійності. Це дуже важливо для забезпечення довіри, адже прозорість процесу є критичним аспектом у виборчих технологіях (див. рисунок 1.2).


Рисунок 1.2 – Логотип додатку FollowMyVote

Існують також проекти, що використовують концепцію "гомоморфного шифрування" для забезпечення безпеки електронного голосування. Такі рішення забезпечують можливість підрахунку голосів без їх попереднього розшифрування. Це означає, що голоси залишаються зашифрованими під час усього процесу підрахунку, що значно знижує ризик несанкціонованого доступу до інформації про виборців. Гомоморфне шифрування забезпечує високий рівень конфіденційності, а також можливість перевірити цілісність даних після завершення голосування [2].

Однак, незважаючи на всі переваги, існуючі рішення мають і свої недоліки та обмеження. Наприклад, у деяких країнах, таких як Нідерланди та Німеччина, електронне голосування зазнало невдачі через проблеми з безпекою та брак довіри до електронних систем. У Нідерландах система голосування через інтернет була відхилена через підозри щодо можливих вразливостей у програмному забезпеченні, що могло б призвести до маніпуляцій з голосами. У Німеччині електронне голосування було заборонене після рішення Конституційного суду, який постановив, що процес голосування має бути "зрозумілим" для кожного виборця і не має вимагати спеціальних технічних знань.

Існуючі рішення у галузі електронного голосування свідчать про великий потенціал цієї технології, але одночасно і про потребу у вдосконаленні методів забезпечення безпеки, доступності та надійності. Усі перелічені системи намагаються знайти баланс між зручністю для виборців, безпекою та прозорістю процесу, але і досі залишаються виклики, які вимагають вирішення для повсюдного впровадження таких технологій.

## 1.3 Використання блокчейну у виборчих системах

Блокчейн — це одна з найбільш інноваційних технологій, що значно вплинула на різні галузі, включаючи фінанси, логістику, а тепер і виборчі процеси. Його використання у виборчих системах дає можливість створити децентралізовану, безпечну і прозору платформу для голосування, яка є стійкою до зовнішніх атак та внутрішніх маніпуляцій. У цьому підрозділі розглянемо, як блокчейн використовується у виборчих системах, які переваги він надає, та які виклики стоять перед його реалізацією.

Блокчейн забезпечує безпеку виборчого процесу завдяки своїй децентралізованій природі. У традиційних виборчих системах дані зберігаються у централізованих базах, які можуть бути уразливими до хакерських атак або внутрішніх маніпуляцій. У випадку з блокчейном інформація зберігається одночасно на багатьох вузлах мережі, що робить її набагато стійкішою до атак. Якщо зловмисник спробує змінити інформацію на одному вузлі, це не вплине на інші вузли, і система виявить спробу втручання (див. рисунок 1.3).

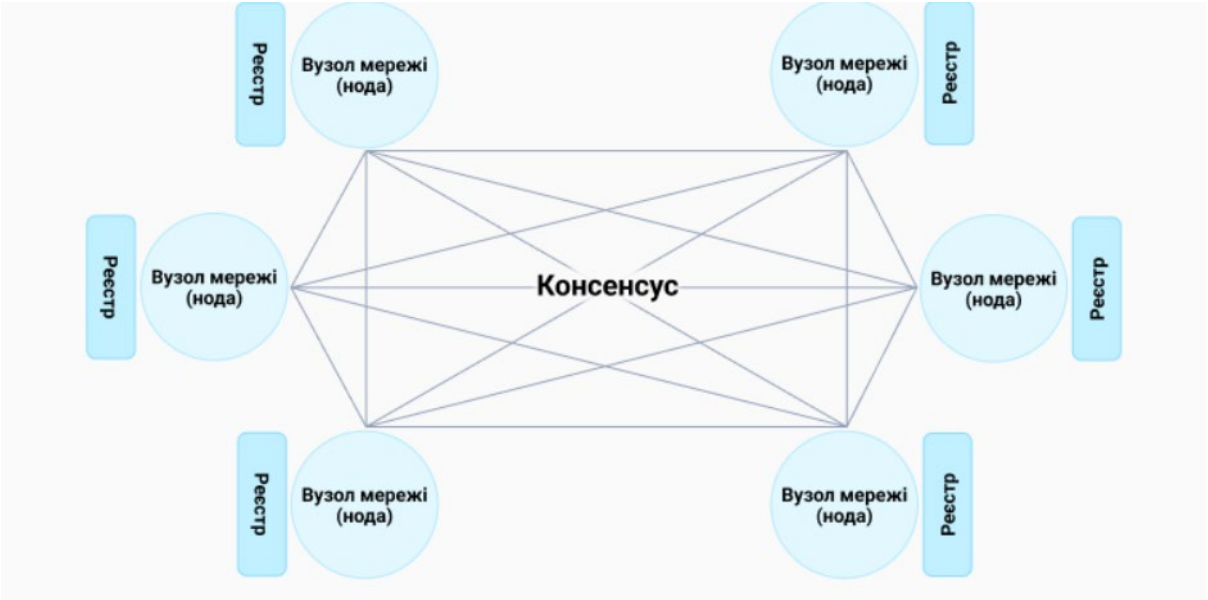


Рисунок 1.3 – Приклад загальної структури розподіленого реєстру

Одна з головних переваг використання блокчейну у виборчих системах — це незмінність записів. Кожен блок у ланцюжку містить хеш попереднього блоку, що створює безперервний ланцюжок, де будь-яка спроба змінити дані одного з блоків вимагала б перерахунку всіх наступних блоків у ланцюжку. Це забезпечує надзвичайно високий рівень безпеки. Голоси виборців, зареєстровані у блокчейні, залишаються незмінними, що гарантує їхню цілісність. Це дуже важливо у виборчому процесі, адже будь-які зміни у результатах голосування є неприпустимими.

У децентралізованій системі голосування кожен голос зберігається як окрема транзакція у ланцюжку блоків. Це дозволяє створити систему, у якій усі транзакції відкриті для перевірки, але при цьому залишаються анонімними. Кожен виборець може переконатися, що його голос був врахований, не розкриваючи своєї ідентичності. Це підвищує довіру до системи, оскільки виборці мають змогу перевірити результати голосування самостійно, що зменшує ризик шахрайства або маніпуляцій результатами.

Цікавою альтернативою є використання блокчейн-платформи для голосування, розробленої університетом Невшатель у Швейцарії. Ця платформа орієнтована на проведення виборів серед студентів і викладачів. Її основою є

консенсусний механізм Proof of Authority (PoA), що дозволяє швидше і ефективніше забезпечувати обробку транзакцій, зберігаючи при цьому високий рівень безпеки. Ключовою перевагою є швидкість, оскільки виборчий процес відбувається у межах обмеженого кола учасників із довіреними вузлами, що зменшує навантаження на мережу та знижує витрати на обробку даних.

Ще одним відомим проектом, який використовує блокчейн у виборчих процесах, є система Electis. Ця платформа дозволяє проводити голосування в організаціях, включаючи неурядові організації, університети та інші спільноти. Electis використовує протокол децентралізованого шифрування, який дає змогу забезпечити максимальну конфіденційність голосів та захист від підробки. Electis не потребує спеціального програмного забезпечення на стороні користувача — голосування може відбуватися через веб-браузер, що робить систему зручною для широкого кола користувачів (див. рисунок 1.4).


Рисунок 1.4 – Логотип компанії Electis

Ще однією технологією, яка активно розвивається у контексті блокчейн-голосування, є використання "обмеженої конфіденційності" (Zero-Knowledge Proofs, ZKP). Цей метод дає можливість перевірити, чи виконано певні умови (наприклад, чи був голос поданий одним з виборців), не розкриваючи жодної додаткової інформації про сам голос. Це дозволяє забезпечити максимальну конфіденційність даних і, водночас, надає можливість перевірки правильності результатів. Zero-Knowledge Proofs дають можливість системі зберігати прозорість і забезпечувати можливість перевірки, не розкриваючи особистих даних виборців.

Однак використання блокчейну у виборчих системах також стикається з певними викликами. По-перше, це питання масштабованості. У випадку масового

голосування, наприклад, на загальнонаціональних виборах, блокчейн може зіткнутися з проблемою обробки великої кількості транзакцій. Це може спричинити затримки та зменшення швидкості системи. Для подолання цієї проблеми деякі розробники використовують гібридні рішення, де блокчейн використовується лише для ключових етапів голосування, а інші етапи виконуються централізовано.

Ще однією проблемою є доступність технології для широкого кола виборців. У деяких країнах, де рівень технологічної освіти або доступу до інтернету є низьким, використання блокчейну для голосування може бути складним завданням. Виборці повинні мати доступ до необхідного обладнання (наприклад, смартфонів або комп'ютерів) та певний рівень знань, щоб використовувати систему голосування на блокчейні.

Крім того, важливим аспектом є проблема конфіденційності. Хоча блокчейн забезпечує анонімність виборців, він також створює незмінний реєстр всіх транзакцій. Це означає, що, хоча імена виборців не зберігаються, вся інформація про транзакції залишається у відкритому доступі, і за певних умов може існувати ризик ідентифікації виборців на основі аналізу патернів даних.

Таким чином, блокчейн відкриває значні можливості для модернізації виборчих процесів, забезпечуючи високий рівень безпеки, прозорості та стійкості до шахрайства. Однак його реалізація потребує врахування низки технологічних та організаційних викликів.

# РОЗДІЛ 2 ТЕОРЕТИЧНА ЧАСТИНА

## 2.1 Основні технології для реалізації децентралізованої системи голосування

Децентралізовані системи голосування ґрунтуються на комбінації передових технологій, що дозволяють забезпечити безпечність, прозорість і конфіденційність виборчого процесу. Основою таких систем є технологія блокчейн, яка дозволяє зберігати всі дані про голоси та виборців у вигляді розподіленої бази даних. Блокчейн забезпечує високий рівень безпеки і незмінність даних завдяки своїй децентралізованій природі. У контексті голосування блокчейн виступає платформою для реєстрації голосів, які фіксуються як транзакції. Це дозволяє уникнути необхідності централізованого органу, який контролює збереження і обробку даних, що значно знижує ризик фальсифікацій або маніпуляцій результатами.

Кожен голос записується як транзакція у блокчейні і додається до загального ланцюга блоків. Завдяки цьому створюється прозора та незмінна історія голосування, яку можуть перевірити всі учасники системи, але ніхто не може змінити вже записані дані. Вся інформація зберігається одночасно на багатьох вузлах мережі, що робить її менш вразливою до атак або зловживань. Така децентралізація дозволяє забезпечити високу стійкість системи навіть за умов можливих технічних збоїв або кібератак (див. рисунок 2.1).

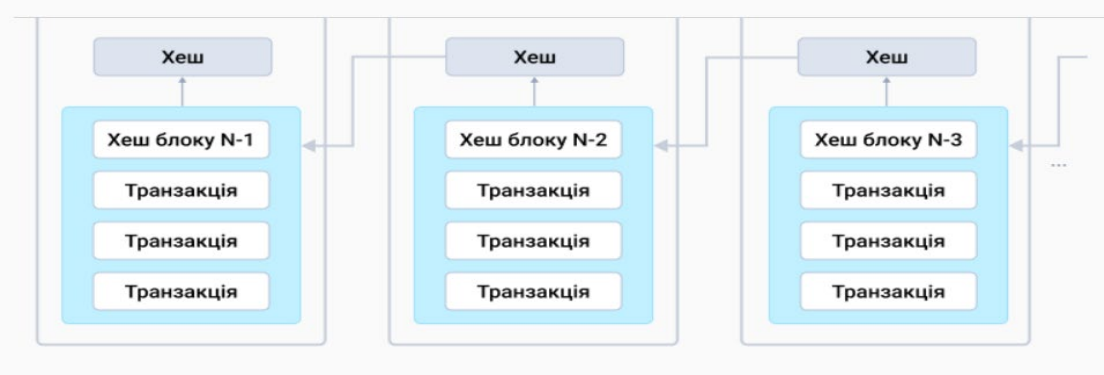


Рисунок 2.1 – Приклад загальної структури та організації блоків

Ще одним важливим компонентом децентралізованих систем голосування є смарт-контракти. Смарт-контракти — це програми, які виконуються автоматично після виконання певних умов, визначених у їхньому коді. У системах голосування смарт-контракти можуть бути відповідальними за реєстрацію голосів, перевірку правомочності виборців, обробку результатів тощо [3]. Основною перевагою смарт-контрактів є автономність і незалежність від третьої сторони. Після розгортання смарт-контракту у блокчейн, його неможливо змінити без внесення змін у сам блокчейн, що забезпечує високу ступінь надійності. Коли виборець відправляє свій голос, смарт-контракт автоматично перевіряє його правомочність, і якщо всі умови виконано, записує голос у блокчейн. Це дозволяє мінімізувати людський фактор і знизити ризик фальсифікацій.

Для забезпечення безпеки голосування використовуються різноманітні криптографічні методи. Хеш-функції, цифрові підписи та Zero-Knowledge Proofs (ZKP) є основними засобами для забезпечення конфіденційності та цілісності даних у таких системах. Хеш-функції створюють унікальні, але незворотні значення на основі вхідних даних, що дозволяє забезпечити цілісність інформації. Цифрові підписи використовуються для аутентифікації виборців, оскільки кожен виборець має свій приватний ключ для підписання голосу, і лише зареєстровані учасники можуть брати участь у голосуванні. Zero-Knowledge Proofs дозволяють виборцеві довести свою правомочність без розкриття додаткової інформації, забезпечуючи конфіденційність і захист персональних даних.

Розподілені обчислення є ще однією важливою складовою децентралізованих систем. Ці обчислення дозволяють обробляти транзакції одночасно на багатьох вузлах, а консенсусні механізми, такі як Proof of Stake або Proof of Authority, забезпечують узгодження даних між вузлами. Консенсус дозволяє всім вузлам мережі мати однакову версію реєстру, що виключає можливість маніпуляцій або помилок [4]. Це особливо важливо у виборчих системах, де всі вузли повинні мати доступ до однакових даних для забезпечення прозорості та коректності результатів.

Для інтеграції з користувачами використовуються фронтенд і бекенд компоненти. Фронтенд дозволяє користувачам взаємодіяти з системою через веб-

інтерфейс, а бекенд виконує функцію посередника між фронтендом і блокчейном. Бекенд зазвичай реалізується на основі серверних технологій, таких як Spring Boot, і використовує API для зв’язку з блокчейном. Це дозволяє користувачам отримувати дані про наявні голосування, реєструвати свої голоси або перевіряти статус голосування через веб-інтерфейс. Фронтенд відправляє запити до бекенду, який через смарт-контракти реєструє голоси у блокчейні.

Завдяки цим технологіям — блокчейну, смарт-контрактам, криптографічним методам захисту, розподіленим обчисленням, а також фронтенд і бекенд компонентам — децентралізовані системи голосування забезпечують високий рівень безпеки, прозорості та надійності. Ці технології дозволяють створити виборчу систему, яка мінімізує людський фактор, виключає можливість маніпуляцій та забезпечує конфіденційність голосування, що є надзвичайно важливим у сучасних умовах.

## 2.2 Блокчейн та консенсусні алгоритми

Блокчейн є однією з ключових технологій у реалізації децентралізованих систем електронного голосування, і його основною особливістю є можливість зберігати дані у вигляді незмінного реєстру. Цей реєстр складається з послідовних блоків, кожен з яких містить запис про транзакції, тобто дії, виконані в мережі. Кожен блок зв’язаний із попереднім через криптографічний хеш, що забезпечує цілісність і безпеку всього ланцюга. У контексті виборчих систем блокчейн виступає як механізм, що гарантує незмінність даних, забезпечуючи таким чином прозорість і надійність голосування.

Кожен голос у системі зберігається як транзакція у блокчейні, що дозволяє створити незмінну історію голосування. Однією з найважливіших переваг блокчейну є його децентралізований характер. Усі дані зберігаються одночасно на численних вузлах мережі, кожен з яких має копію всього ланцюга блоків. Завдяки цьому система стає стійкою до атак і зловживань, оскільки зміни в одному вузлі не впливають на решту мережі, а будь-яка невідповідність швидко виявляється

іншими вузлами. Це особливо важливо у контексті виборів, де прозорість і незмінність даних мають ключове значення для забезпечення довіри до результатів.

Консенсусні алгоритми є важливою складовою блокчейн систем, оскільки вони відповідають за забезпечення узгодженості між вузлами мережі. Існує кілька типів консенсусних алгоритмів, і кожен з них має свої особливості та переваги. У контексті децентралізованого голосування найбільш часто використовуються такі алгоритми консенсусу, як Proof of Work (PoW), Proof of Stake (PoS) та Proof of Authority (PoA). Кожен з цих алгоритмів вирішує питання узгодження даних різними способами, забезпечуючи баланс між безпекою, ефективністю і децентралізацією.

Proof of Work (PoW) є одним із перших і найбільш відомих алгоритмів консенсусу. Він використовується у блокчейнах таких, як Bitcoin та Ethereum (до переходу на PoS). Основна ідея PoW полягає в тому, що вузли (майнери) змагаються за право додати новий блок до блокчейну, виконуючи складні математичні обчислення. Це забезпечує безпеку мережі, оскільки для внесення змін у блокчейн необхідно контролювати більшість обчислювальної потужності, що є дуже складним і дорогим процесом. Однак, PoW має свої недоліки, серед яких — велика витрата енергії та низька швидкість обробки транзакцій.

Proof of Stake (PoS) є альтернативою PoW, яка була розроблена для покращення енергоефективності та підвищення швидкості роботи блокчейну. У PoS вузли, які додають нові блоки до блокчейну, обираються на основі кількості криптовалюти, що вони володіють, або яку готові поставити у якості застави. Цей підхід значно знижує енергоспоживання, оскільки відпадає необхідність у складних обчисленнях. У контексті виборчих систем PoS забезпечує достатню безпеку та швидкість, що дозволяє оперативно обробляти голоси. Виборчі системи, що використовують PoS, можуть забезпечити швидку обробку голосів без значних енергетичних витрат, що робить цей підхід більш стійким з екологічної точки зору [5].

Proof of Authority (PoA) є ще одним підходом до узгодження даних у блокчейні. У PoA вузли, що мають право додавати нові блоки, є авторизованими та

перевіреними учасниками мережі. Цей алгоритм забезпечує високу швидкість і продуктивність, оскільки немає потреби у виконанні складних обчислень або ставці значної кількості криптовалюти. PoA часто використовується в приватних або корпоративних блокчейнах, де безпека забезпечується не тільки технічними засобами, але й юридичною відповідальністю авторизованих учасників. У виборчих системах PoA може бути корисним для невеликих локальних голосувань, де учасники мережі є відомими і надійними особами, такими як державні організації або спеціально уповноважені особи.

Однією з ключових переваг використання консенсусних алгоритмів у виборчих системах є їх здатність забезпечувати надійність і узгодженість даних без необхідності у центральному органі. Це означає, що жоден вузол або група вузлів не можуть маніпулювати даними без узгодження з іншими вузлами мережі. Завдяки цьому досягається висока ступінь прозорості, оскільки всі транзакції (голоси) мають бути узгоджені більшістю вузлів. Це унеможливлює зміну вже поданих голосів, що забезпечує довіру до результатів виборів.

Крім того, консенсусні алгоритми дозволяють адаптувати блокчейн під конкретні потреби системи голосування. Наприклад, для великих національних виборів може використовуватися PoS, оскільки він забезпечує високу продуктивність і безпеку. Для локальних виборів або корпоративних голосувань може бути використаний PoA, оскільки він має менші вимоги до ресурсів і забезпечує високу швидкість обробки. Вибір алгоритму консенсусу залежить від багатьох факторів, включаючи масштаби голосування, кількість виборців, необхідний рівень безпеки та технічні можливості.

Консенсусні алгоритми є ключовим елементом, що робить блокчейн безпечним і надійним засобом для зберігання даних про голосування. Вони забезпечують узгодженість інформації між усіма вузлами мережі, захищаючи систему від можливих атак або фальсифікацій. Це особливо важливо для виборчих систем, де необхідно забезпечити максимальну прозорість і довіру з боку виборців.

Таким чином, використання блокчейну і консенсусних алгоритмів у системах електронного голосування дозволяє забезпечити високий рівень безпеки,

децентралізації та прозорості. Кожен з алгоритмів консенсусу має свої переваги та недоліки, і вибір конкретного алгоритму залежить від вимог і умов конкретної системи голосування. Завдяки цим технологіям децентралізовані системи можуть забезпечити надійність виборчих процесів, що є особливо важливим у сучасному світі.

## 2.3 Основи криптографії для забезпечення конфіденційності та безпеки голосування

Однією з ключових складових децентралізованих систем електронного голосування є криптографія. Вона забезпечує конфіденційність, автентичність і цілісність даних, що є критичними аспектами для виборчих процесів. Криптографія дозволяє гарантувати, що голоси, які надсилаються виборцями, не можуть бути змінені або підроблені, а також забезпечує захист від зловмисного доступу до інформації.

Основним завданням криптографії у системах голосування є захист конфіденційності голосів. Це означає, що ніхто, окрім самого виборця, не може дізнатися, за кого було подано голос. Для досягнення цієї мети застосовуються різноманітні криптографічні методи, такі як шифрування, цифрові підписи, хешування та протоколи Zero-Knowledge Proof (ZKP).

Шифрування є одним із найважливіших засобів для забезпечення конфіденційності даних. У системах голосування голоси шифруються перед відправкою у блокчейн. Це означає, що навіть якщо хтось отримає доступ до записаних транзакцій, він не зможе дізнатися, за кого проголосував конкретний виборець, оскільки голос зашифрований. Використання асиметричного шифрування є поширеною практикою у виборчих системах. У цьому випадку кожен виборець має пару ключів — публічний і приватний. Публічний ключ використовується для шифрування голосу, тоді як приватний ключ дозволяє самому виборцю підтвердити свій голос і гарантує, що тільки він може здійснити цю дію.

Цифрові підписи забезпечують автентичність і цілісність даних. Коли виборець подає свій голос, він підписує його своїм цифровим підписом, який створюється на основі його приватного ключа. Це дозволяє вузлам блокчейну перевірити, що голос був поданий саме цим виборцем і що дані не були змінені під час передачі. Завдяки цифровим підписам забезпечується захист від підробки голосів, оскільки лише власник відповідного приватного ключа може створити підпис, який відповідає публічному ключу.

Хешування є ще одним важливим інструментом у забезпеченні безпеки децентралізованих систем голосування. Хешування використовується для створення унікального та незворотного представлення даних, таких як інформація про транзакції чи голоси. Хеш-функції перетворюють будь-яке вхідне значення у фіксовану довжину, і навіть невелика зміна вхідних даних призводить до створення абсолютно іншого хешу. У контексті голосування хешування може використовуватися для перевірки цілісності даних, що дозволяє переконатися, що голоси не були змінені після їх запису у блокчейн.

Ще однією важливою криптографічною технологією, що застосовується у виборчих системах, є Zero-Knowledge Proof (ZKP). Цей протокол дозволяє одній стороні довести іншій, що вона володіє певною інформацією, не розкриваючи саму цю інформацію. У виборчих системах ZKP може використовуватися для того, щоб виборець довів свою правомочність брати участь у голосуванні без розкриття своїх особистих даних. Наприклад, виборець може довести, що він зареєстрований і має право голосувати, не розкриваючи свого імені або інших деталей. Це забезпечує високу конфіденційність і захист особистих даних виборців, що є важливим аспектом демократичного процесу.

Протоколи шифрування з гомоморфізмом також набувають популярності у децентралізованих системах голосування. Гомоморфне шифрування дозволяє виконувати математичні операції над зашифрованими даними без необхідності їх розшифрування. Це означає, що результати голосування можуть бути підраховані безпосередньо з зашифрованих голосів, що забезпечує додатковий рівень конфіденційності. Наприклад, жоден з вузлів, що беруть участь у підрахунку, не

може знати, за кого конкретно проголосував виборець, але при цьому всі можуть перевірити правильність підрахунку голосів.

Вибір конкретних криптографічних методів і алгоритмів залежить від вимог до системи, таких як рівень конфіденційності, швидкість обробки даних і масштабність голосування. Криптографія забезпечує надійний захист від багатьох типів атак, таких як атаки на цілісність даних, підміну особистості або несанкціонований доступ до інформації. Виборчі системи повинні гарантувати, що жоден виборець не може проголосувати двічі, що голоси не можуть бути змінені або видалені, і що результат підрахунку є чесним і неупередженим.

Усі ці криптографічні методи об'єднуються у складну систему безпеки, яка забезпечує захист децентралізованої виборчої системи. Криптографія допомагає створити систему, у якій виборці можуть бути впевнені в тому, що їх голос буде враховано, а їх особисті дані залишаться конфіденційними. Крім того, завдяки розподіленій природі блокчейну, система стає стійкою до атак, що спрямовані на центральні сервери, оскільки дані зберігаються на багатьох вузлах одночасно.

Таким чином, криптографія є фундаментом для забезпечення конфіденційності, цілісності та автентичності у системах децентралізованого голосування. Без надійних криптографічних методів неможливо забезпечити безпеку голосів, захист від фальсифікацій та маніпуляцій, а також довіру до результатів виборів. Застосування криптографічних методів у поєднанні з блокчейн-технологією дозволяє створити надійну виборчу систему, яка відповідає сучасним вимогам безпеки та прозорості.

## 2.4 Порівняння централізованих і децентралізованих систем голосування

Голосування є одним із найважливіших аспектів демократичного суспільства, і вибір між централізованими та децентралізованими системами є критичним для забезпечення безпеки, прозорості та довіри до виборчого процесу. Ці дві моделі мають суттєві відмінності в тому, як зберігаються, обробляються та

контролюються дані голосів, що безпосередньо впливає на їх ефективність, безпеку та ступінь незалежності від зовнішніх впливів.

Централізовані системи голосування є традиційним підходом до організації виборчих процесів, де вся інформація зберігається і контролюється одним центральним сервером або групою серверів. Всі дані про виборців, їхні голоси, результати виборів та інші критичні елементи знаходяться в одному місці, що дозволяє зручно керувати та обробляти всю інформацію. Основною перевагою централізованих систем є висока швидкість роботи та легкість адміністрування. Центральна система дає змогу легко вносити зміни, оновлювати дані, додавати нових виборців або кандидатів, а також автоматизувати підрахунок голосів, що забезпечує швидке оприлюднення результатів виборів.

Однак, централізовані системи мають свої слабкі сторони. Оскільки всі дані знаходяться в одному місці, така система стає дуже вразливою до атак хакерів. Якщо зловмисник отримує доступ до центрального сервера, він потенційно може змінити дані про голосування або навіть підробити результати виборів. Така концентрація інформації також робить систему вразливою до атак типу DDoS, які можуть вивести з ладу сервер, що відповідає за підрахунок голосів. Крім того, централізовані системи можуть бути об'єктом маніпуляцій з боку осіб, які мають доступ до керування сервером, що підриває довіру виборців до виборчого процесу.

Децентралізовані системи голосування, натомість, використовують технології блокчейну, що дозволяє розподілити дані між багатьма вузлами, забезпечуючи їх незмінність і прозорість. У децентралізованій системі кожен голос зберігається як транзакція у блокчейні, який розподілений між численними вузлами по всьому світу. Кожен з цих вузлів має копію всієї бази даних, і зміна даних можлива тільки після узгодження більшістю вузлів. Це забезпечує високу стійкість до атак і захищає систему від однієї з головних вразливостей централізованих систем — зосередження інформації в одному місці (див. рисунок 2.2) [6].

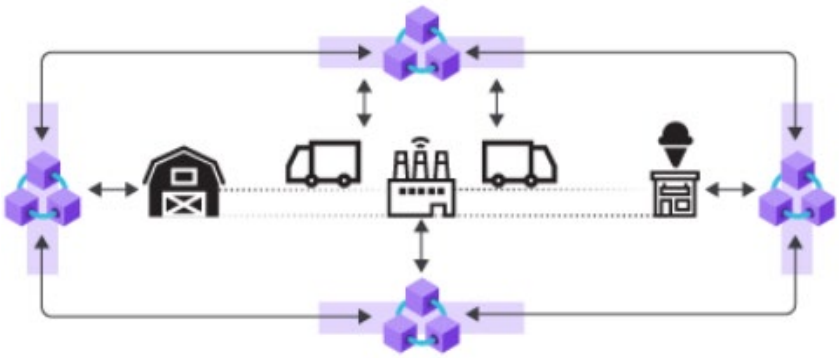

Рисунок 2.2 – Схематичне зображення децентралізованої бази даних з використанням блокчейну

Однією з головних переваг децентралізованих систем є їхня прозорість. Усі транзакції (голоси) записуються у відкритий реєстр, який доступний для перегляду будь-якою зацікавленою стороною. Це забезпечує високий рівень довіри, оскільки виборці можуть бути впевнені, що їх голоси враховані правильно і жодних маніпуляцій не було. Крім того, використання криптографії у децентралізованих системах гарантує конфіденційність голосування, а також захист від повторного голосування. Виборець може проголосувати лише один раз, і цей факт підтверджується усіма вузлами мережі [7].

Однак децентралізовані системи також мають свої виклики. Один з них — швидкість обробки. Оскільки кожен голос повинен бути узгоджений усіма вузлами, це може суттєво вплинути на час обробки голосів, особливо в великих масштабах. Алгоритми консенсусу, такі як Proof of Work (PoW), можуть займати багато часу і ресурсів, що знижує ефективність системи порівняно з централізованими підходами. З іншого боку, більш сучасні алгоритми, такі як Proof of Stake (PoS) або Proof of Authority (PoA), намагаються вирішити ці проблеми, забезпечуючи баланс між швидкістю та безпекою.

Ще одним викликом для децентралізованих систем є складність впровадження та юридичне регулювання. Технологія блокчейн є порівняно новою і складною для багатьох виборчих комісій, що може створити певні труднощі при її інтеграції у існуючі виборчі процеси. Крім того, використання блокчейну може

вимагати нових правових рамок для забезпечення його відповідності вимогам прозорості, конфіденційності та законності, особливо у випадках національних виборів.

Порівнюючи ці дві системи, можна зробити висновок, що централізовані системи є більш простими у впровадженні та забезпечують високу швидкість обробки даних, але їх вразливість до атак та залежність від одного централізованого органу знижують їх надійність та довіру з боку виборців. Децентралізовані системи, натомість, забезпечують високу прозорість і стійкість до атак, але вимагають більше ресурсів для обробки транзакцій і є складнішими у впровадженні.

Вибір між централізованими та децентралізованими системами залежить від конкретних вимог і умов. Для великих національних виборів, де питання прозорості та довіри мають ключове значення, децентралізовані системи можуть бути більш привабливим варіантом [8]. Водночас, для менш масштабних голосувань, таких як корпоративні або місцеві вибори, централізовані системи можуть забезпечити необхідну ефективність і швидкість з мінімальними витратами.

## 2.5 Архітектура децентралізованої системи голосування

### 2.5.1 Виявлення акторів

На рисунку 2.3 представлені актори системи.

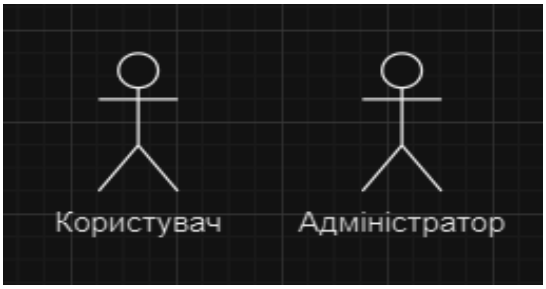


Рисунок 2.3 – Актори системи

Таблиця 2.1 – Виявлення акторів

| Актор | Короткий опис |
|---|---|
| Користувач | Має можливість створювати, переглядати всі голосування. Переглядати деталі голосування, а також голосувати за кандидата. |
| Адміністратор | Має всі можливості, які має користувач, а також створювати голосування. |

2.5.2 Виявлення варіантів використання

Виявлення варіантів використання представлені таблиці 2.

Таблиця 2.2 – Варіанти використання

| | | |
|---|---|---|
| Адміністратор | Створення голосування | Можливість адміністратору створювати голосування. |
| Користувач | Перегляд усіх голосувань | Можливість користувачу переглядати список всіх голосувань із загальною інформацією. |
| Користувач | Перегляд деталей голосування | Можливість користувачу переглядати конкретне голосування з повною інформацією. |
| Користувач | Голосування за кандидата | Можливість користувачу проголосувати за певного кандидата в голосуванні. |

2.5.3 Розробка варіантів використання

На рисунку 2.4 зображено діаграму варіантів використання для користувача.

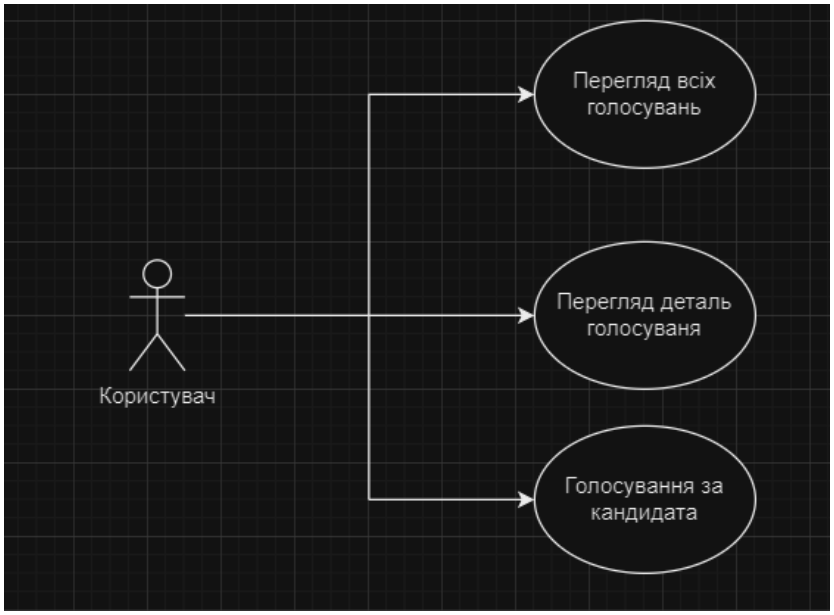


Рисунок 2.4 – Варіанти використання для користувача

На рисунку 2.5 зображено діаграму варіантів використання для адміністратора.

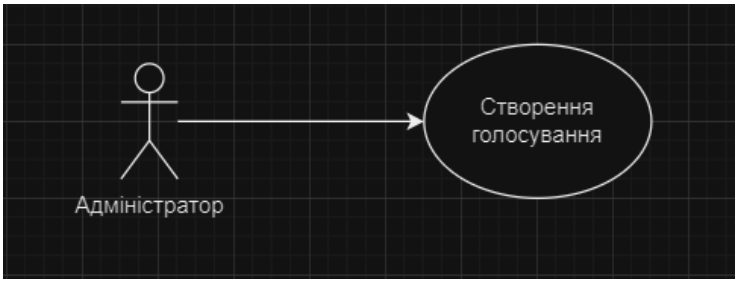


Рисунок 2.5 – Варіанти використання для адміністратора

### 2.5.4 Діаграми послідовності

Діаграма послідовності (sequence diagram) – це вид діаграми взаємодії в UML (Unified Modeling Language), яка демонструє порядок взаємодій між об'єктами або компонентами системи. Вона показує, як об'єкти обмінюються повідомленнями або викликають методи один у одного протягом певного часу.

На діаграмі послідовності об'єкти представлені вертикальними лініями життєвого циклу, а повідомлення, що передаються між ними, відображені стрілками. Ця діаграма показує динамічну поведінку та співпрацю між об'єктами

під час певного сценарію або сценарію використання.

Діаграми послідовності часто використовують для візуалізації та розуміння взаємодій і потоків повідомлень у складних системах. Вони допомагають у проектуванні, документуванні та поясненні поведінки системи, відображаючи динамічні аспекти співпраці об'єктів, необхідні для виконання певних функцій.

Аналіз діаграм послідовності дозволяє розробникам і дизайнерам зрозуміти взаємодії між об'єктами, порядок обміну повідомленнями та послідовність подій, що допомагає виявити можливі проблеми, краще зрозуміти загальну поведінку системи та перевірити правильність дизайну та логіки.

Процес створення голосування відбувається таким чином: спочатку користувач заходить у систему, щоб створити голосування. Заповнює всі потрібні дані і надсилає запит системі. Далі система обробляє запит і виводить повідомлення про успішне створення голосування (див. рисунок 2.6).

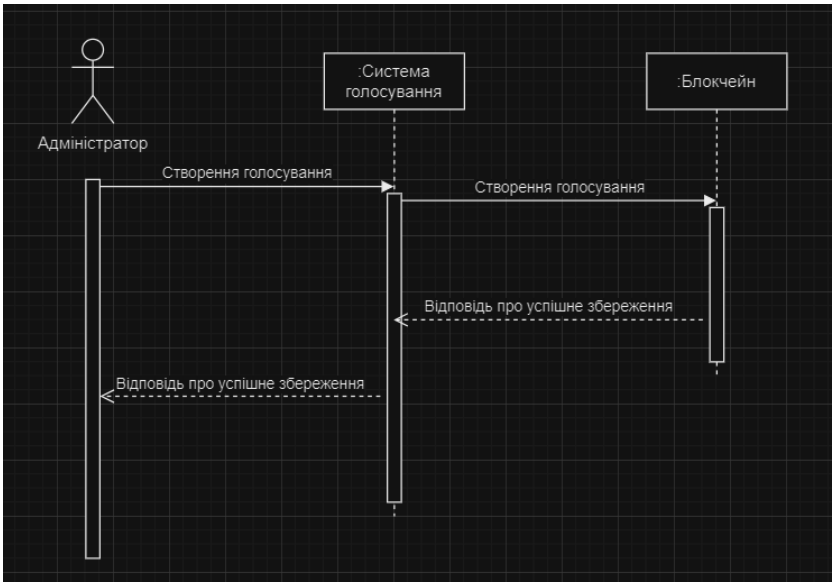


Рисунок 2.6 – Діаграма послідовності створення голосування

Процес отримання інформації про всі голосування відбувається таким чином: спочатку користувач заходить у систему, щоб переглянути список наявних голосувань. Далі система обробляє запит і повертає список голосувань (див. рисунок 2.7).

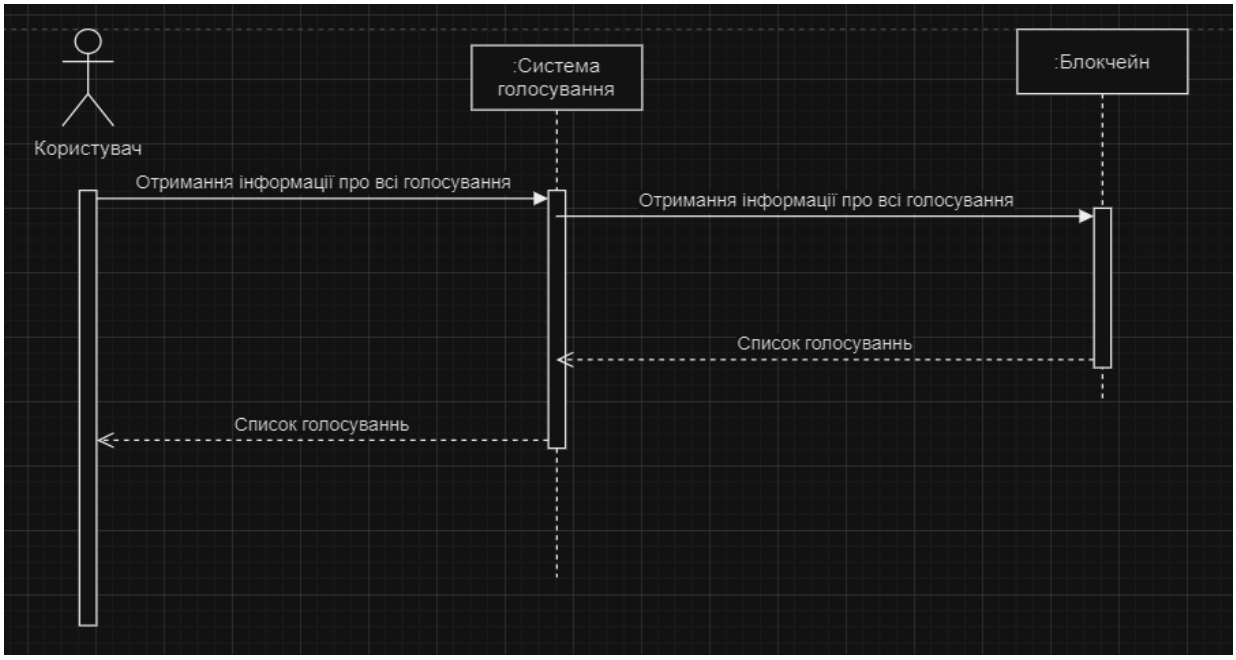


Рисунок 2.7 – Діаграма послідовності отримання всіх голосувань

Процес отримання деталей голосування відбувається таким чином: спочатку користувач заходить у систему, щоб переглянути список наявних голосувань. Вибирає конкретне голосування. Далі система обробляє запит і повертає деталі для конкретного голосування (див. рисунок 2.8).

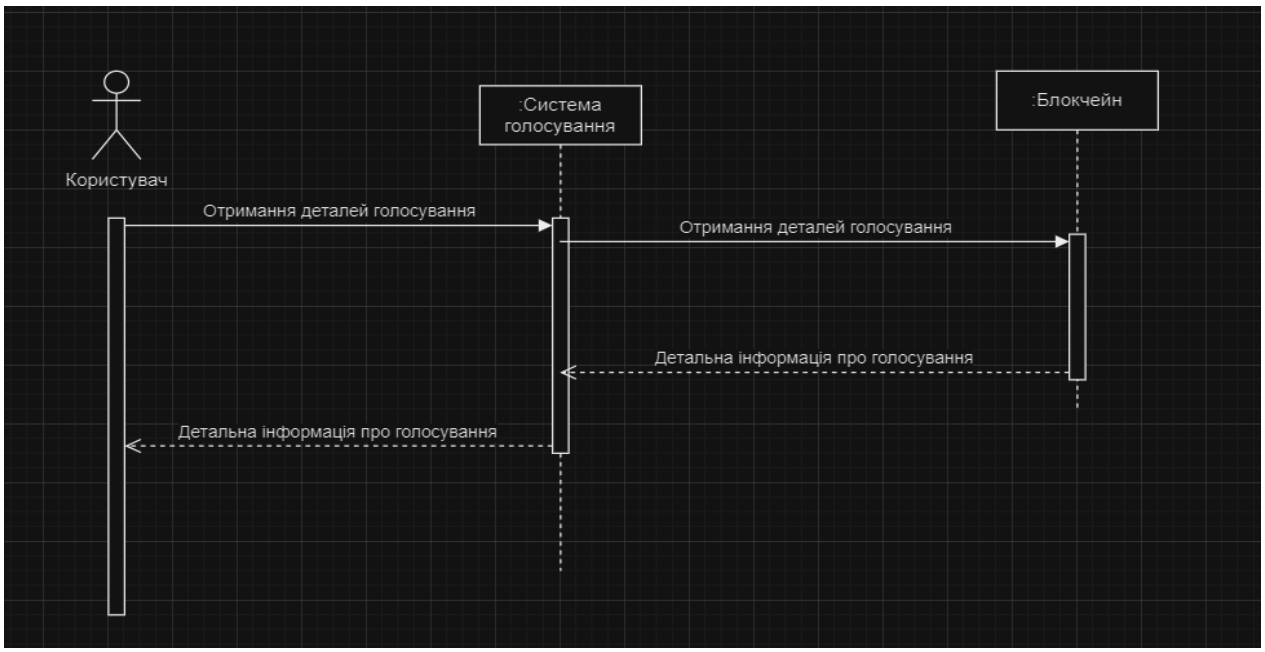


Рисунок 2.8 – Діаграма послідовності отримання деталей голосування

Процес голосування за кандидата відбувається таким чином: спочатку користувач заходить у систему, переглянути список наявних голосувань. Вибирає конкретне голосування. Вибирає конкретного кандидата і натискає кнопку для того щоб проголосувати. Далі система обробляє запит і виводить повідомлення про успішне зарахування голосу (див. рисунок 2.9).

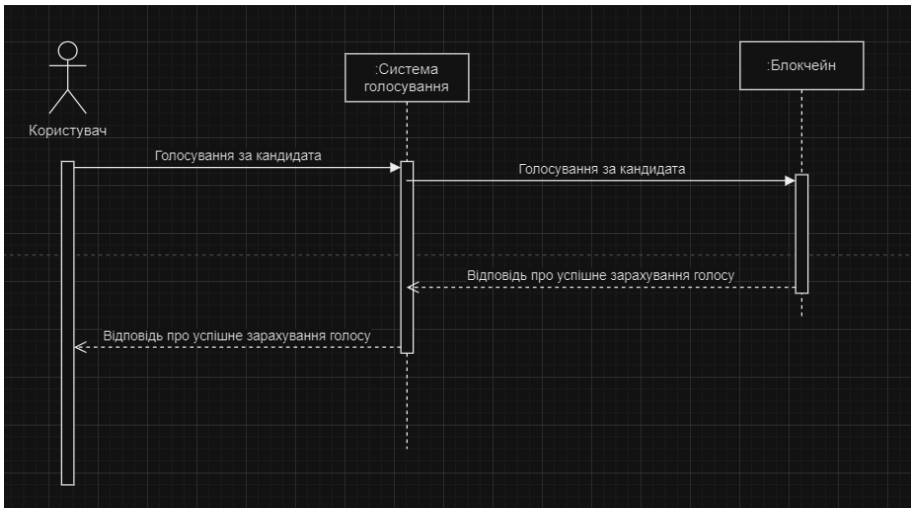


Рисунок 2.9 – Діаграма послідовності голосування за кандидата

# РОЗДІЛ 3 ПРАКТИЧНА ЧАСТИНА

## 3.1 Архітектура децентралізованої системи голосування

Після отримання всієї необхідної інформації про систему можна переходити до проектування її архітектури, а саме — визначення, якими класами вона буде володіти. Важливо встановити всі класи системи, після чого створити UML-діаграму класів, що стане основою архітектури.

Діаграма класів (Class Diagram) є інструментом для графічного представлення структури системи, базованої на об’єктно-орієнтованому програмуванні. Вона надає візуальне представлення класів, типів даних, а також зв'язків між ними. Така діаграма допомагає не лише ілюструвати самі класи та їхні атрибути, але й взаємодію між інтерфейсами, об'єктами та іншими елементами системи, що сприяє кращому розумінню їх поведінки та ролі в системі [9].

У програмуванні клас є ключовою одиницею, яка визначає шаблон для створення об’єктів. Клас слугує основою для створення екземплярів, які представляють конкретні реалізації об'єктів на основі визначених характеристик класу.

Клас об'єднує дані (атрибути або властивості) та поведінку (методи або функції), які пов'язані з певною сутністю або концепцією. Атрибути відображають стан або характеристики об'єкта, тоді як методи визначають операції або дії, які об'єкт може виконувати.

Класи є центральним елементом об'єктно-орієнтованого програмування (ООП) — програмної парадигми, яка надає перевагу організації коду у вигляді повторно використовуваних об'єктів. ООП сприяє модульності, структурованості та ефективності розробки програмного забезпечення, а також заохочує повторне використання коду, інкапсуляцію та абстракцію.

Створення кількох екземплярів одного класу дозволяє створювати об'єкти з однаковими характеристиками та поведінкою, визначеними класом. При цьому

кожен об'єкт може мати власний стан (значення атрибутів) і виконувати методи незалежно від інших об'єктів цього ж класу.

Класи надають можливість моделювати реальні сутності, системи або абстрактні концепції в програмному забезпеченні. Вони дозволяють організувати код таким чином, щоб його було легко повторно використовувати, а також забезпечують структурованість під час проєктування та реалізації складних програмних систем. Це допомагає розробникам ефективніше опрацьовувати логіку та дотримуватися принципів об'єктно-орієнтованого програмування.

Загалом, клас виступає як шаблон або основа, що визначає структуру, атрибути і поведінку об'єктів у контексті ООП [10].

Для серверної частини проєкту була обрана багатошарова архітектура, яка є стандартом для багатьох серверних додатків. Цей архітектурний підхід добре зарекомендував себе завдяки своїй природній відповідності процесу розробки програмного забезпечення, організації комунікації між компонентами і наданню структури бізнес-логіки. Використання багатошарової архітектури дозволяє спростити процес розробки та тестування, оскільки кожен компонент має чітко визначені завдання та функції.

У багатошаровій архітектурі компоненти системи організовані у вигляді вертикальних шарів, кожен з яких відповідає за певний функціонал. Наприклад, існують окремі шари для представлення даних, бізнес-логіки, доступу до даних, а також для роботи з базою даних. Конкретна кількість шарів може відрізнятися залежно від вимог та складності додатку: для менших проєктів кількість шарів може бути зменшена, тоді як для великих і складних додатків можуть бути додані додаткові шари з метою підвищення гнучкості та масштабованості системи [11].

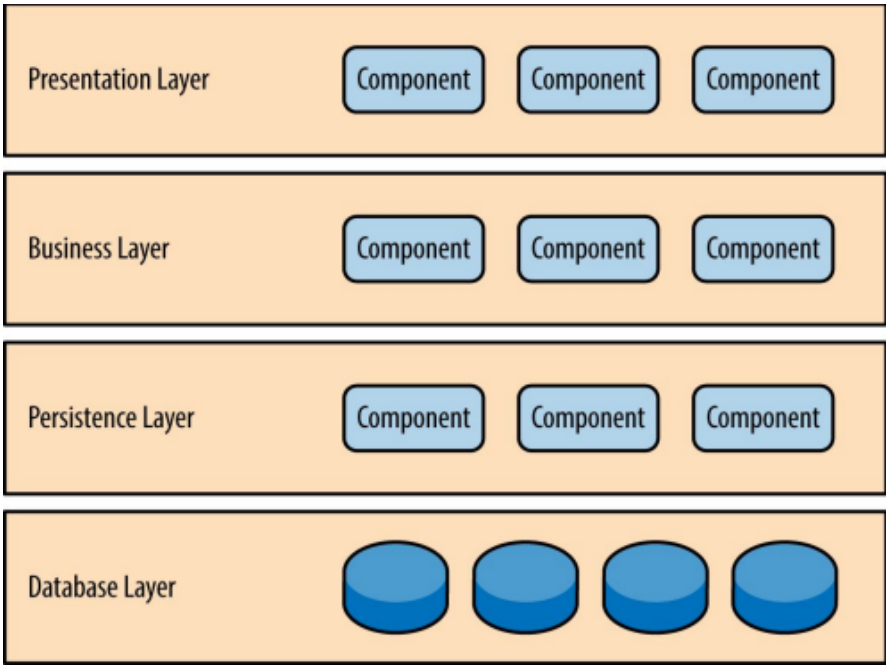


Рисунок 3.1 – Багатошарової архітектура

Структура класів представлена на рисунку 3.2.

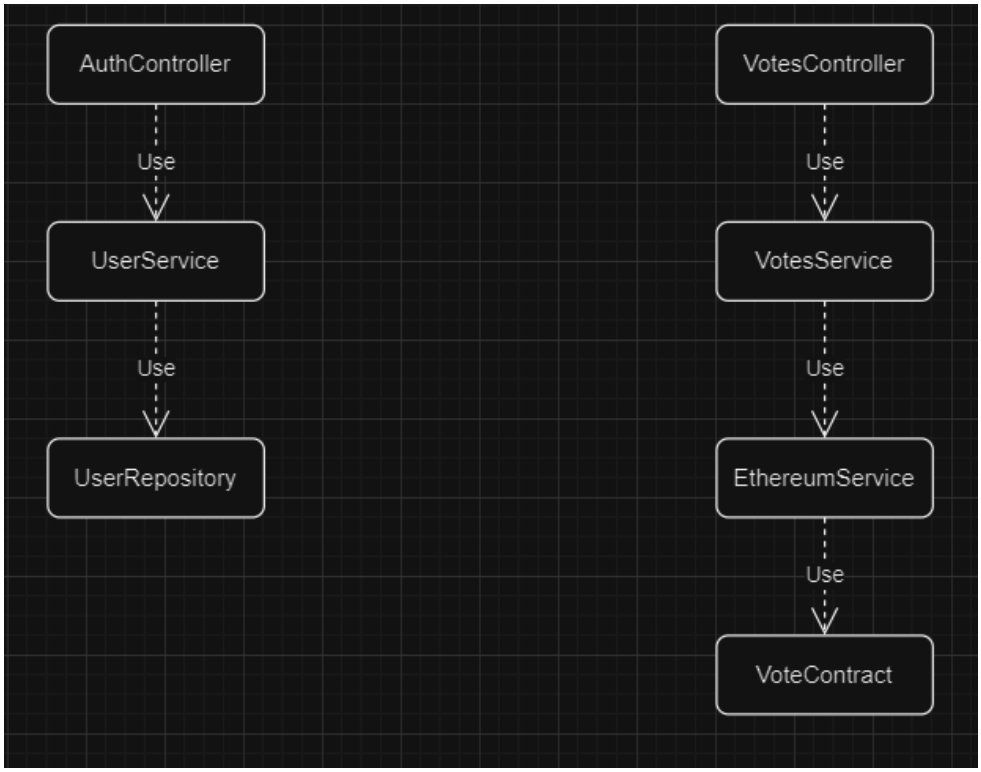


Рисунок 3.2 – Діаграма класів серверного додатку

Архітектура клієнтської частини представлена на рисунку 3.3.

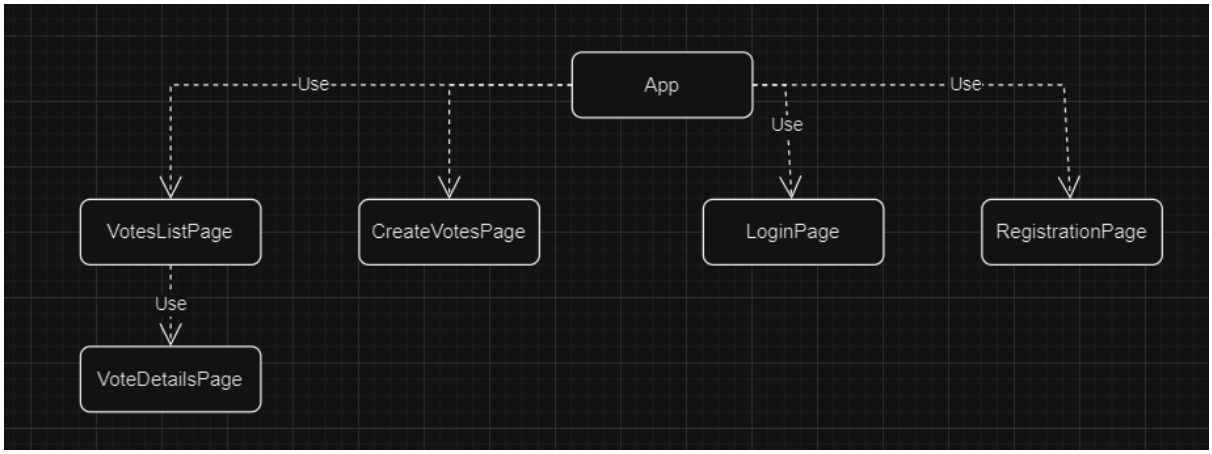


Рисунок 3.3 – Діаграма компонентів клієнтської частини

## 3.2 Вибір технологій та інструментів для реалізації системи

Архітектура клієнт-сервер є моделлю обчислення, де клієнтські пристрої або додатки звертаються до централізованого сервера для запиту та отримання різних послуг, ресурсів або даних. У цій архітектурі клієнт і сервер є окремими компонентами системи, кожен з яких має власні функціональні завдання.

Клієнт, який також називається фронтендом, являє собою додаток або пристрій, що взаємодіє з кінцевим користувачем. Основними завданнями клієнта є забезпечення інтерфейсу користувача, збір і обробка введених даних, а також ініціювання запитів до сервера для подальшої обробки або отримання інформації.

Сервер, також відомий як бекенд, виступає як потужний обчислювальний ресурс, здатний обслуговувати численні клієнтські запити. Сервер відповідає за отримання, обробку та виконання запитів, а також за зберігання даних, управління бізнес-логікою та координацію загальної функціональності системи. Це дозволяє забезпечити цілісність даних та виконання складних обчислювальних процесів, які не могли б бути ефективно реалізовані на клієнтській стороні.

Комунікація між клієнтом і сервером відбувається через мережу, таку як Інтернет. Клієнт надсилає запити до сервера, використовуючи стандартизовані протоколи, зокрема HTTP (протокол передачі гіпертексту) у випадку веб-додатків.

Сервер обробляє ці запити та повертає відповіді клієнту, що забезпечує користувача необхідною інформацією або виконує певні дії.

Клієнт-серверна архітектура відома своїми перевагами, зокрема можливістю масштабування, де декілька клієнтів можуть звертатися до одного сервера або групи серверів для отримання послуг. Це забезпечує ефективне використання обчислювальних ресурсів та можливість обслуговувати велику кількість користувачів одночасно. Крім того, централізоване зберігання даних на сервері сприяє підвищенню безпеки та полегшує управління доступом, що є важливим аспектом для забезпечення цілісності і конфіденційності даних у системі. Ця архітектура часто використовується для створення веб-додатків, систем керування базами даних, поштових серверів, а також інших мережевих систем (див. рисунок 3.4) [12].

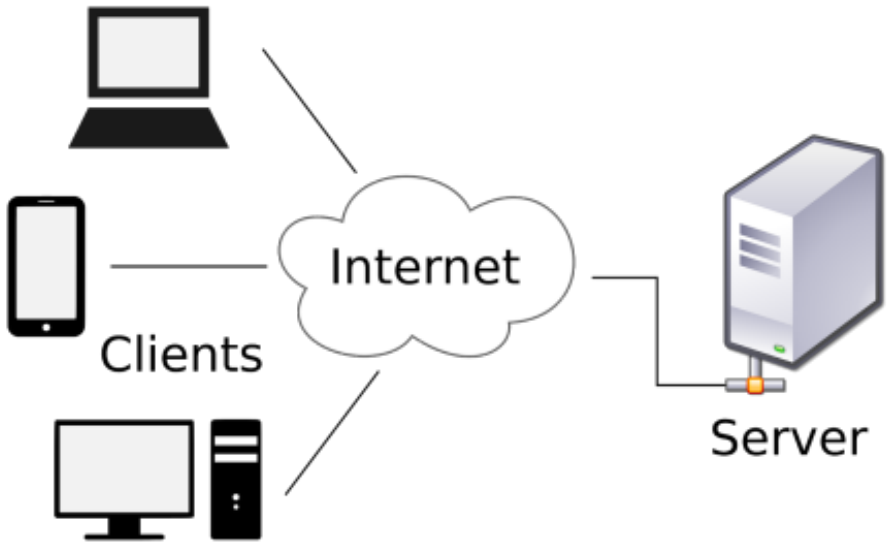


Рисунок 3.4 – Клієнт-серверна архітектура

У розробці серверної частини проекту використовується мова програмування Java, яка відповідає за збереження бази даних та обробку запитів від клієнтів. Java забезпечує надійність, ефективність та масштабованість, що робить її чудовим вибором для розробки серверної частини децентралізованої системи голосування. Ця мова є загального призначення, підтримує паралельність, використовує

об'єктно-орієнтований підхід і має багато спільного з мовами C та C++, хоча й відрізняється спрощенням у деяких аспектах. Основною метою при створенні Java було забезпечення використання в реальних проектах, з акцентом на стабільність і простоту. Відповідно, Java уникає включення нових або неперевірених функцій, що підвищує її надійність у практичних умовах розробки [13].

Для реалізації серверної частини веб-додатку, зокрема для розробки API, обрано Spring Framework. Spring є високопродуктивним інструментом для розробки програмного забезпечення на платформі Java, що надає розробникам комплексне середовище та набір готових компонентів і бібліотек. Це значно спрощує процес створення додатків, особливо тих, що потребують розширюваності та модульності.

Spring базується на принципах інверсії керування (Inversion of Control, IoC) та аспектно-орієнтованого програмування (Aspect-Oriented Programming, AOP). Інверсія керування дозволяє відокремлювати залежності від основного бізнес-логічного коду, що робить програму більш структурованою та легко тестованою. Аспектно-орієнтоване програмування, в свою чергу, дає можливість виділяти перетини логіки, наприклад, безпеку або логування, в окремі модулі, що дозволяє зберігати чистоту основного коду. Завдяки цим підходам, Spring сприяє підвищенню модульності коду, полегшує повторне використання та тестування компонентів.

Крім того, Spring підтримує широкий спектр функцій, зокрема інтеграцію з базами даних, кешування, захист додатків та розробку веб-інтерфейсів. Він також є одним із найпопулярніших фреймворків у світі Java та підтримується великою спільнотою розробників. Наявність великої кількості документації, прикладів коду та активна підтримка від спільноти значно спрощує використання цього фреймворку в розробці проєктів різної складності (див. рисунок 3.5) [14].

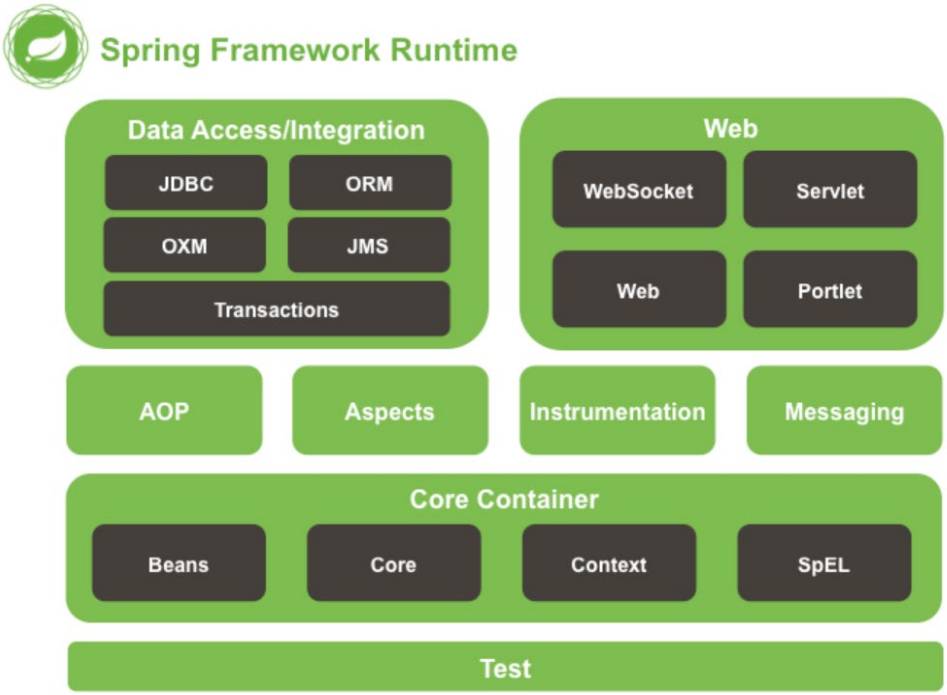


Рисунок 3.5 – Архітектура Spring Framework

Для зберігання даних в рамках розробки системи було обрано PostgreSQL як основну систему управління базами даних. PostgreSQL є потужною об'єктно-реляційною системою з відкритим вихідним кодом, яка вже багато років активно розвивається і зарекомендувала себе як високонадійна, продуктивна і розширювана платформа для роботи з великими обсягами даних. Завдяки розширеним можливостям, таким як підтримка складних запитів, індексації та транзакцій, PostgreSQL ідеально підходить для потреб децентралізованої системи голосування, забезпечуючи стійкість і ефективність у роботі з критичними даними. Крім того, PostgreSQL підтримується активною спільнотою з відкритим вихідним кодом, що надає багато ресурсів, включаючи офіційну документацію, навчальні матеріали та приклади коду для підтримки розробників на кожному етапі реалізації (див. рисунок 3.6) [15].


Рисунок 3.6 – Логотип PostgreSQL

Клієнтська частина системи була розроблена за допомогою бібліотеки React. React є популярним відкритим фреймворком і бібліотекою JavaScript, створеною компанією Facebook. Він призначений для швидкої і зручної розробки інтерактивних інтерфейсів користувача та веб-додатків. Завдяки своїй компонентній архітектурі, React дозволяє розробляти повторно використовувані модулі, що значно прискорює розробку і полегшує підтримку додатків. Крім того, використання Virtual DOM в React дозволяє підвищити ефективність роботи додатку, адже оновлення сторінки відбуваються тільки в необхідних місцях, що забезпечує швидкість і плавність взаємодії користувача з системою (див. рисунок 3.7) [16].

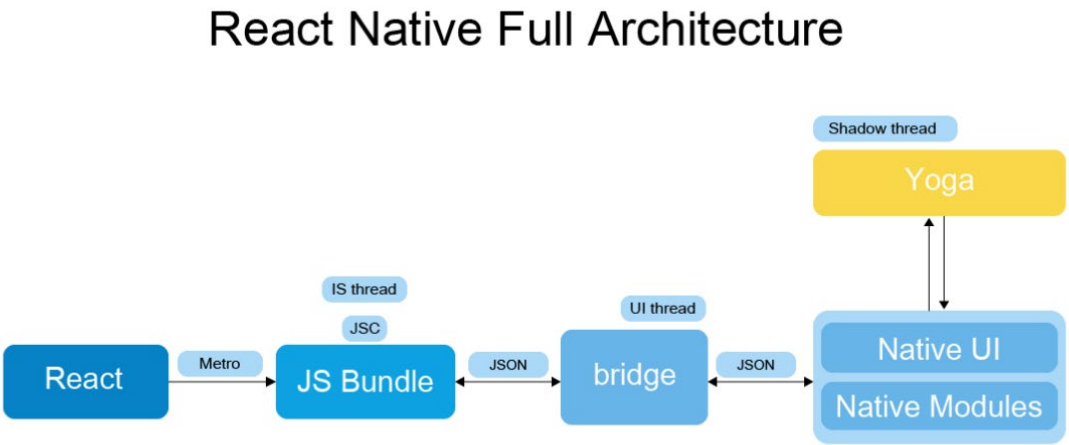


Рисунок 3.7 – Архітектура React

HTML (HyperText Markup Language) і CSS (Cascading Style Sheets) є основними технологіями для створення розмітки та стилів веб-сторінок системи. HTML використовується для структуризації і представлення вмісту на веб-сторінках. За допомогою різних елементів HTML можна створювати текстові абзаци, вставляти зображення, формувати таблиці і створювати посилання. Кожен елемент HTML має свою функцію, яка визначається відповідним тегом [17].

CSS, в свою чергу, відповідає за стилізацію веб-сторінок. CSS дозволяє розробникам визначати, як виглядатиме кожен елемент на сторінці, включаючи кольори, шрифти, відступи, розташування, анімації та багато іншого. CSS дозволяє відокремлювати структуру веб-сторінки (HTML) від її представлення (стилі), що робить код більш зрозумілим і легким для підтримки. Завдяки цьому підходу, можна використовувати один файл CSS для стилізації багатьох сторінок, що спрощує процес розробки і підтримки [18].

Для управління станом клієнтської частини в рамках React додатку, було обрано бібліотеку Redux. Redux є популярним інструментом для управління станом у веб-додатках. Вона забезпечує централізоване сховище стану, що дозволяє ефективно синхронізувати різні компоненти додатку. Використовуючи Redux, стан додатку зберігається в одному місці, а зміни в стані ініціюються за допомогою дій (actions) та редукторів (reducers). Це забезпечує передбачувану поведінку додатку та полегшує тестування і налагодження коду [19].

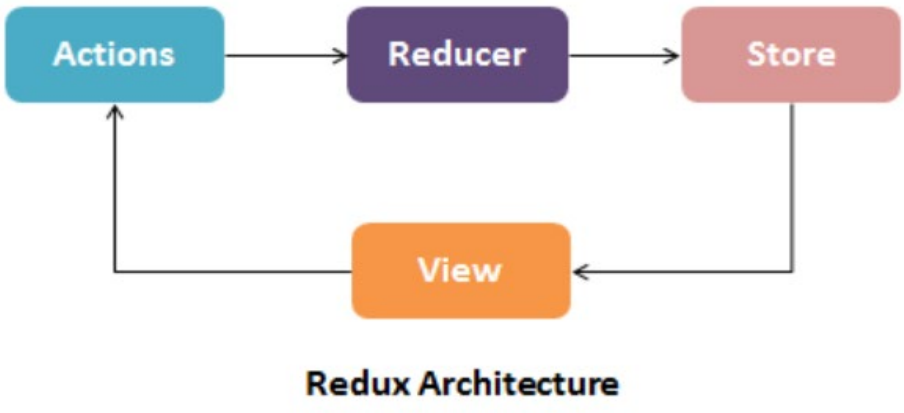


Рисунок 3.8 – Архітектура Redux

IntelliJ IDEA було обрано як основне середовище розробки для цього проекту. Це інтегроване середовище розробки (IDE), яке створене для підвищення продуктивності розробника, особливо при роботі з мовами JVM. IntelliJ IDEA автоматизує рутинні та повторювані завдання, що дозволяє розробникам зосередитись на ключових аспектах розробки. IDE надає функції розумного автодоповнення, статичного аналізу коду та можливості для рефакторингу, що полегшує процес написання чистого, зрозумілого та ефективного коду. Використання IntelliJ IDEA допомагає підвищити продуктивність розробки, мінімізувати помилки, і створити комфортне середовище для розробника (див. рисунок 3.9) [20].

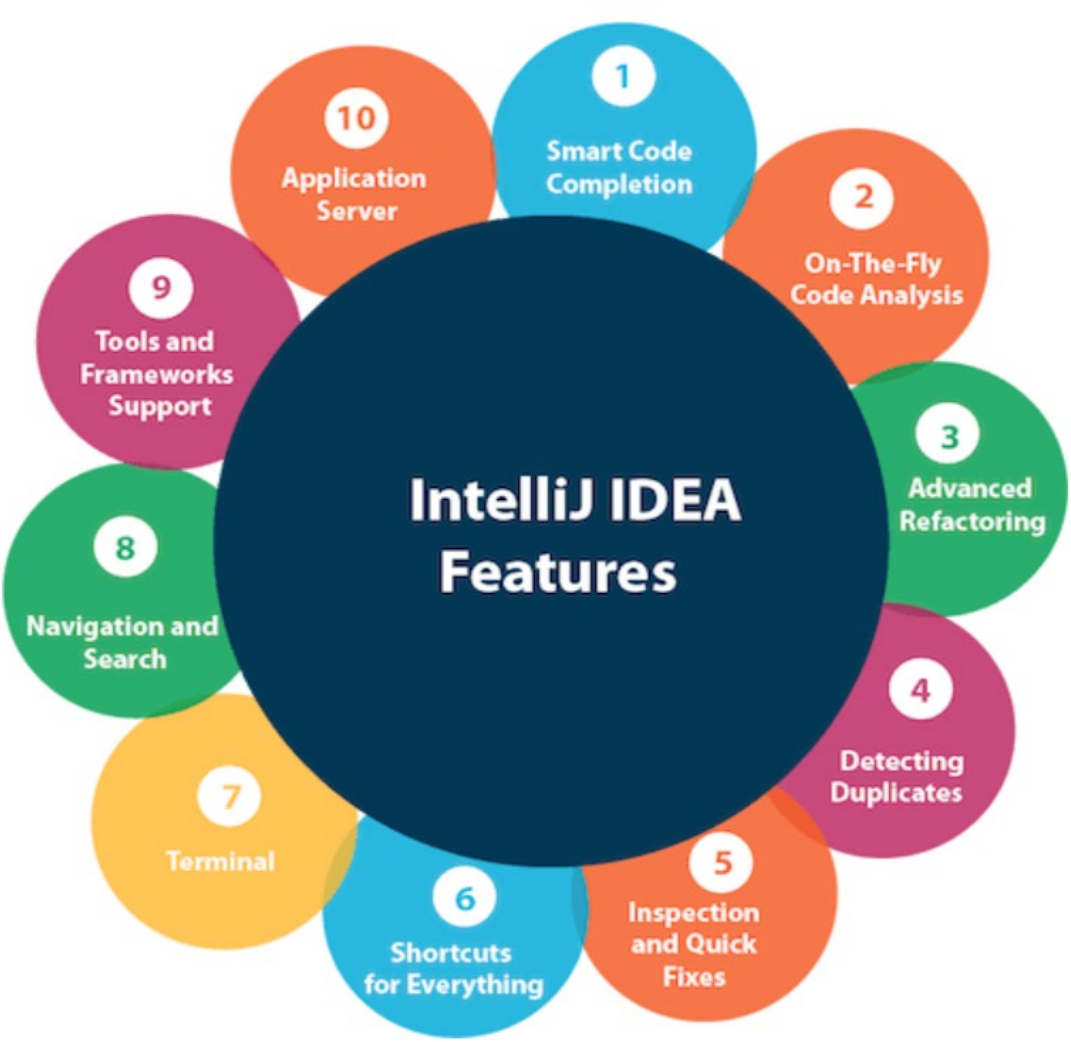


Рисунок 3.9 – Особливості Intellij Idea

## 3.3 Розробка смарт-контрактів для електронного голосування

Розробка смарт-контрактів для електронного голосування є критичним етапом реалізації децентралізованої системи голосування, оскільки саме смарт-контракти забезпечують автономність, прозорість та безпеку виборчого процесу. Смарт-контракт – це комп'ютерний протокол, що дозволяє здійснювати транзакції та автоматизувати виконання угод у блокчейн-мережі. Він виконується без втручання третіх сторін, що забезпечує довіру і виключає можливість маніпуляцій.

Смарт-контракти для системи голосування реалізовані з використанням мови Solidity, яка є основною мовою програмування для розробки смарт-контрактів на платформі Ethereum. Solidity дозволяє створювати та розгортати децентралізовані додатки, включаючи електронні голосування, що гарантує незмінність даних і захист від фальсифікацій. Solidity підтримує об'єктно-орієнтовані принципи програмування, що дозволяє організувати код у вигляді класів і структур, забезпечуючи чітку і логічну організацію всієї системи.

Для розробки смарт-контракту в системі голосування були визначені такі основні функції:

- Створення голосування – визначення назви голосування та списку кандидатів;
- Голосування за кандидата – користувач має можливість проголосувати за обраного кандидата;
- Перевірка права голосу – перевірка, чи користувач вже голосував у певному голосуванні;
- Отримання результатів голосування – ця функція дозволяє переглядати кількість голосів для кожного кандидата у конкретному голосуванні.

Однією з ключових особливостей смарт-контрактів є їх прозорість. Усі дії, які відбуваються в рамках смарт-контракту, записуються в блокчейн, де кожна транзакція зберігається в незмінному вигляді. Це означає, що будь-який користувач може переглянути всі транзакції, пов'язані з голосуванням, що додає рівень довіри

до системи. Завдяки тому, що дані зберігаються в децентралізованій мережі, немає єдиного центру, який міг би маніпулювати результатами.

Для реалізації інтерактивної частини голосування використовуються смарт-контракти з подіями. Події в Solidity дозволяють генерувати лог-дані, які можуть бути відслідковані додатками, що працюють з блокчейном. Наприклад, під час створення нового голосування чи голосування користувачем генерується відповідна подія, яка може бути оброблена клієнтським додатком. Це дозволяє в реальному часі оновлювати дані на стороні клієнта, що є важливим аспектом інтерактивності додатку.

Для зручності тестування та налагодження смарт-контрактів було використано Ganache. Ganache дозволяє створити локальну блокчейн-мережу, що імітує реальну мережу Ethereum, але не вимагає використання реальних токенів. Це дає змогу ефективно тестувати різні сценарії роботи смарт-контрактів, включаючи тестування на стійкість до можливих атак та помилок. Ganache надає інструменти для відстеження всіх транзакцій, а також для швидкої зміни стану мережі, що значно полегшує процес розробки та тестування.

Для інтеграції смарт-контрактів із серверною частиною була використана бібліотека Web3j. Web3j є Java-бібліотекою, яка забезпечує взаємодію з блокчейн-мережею Ethereum. Використовуючи Web3j, додаток може звертатися до смарт-контрактів, надсилати транзакції та отримувати дані з блокчейну. Це спрощує реалізацію бізнес-логіки на стороні серверу та забезпечує надійний зв'язок з децентралізованою мережею. Крім того, Web3j дозволяє створювати та підписувати транзакції, що дозволяє безпечно взаємодіяти з блокчейном, не розкриваючи приватні ключі.

Важливим аспектом розробки смарт-контрактів є забезпечення їх безпеки. Під час реалізації були враховані потенційні вразливості, такі як повторне використання транзакцій (reentrancy attack) та атаки з боку користувачів, що намагаються маніпулювати голосуванням. Для цього було застосовано патерн "переведення коштів до оновлення стану" (Checks-Effects-Interactions), який мінімізує ризик повторних атак. Крім того, контракт був перевірений з

використанням статичного аналізу, що дозволило виявити та усунути можливі проблеми ще на етапі розробки.

Смарт-контракт складається з кількох структур, таких як Candidate та Voting, які визначають логічну структуру виборчого процесу. Кожне голосування має ідентифікатор, назву та список кандидатів, які є частиною структури Voting. Кандидат, у свою чергу, має атрибути, що описують його ідентифікатор, ім'я та кількість голосів. Це забезпечує гнучкість системи, оскільки нові голосування можуть бути легко створені та налаштовані залежно від конкретних вимог.

## 3.4 Тестування системи

Тестування є ключовим етапом у процесі розробки, який спрямований на оцінку якості програмного продукту. Основною метою є забезпечення відповідності продукту встановленим вимогам. Програмна система повинна реалізовувати всі необхідні сценарії використання, не містити дефектів і забезпечувати високу надійність, безпеку, продуктивність, зручність у використанні та можливість подальшого розширення. Тестування дозволяє не лише виявляти недоліки, але й оцінювати ключові характеристики системи, сприяючи створенню якісного та функціонального продукту [21].

Зі стрімким розвитком технологій інтерфейс користувача стає важливою складовою успішних проектів. Це зумовлено тим, що задоволення користувачів та їх взаємодія з інтерфейсом відіграють ключову роль у популярності та успіху сервісів чи веб-сайтів. Зручний і привабливий інтерфейс сприяє тривалішій взаємодії користувачів із сервісом, підвищуючи їхню лояльність. Зворотний зв'язок від користувачів дозволяє вдосконалювати інтерфейс, адаптуючи його до потреб та очікувань цільової аудиторії. Таким чином, якісний інтерфейс користувача стає одним із основних факторів популярності та ефективності сучасних проектів.

Користувацький API (або API інтерфейсу користувача) являє собою набір інструментів і методів, що дозволяють розробникам взаємодіяти з користувацьким інтерфейсом програми чи системи. Цей API забезпечує можливість створення,

зміни та управління візуальними елементами та функціональністю інтерфейсу, що значно спрощує розробку та налаштування взаємодії користувача із системою.

За допомогою користувацького API розробники можуть отримати доступ до таких елементів інтерфейсу, як кнопки, форми, меню, діалогові вікна та інші інтерактивні компоненти. Вони мають змогу програмно змінювати їхній вигляд, розташування та поведінку, обробляти введення користувачів і реагувати на їхні дії. Це створює передумови для гнучкого налаштування інтерфейсу відповідно до потреб користувачів та бізнес-логіки програми.

Користувацький API приховує технічну складність роботи з елементами інтерфейсу, пропонуючи стандартизований і зручний інструментарій для розробників. До його функціональних можливостей належать методи для створення елементів, управління їхніми властивостями та стилями, обробки дій користувачів і динамічного оновлення інтерфейсу на основі логіки програми або змін у даних.

Завдяки користувацькому API можна створювати багатофункціональні, адаптивні та інтуїтивно зрозумілі інтерфейси. Це дозволяє налаштовувати вигляд і функціонал програмного забезпечення, забезпечуючи ефективну взаємодію користувачів із системою.

У процесі розробки інтерфейсу для цього проекту основна увага була приділена простоті використання та зручності взаємодії. Головні елементи інтерфейсу були ретельно продумані, щоб гарантувати легкість освоєння і безперешкодне користування сервісом.

Коли користувач заходить у систему, йому доступні дві можливості: авторизація і реєстрація. Допоки користувач не авторизується у системі, інші функції йому будуть не доступні.

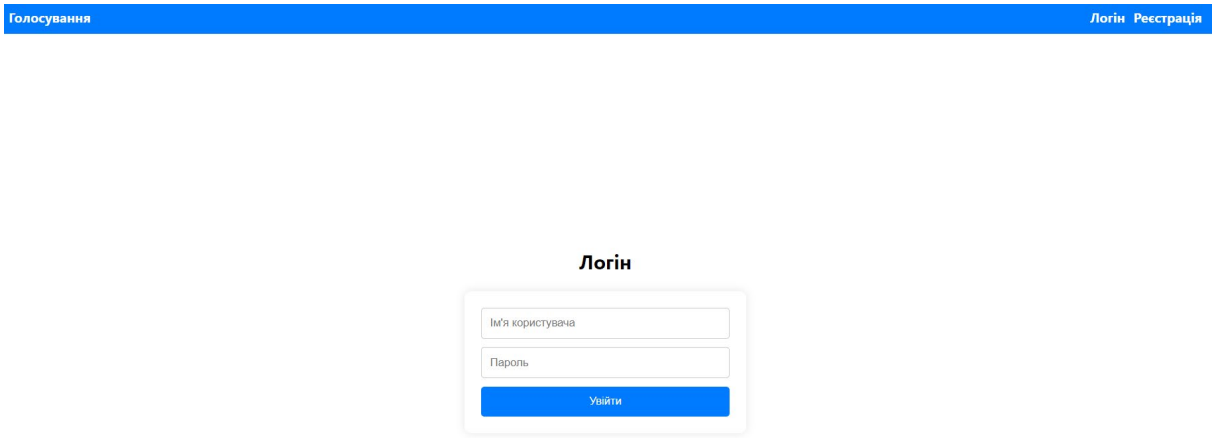


Рисунок 3.10 – Сторінка авторизації

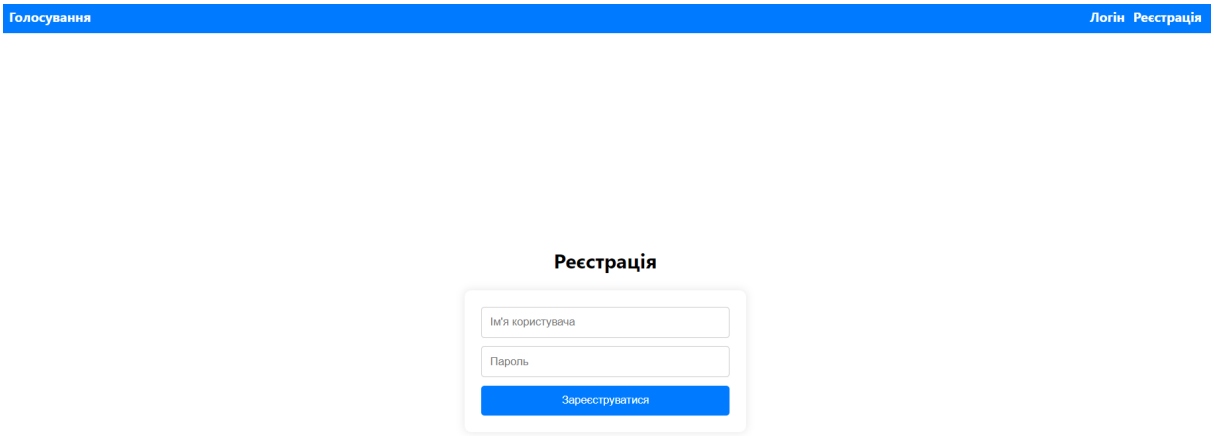


Рисунок 3.11 – Сторінка реєстрації

При успішній авторизації, користувача перенаправляє на сторінку зі списком всіх голосувань. Також зверху, адміністратор може побачити шапку з посиланнями, такі як: «Голосування», «Створити голосування». Проте звичайному користувачу посилання для створення голосування не є доступним і не відображається.

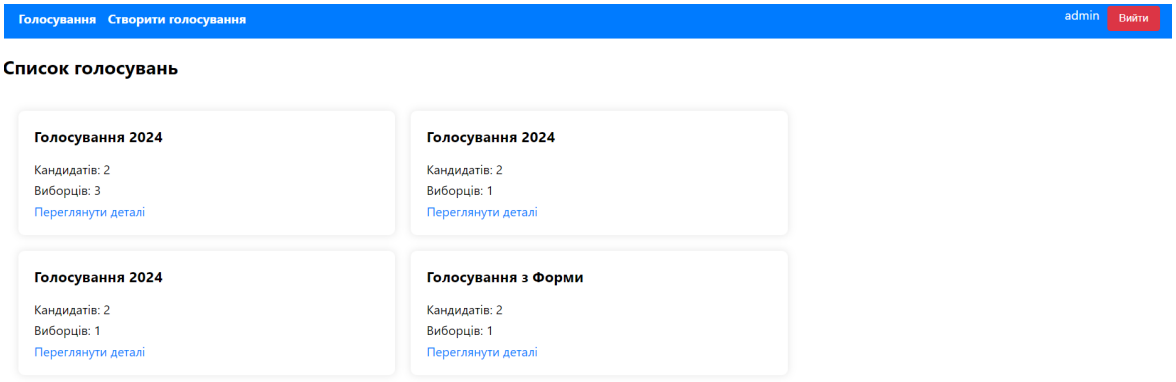


Рисунок 3.12 – Сторінка списку всіх голосувань

На сторінці для створення голосувань, адмін може створити голосування. Для цього йому потрібно ввести назву голосування і додати кандидатів. При успішному створенні голосування, виводиться повідомлення. А також викликається метод смарт контракту.

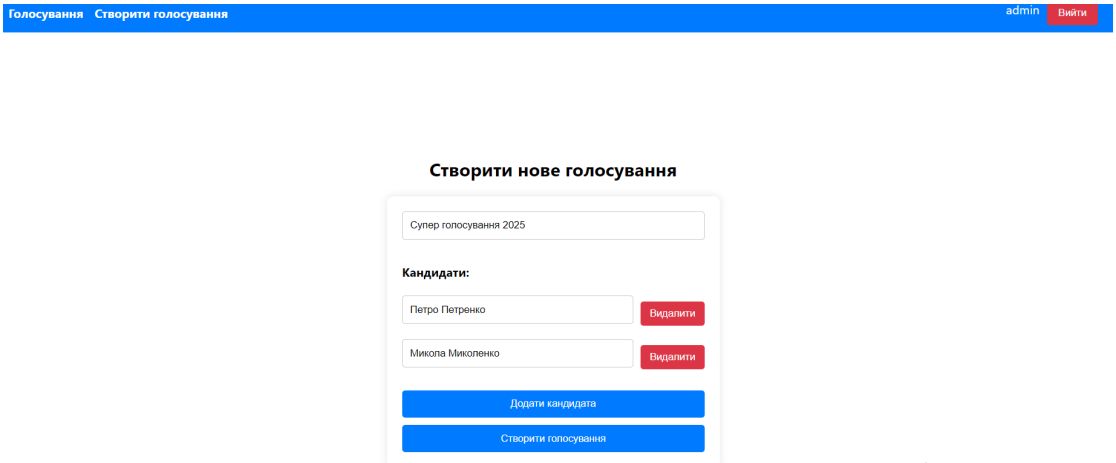


Рисунок 3.13 – Сторінка створення голосування

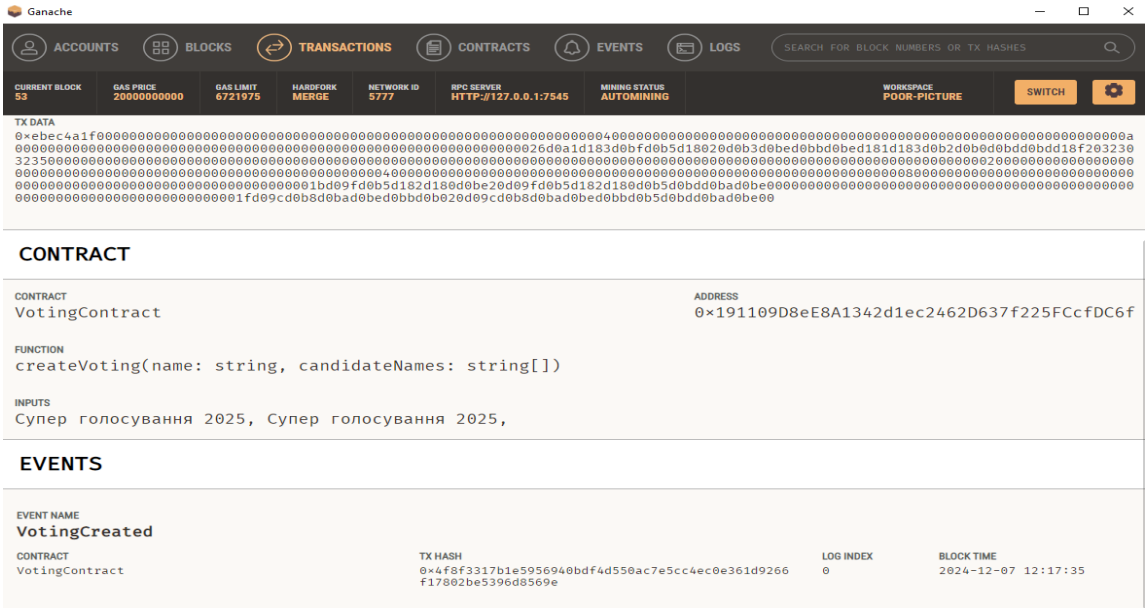


Рисунок 3.14 – Виклик смарт-контракту для створення голосування

На сторінці деталей голосування, користувач має можливість переглянути список всіх кандидатів, а також кількість голосів, які вже має кандидат. Також є можливість проголосувати за кандидата, якщо це не було зроблено. В межах одного голосування можна проголосувати тільки за одного кандидата.

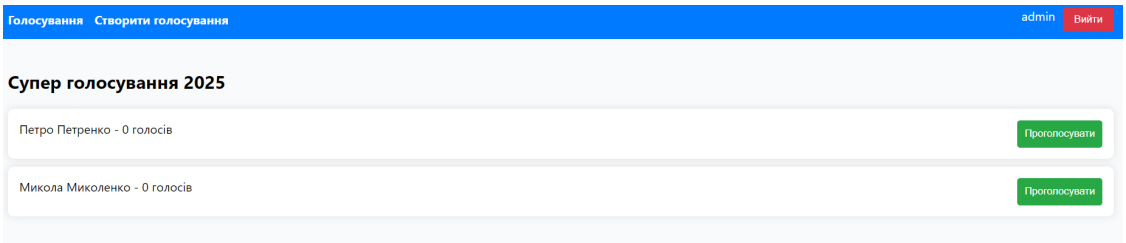


Рисунок 3.15 – Сторінка деталей голосування

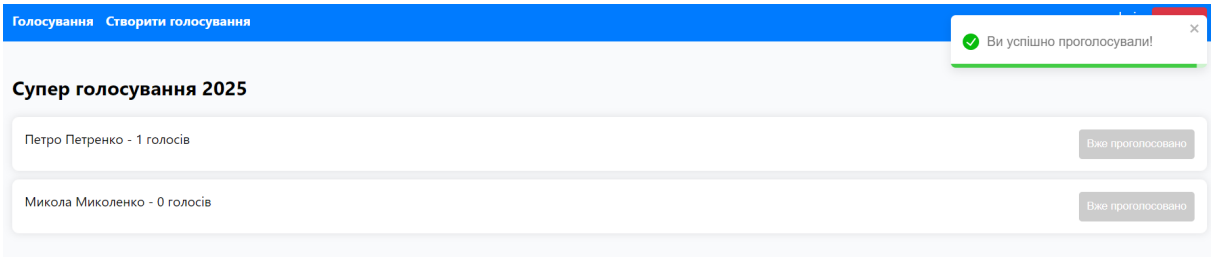


Рисунок 3.16 – Процес голосування за кандидата

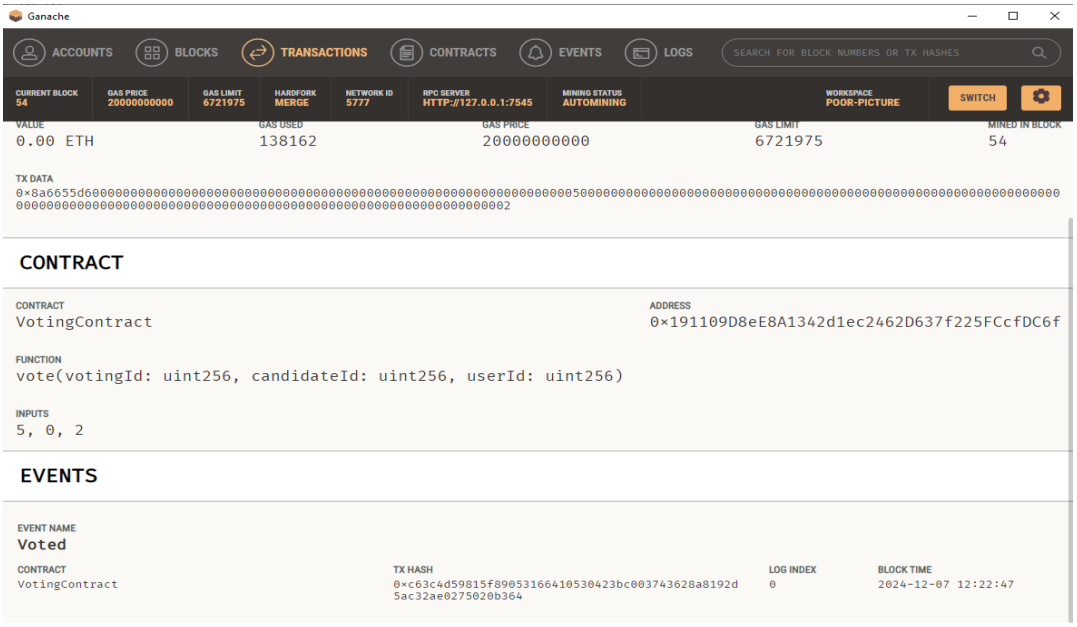


Рисунок 3.17 – Виклик смарт-контракту для процесу голосування

Тестування програмного забезпечення є ключовим етапом у розробці будь-якої інформаційної системи, зокрема децентралізованої системи електронного голосування. У цьому розділі було проведено перевірку основних функціональних аспектів системи, таких як створення голосувань, подача голосів, обробка результатів та взаємодія між клієнтською, серверною частинами й блокчейном.

Функціональне тестування підтвердило, що всі основні сценарії роботи системи виконуються коректно відповідно до специфікацій. Користувачі змогли створювати нові голосування, подавати голоси та переглядати результати без збоїв.

Хоча блокчейн забезпечує високий рівень безпеки, існують потенційні загрози, які слід враховувати під час розробки та експлуатації системи:

- Атаки на смарт-контракти;
- Фішинг та соціальна інженерія;
- DDoS-атаки.

Проведене тестування показало, що система відповідає заявленим функціональним вимогам, однак забезпечення її стійкості до потенційних загроз вимагає постійного моніторингу, аудиту безпеки та впровадження відповідних механізмів захисту.

# 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

## 4.1. Охорона праці

Розробка децентралізованої системи електронного голосування на основі блокчейну здійснювалась з використанням вектора метрики якості з використанням портативних пристроїв та персональних комп'ютерів. Відповідно, в ході виконання експлуатації необхідно працювати з комп'ютерною технікою. Тому надзвичайно важливим фактором безпеки праці є дотримання правил техніки безпеки, охорони праці та протипожежної безпеки при користуванні комп'ютерною технікою.

Питання охорони праці регулюються певними законодавчими та нормативно-правовими актами, які, зокрема, визначають обов'язки роботодавця із забезпечення робітникам комфортних та безпечних умов для здійснення роботи. Ці обов'язки, а також права робітників на таких умовах праці передбачені частиною 2 ст.2 і частиною 1 ст.21 КЗпП, а також ст.13 Закону України «Про охорону праці» [1], у яких визначаються основні положення з реалізації конституційного права робітників. Існує цілий ряд вимог, які визначають специфіку заходів з охорони праці при роботі з персональним комп'ютером. Законодавчі та нормативно-правові акти, які за участі відповідних органів державної влади регулюють відносини між роботодавцем та робітником з питань безпеки, гігієни праці та виробничого середовища, а також встановлюють єдиний порядок організації охорони праці в Україні. На їх основі розроблені чисельні документи: правила, інструкції, норми, державні санітарні правила та інші нормативно-правові документи, якими мають керуватись роботодавці та які регламентують певні питання щодо конструкції електронно-обчислювальної техніки, та особливостей їх розміщення .

На сьогодні основними документами, які регламентують питання охорони праці при використанні працівниками персональних комп'ютерів, можна вважати наступні підзаконні акти:

− НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями»[2];

− ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин»[3].

У відповідності з цими документами, при розробці будь-якого програмного забезпечення, в тому числі при розробці системи електронного голосування на основі блокчейну, необхідно вжити всіх необхідних заходів з охорони праці.

Окрім цього, в залежності від кількості працівників в ІТ компанії та від кількості працівників задіяних в проекті по розробці системи електронного голосування на основі блокчейну служба охорони праці виглядає наступним чином:

− В ІТ компанії або проекті з кількістю працюючих 50 і більше осіб, роботодавець створює службу охорони праці відповідно до типового положення, що затверджується центральним органом виконавчої влади, що забезпечує формування державної політики у сфері охорони праці.

− В ІТ компанії або проекті з кількістю працюючих менше 50 осіб, функції служби охорони праці можуть виконувати в порядку сумісництва особи, які мають відповідну підготовку.

− В ІТ компанії або проекті з кількістю працюючих менше 20 осіб, для виконання функцій служби охорони праці можуть залучатися сторонні спеціалісти на договірних засадах, які мають відповідну підготовку.

Підпорядковується служба охорони праці згідно із законодавством безпосередньо роботодавцеві [1]. Проте, роботодавець може доручити функціональне управління (кураторство) діяльністю служби іншій посадовій особі, скажімо, головному інженерові, заступникові директора з охорони праці тощо. Покладення таких обов’язків потрібно закріпити наказом або відобразити в посадовій інструкції уповноваженої особи. Робота служби охорони праці підприємства має здійснюватися відповідно до плану роботи та графіків обстежень, затверджених роботодавцем. Функції служби охорони праці:

- Підготовка проектів наказів (розпоряджень) з питань охорони праці і внесення їх на розгляд роботодавцю. Проведення спільно з представниками інших структурних підрозділів і за участю представників професійної спілки підприємства або, за її відсутності, уповноважених найманими працівниками осіб із питань охорони праці перевірок дотримання працівниками вимог нормативно-правових актів з охорони праці.
- Проведення з працівниками вступного інструктажу з питань охорони праці та супутніх інструктажів.
- Ведення обліку та проведення аналізу причин виробничого травматизму, професійних захворювань, аварій на виробництві, заподіяної ними шкоди.
- Забезпечення належного оформлення і зберігання документації з питань охорони праці.
- Складання звітності з охорони праці за встановленими формами.
- Складання за участю керівників підрозділів підприємства переліків професій, посад і видів робіт, на які повинні бути розроблені інструкції з охорони праці, що діють в межах підприємства, надання методичної допомоги під час їх розроблення.
- Інформування працівників про основні вимоги законів, інших нормативно-правових актів та актів з охорони праці, що діють в межах підприємства.
- Розгляд питань про підтвердження наявності небезпечної виробничої ситуації, що стала причиною відмови працівника від виконання дорученої роботи відповідно до законодавства (у разі необхідності).
- Організація - забезпечення підрозділів нормативно-правовими актами з охорони праці та актами з охорони праці, що діють в межах підприємства, посібниками, навчальними матеріалами з цих питань [1].

Дотримання даних вимог є необхідним при роботі з системою електронного голосування на основі блокчейну.

## 4.2 Безпека в надзвичайних ситуаціях

Проєктування та розробка веб-додатку передбачає безперервну роботу з персональним комп’ютером, тому вкрай важливо правильно організувати робоче місце оператора ПК. Ергономічно оптимізована робоча зона відіграє важливу роль у сприянні здоров’ю, комфорту та продуктивності людей, які проводять тривалий час за комп’ютером.

Ергономіка досліджує, розробляє та дає рекомендації щодо конструювання, виготовлення та експлуатації технічних засобів, які забезпечують людині в процесі праці необхідні зручності, зберігають її сили, працездатність та здоров'я [25]. Це особливо актуально для користувачів ПК, оскільки неправильна організація робочого місця може призвести до фізичного дискомфорту, втоми, зниження продуктивності, а також розвитку професійних захворювань, таких як тунельний синдром, біль у спині, напруження очей.

Робочі місця працівників, які обладнані комп’ютерами пристроями, повинні відповідати вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце під час виконання робіт стоячи» [26] та ДСанПІН 3.3.2.007-98 «Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [27].

Основні положення ДСТУ 8604:2015:

Конструкція робочого місця та взаємне розташування всіх його елементів (сидіння, органів керування, засобів відображення інформації тощо) повинні відповідати антропометричним, фізіологічним і психологічним вимогам, а також характеру виконуваної роботи.

Конструкцією робочого місця повинно бути забезпечено виконання трудових операцій у межах зони досяжності моторного поля.

Висоти сидіння та підставки для ніг (у разі нерегульованої висоти робочої поверхні). У цьому разі висоту робочої поверхні встановлюють за номограмою для працюючих зростом 1800 мм. Оптимальна робоча поза для менших за зростом працюючих досягається збільшенням висоти робочого сидіння й підставки для ніг

на величину, що дорівнює різниці між висотою робочої поверхні для працюючого зростом 1800 мм і висотою робочої поверхні, оптимальної для зросту даного працюючого.

Підставка для ніг повинна бути регульованою по висоті. Її ширина повинна бути не менше ніж 300 мм, довжина - не менше ніж 400 мм. Поверхня підставки повинна бути рифленою. По передньому краю доцільно передбачати бортик висотою 10 мм.

Основні положення ДСанПІН 3.3.2.007-98:

Площа на одне робоче місце має становити не менше ніж 6,0 м2, а об'єм не менше ніж 20,0 м3.

Приміщення для роботи повинні мати природне та штучне освітлення.

Природне освітлення має здійснюватись через світлові прорізи, орієнтовані переважно на північ чи північний схід і зебезпечувати коефіцієнт природною освітленості (КПО) не нижче ніж 1,5%.

Для внутрішнього оздоблення приміщень слід використовувати дифузно-відбивні матеріали з коефіцієнтами відбиття для стелі 0,7-0,8, для стін 0,5-0,6.

Яскравість світильників загального освітлення в зоні кутів випромінювання від 50 до 90 градусів з вертикаллю в повздовжній та поперечній площинах має становити не більше ніж 200 кд/м2, захисний кут світильників - не менше ніж 40 градусів

При розміщенні робочих столів, відстані між бічними поверхнями: 1,2 м, відстань від тильної поверхні одного до екрана іншого - 2,5 м.

Показник засліплення у разі використання джерел загального штучного освітлення у виробничих приміщеннях має не перевищувати 20.

Необхідно обмежувати нерівномірність розподілу яскравості в полі зору працюючих з екранами. При цьому співвідношення яскравості робочих поверхонь має бути не більшим ніж 3:1, а співвідношення яскравості робочих поверхонь та поверхонь стін, обладнання тощо - 5:1.

Крім того, у контексті сучасних офісів та віддаленої роботи варто враховувати динамічну ергономіку, яка включає зміну положення тіла протягом

робочого дня, використання регульованих столів для роботи сидячи та стоячи, а також перерви для розминки [28].

Захист зору та рекомендації щодо екрану:

− Екран має розташовуватись на відстані 50-70 см від очей користувача, з верхньою частиною екрана на рівні очей або трохи нижче;

− Яскравість екрану повинна відповідати рівню освітлення приміщення, щоб уникнути надмірного напруження очей [28].

Оснащення робочого місця:

− Крім стандартного столу і стільця, важливо передбачити підставки для ноутбука, зовнішню клавіатуру та мишу для зменшення навантаження на кисті рук і плечі;

− Використання додаткових моніторів повинно відповідати нормам розташування і яскравості, щоб уникнути перевтоми [28].

# ВИСНОВКИ

У процесі виконання магістерської роботи була розроблена децентралізована система електронного голосування на основі блокчейн-технологій. Система забезпечує прозорість, безпеку та надійність виборчого процесу, відповідаючи сучасним вимогам до цифровізації демократичних процедур.

Розробка базувалася на багатошаровій архітектурі, яка дозволила чітко розмежувати логіку клієнтської та серверної частин. Серверна частина реалізована на мові Java з використанням Spring Boot, що забезпечує гнучкість, масштабованість та ефективну взаємодію з базою даних PostgreSQL. Клієнтська частина побудована з використанням React.js, що дало змогу створити інтуїтивно зрозумілий інтерфейс для користувачів різних ролей.

Для забезпечення безпеки голосування застосовані сучасні криптографічні методи, а смарт-контракти на мові Solidity автоматизують ключові процеси, виключаючи можливість маніпуляцій з даними. Тестування системи виконано у локальному середовищі за допомогою Ganache, що дозволило перевірити функціональність усіх компонентів.

Результати роботи демонструють перспективність застосування децентралізованих технологій у виборчих процесах. Запропонована система має потенціал до впровадження в реальних умовах, забезпечуючи довіру громадян, анонімність та прозорість голосування.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Krimmer, R., Volkamer, M., & Barrat, J. (2010). Electronic Voting: Theory and Practice. URL: https://link.springer.com/book/10.1007/978-3-642-12980-3

2. Gentry, C. (2009). Fully Homomorphic Encryption Using Ideal Lattices. URL: https://crypto.stanford.edu/craig/

3. Buterin, V. (2014). A Next-Generation Smart Contract and Decentralized Application Platform. URL: https://ethereum.org/en/whitepaper/

4. Tymoshchuk, D., Yasniy, O., Mytnyk, M., Zagorodna, N., Tymoshchuk, V., (2024). Detection and classification of DDoS flooding attacks by machine learning methods. CEUR Workshop Proceedings, 3842, pp. 184 - 195.

5. King, S., & Nadal, S. (2012). PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake. URL: https://peercoin.net/assets/paper/peercoin-paper.pdf

6. Lypa, B., Horyn, I., Zagorodna, N., Tymoshchuk, D., Lechachenko T., (2024). Comparison of feature extraction tools for network traffic data. CEUR Workshop Proceedings, 3896, pp. 1-11.

7. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. URL: https://bitcoin.org/bitcoin.pdf

8. Zyskind, G., Nathan, O., & Pentland, A. (2015). Decentralizing Privacy: Using Blockchain to Protect Personal Data. URL: https://ieeexplore.ieee.org/document/7163223

9. Tymoshchuk, D., & Yatskiv, V. (2024). Slowloris ddos detection and prevention in real-time. Collection of scientific papers «ΛΌГОΣ», (August 16, 2024; Oxford, UK), 171-176.

10. Visual Paradigm: Class DiagramGuide. URL: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/

11. ТИМОЩУК, Д., & ЯЦКІВ, В. (2024). USING HYPERVISORS TO CREATE A CYBER POLYGON. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES, (3), 52-56.

12. Клієнт-серверна архітектура. URL:

https://www.wikiwand.com/uk/Клієнт-серверна_архітектура

13. Tymoshchuk, V., Pakhoda, V., Dolinskyi, A., Karnaukhov, A., & Tymoshchuk, D. (2024). MODELLING CYBER THREATS AND EVALUATING THE PERFORMANCE OF INTRUSION DETECTION SYSTEMS. Grail of Science, (46), 636–641. https://doi.org/10.36074/grail-of-science.29.11.2024.081

14. Офіційна документація Spring. URL: https://spring.io

15. Офіційна документація PostgreSQL. URL:https://www.postgresql.org

16. Офіційна документація React. URL:https://uk.legacy.reactjs.org

17. Офіційна документація HTML. URL: https://www.w3.org/TR/html53/

18. Офіційна документація CSS. URL: https://www.w3.org/Style/CSS/current-work

19. Офіційна документація Redux. URL: https://redux.js.org

20. Офіційна документація Intellij Idea. URL: https://www.jetbrains.com/help/idea/discover-intellij-idea.html

21. ТИМОЩУК, Д., ЯЦКІВ, В., ТИМОЩУК, В., & ЯЦКІВ, Н. (2024). INTERACTIVE CYBERSECURITY TRAINING SYSTEM BASED ON SIMULATION ENVIRONMENTS. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES, (4), 215-220.

22. Закон України «Про охорону праці». URL: https://zakon.rada.gov.ua/laws/show/2694-12

23. Закон України «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». URL: https://zakon.rada.gov.ua/laws/show/z0508-18

24. Закон України «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» . URL: https://zakon.rada.gov.ua/rada/show/v0007282-98

25. В. Ц. Жидецький, В. С. Джигирей, О. В. Мельников. Основи охорони праці. — Вид. 2-е, стереотипне. — Львів: Афіша, 2004., 16 с.

26. ДСТУ 8604:2015. URL: https://zakon.rada.gov.ua/rada/show/v0204774-15

27. ДСанПІН 3.3.2.007-98. URL: https://zakon.rada.gov.ua/rada/show/v0007282-98

28. Occupational Safety and Health Administration (OSHA). Ergonomics in the Workplace. URL: https://www.osha.gov/ergonomics

Додаток А Публікація

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**
**ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**
**імені ІВАНА ПУЛЮЯ**
**НАУКОВЕ ТОВАРИСТВО ім. ШЕВЧЕНКА**

**М А Т Е Р І А Л И**

**XII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ**

# «ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»

**18-19 грудня 2024 року**

**ТЕРНОПІЛЬ**

**2024**

УДК 001
М34

## ПРОГРАМНИЙ КОМІТЕТ

**Голова:** Приймак Микола – професор кафедри комп'ютерних систем та мереж, д.т.н., професо**р.**

**Співголови:** Марущак Павло – проректор з наукової роботи, докт. техн. наук, професор.

Баран Ігор – канд. техн. наук, доцент, декан факультету ФІС.

**Науковий секретар:** Надія Крива– старший викладач.

**Члени:** Василь Кривень - завідувач кафедри математичних методів в інженерії д.ф.-м.н., професор; Галина Осухівська – завідувач кафедри комп'ютерних систем та мереж, к.т.н., доцент; Микола Карпінський - професор кафедри кібербезпеки, д.т.н., професор; Жанна Баб'як - завідувач кафедри української та іноземних мов, к.пед. н., доцент; Ярослав Литвиненко – професор кафедри комп'ютерних наук, д.т.н., професор; Михайло Петрик - завідувач кафедри програмної іженерії, д.ф.-м.н., професор; Наталія Загородна – завідувач кафедри кібербезпеки, к.т.н., доцент.

## ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

**Голова:** Скоренький Юрій Любомирович – канд. техн. наук, доцент кафедри фізики.

**Члени:** доцент кафедри комп'ютерних наук, к.т.н. В. Никитюк; доцент кафедри програмної інженерії, к.т.н. Д. Михалик; доцент кафедри кібербезпеки, к.т.н. М. Стадник; доцент кафедри комп'ютерних систем та мереж, к.т.н. Є. Тиш; ст. викладач Л. Джиджора.

М34 Матеріали XII науково-технічної конфції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 18–19 грудня 2024 р.). – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2024.

**Адреса оргкомітету:** ТНТУ ім. І. Пулюя, м. Тернопіль, вул. Руська, 56, 46001,
тел. (0352) 52-41-33, факс (0352) 25-49-83.
E-mail: conffis2024@gmail.com

Редагування, оформлення та верстка: Надія Крива

## СЕКЦІЇ КОНФЕРЕНЦІЇ, ЯКІ ПРЕДСТВЛЕНІ В ЗБІРНИКУ

- Математичне моделювання
- Інформаційні системи та технології, кібербезпека
- Комп'ютерні системи та мережі
- Програмна інженерія та моделювання складних розподілених систем
- Новітні фізико-технічні та освітні технології

В збірнику надруковано тези доповідей XII науково-технічної конференції «Інформаційні моделі, системи та технології» (Тернопіль, 18–19 грудня 2024 р.) за такими науковими напрямками: математичне моделювання; інформаційні системи та технології, кібербезпека; комп'ютерні системи та мережі; програмна інженерія та моделювання складних розподілених систем; новітні фізико-технічні та освітні технології.

Розрахований на науковців, викладачів та студентів вузів.

**За зміст тез та дотримання норм академічної доброчесності відповідальність несе автор**.

**УДК 004.77**
**Микола Ратишин, студент гр.СБм-62**
Тернопільський національний технічний університет імені Івана Пулюя

## РОЗРОБКА ДЕЦЕНТРАЛІЗОВАНОЇ СИСТЕМИ ЕЛЕКТРОННОГО ГОЛОСУВАННЯ НА ОСНОВІ БЛОКЧЕЙНУ

**Mykola Ratyshyn, student of gr. СБм-62**
## DEVELOPMENT OF A DECENTRALIZED ELECTRONIC VOTING SYSTEM BASED ON BLOCKCHAIN

Сучасні виборчі процеси стикаються з низкою проблем, серед яких маніпуляції результатами голосування, недостатня прозорість, а також обмежена довіра з боку виборців. Тому розробка децентралізованої системи електронного голосування є надзвичайно актуальною, особливо у світі, що прагне до цифрової трансформації та забезпечення безпеки виборчих процесів [1].

Розробка децентралізованої системи електронного голосування може бути реалізована на основі технології блокчейн. Блокчейн забезпечує надійність, прозорість і незмінність даних, завдяки збереженню інформації у вигляді ланцюга блоків, які пов'язані криптографічними хешами [2]. Це дозволяє запобігти будь-яким фальсифікаціям і забезпечити повну прозорість виборчого процесу.

Використання смарт-контрактів на платформі Ethereum дозволяє автоматизувати правила та процес голосування. Смарт-контракти відповідають за обробку голосів і збереження результатів у децентралізованій мережі, що виключає можливість зовнішнього впливу чи маніпуляцій з даними [3].

Реалізація системи здійснена з використанням таких технологій, як Spring Boot для серверної частини, бібліотеки Web3j для інтеграції з блокчейн-мережею та React.js для клієнтського інтерфейсу. Система протестована у локальному середовищі за допомогою Ganache — інструменту для імітації мережі Ethereum, що дозволило перевірити взаємодію між клієнтом, сервером та блокчейном.

Алгоритм голосування в системі побудований з використанням смарт-контрактів, які виконують функції створення голосування, додавання кандидатів, перевірки користувача, та запису голосу. Смарт-контракт отримує запит від авторизованого користувача на голосування, перевіряє, чи користувач вже проголосував, та у разі позитивного результату записує голос у блокчейн. Цей процес побудований на основі функцій Solidity, що забезпечують всі етапи від створення голосування до його завершення. Таким чином, алгоритми системи гарантують, що кожен користувач може проголосувати лише один раз, а результати зберігаються в незмінному вигляді у блокчейн-мережі.

Таким чином, запропонована система має високий потенціал для застосування в умовах виборчих процесів, забезпечуючи анонімність, надійність і прозорість голосування, що є важливими вимогами для сучасних демократичних систем. Використання блокчейн-технології разом із криптографією робить цю систему безпечною та важко зламною, що є ключовою перевагою для демократичних процесів.

**Література:**

1. Anil Kumar M. "Blockchain for Securing E-Voting." International Journal of Computer Applications, 2020.
2. Wood G. Ethereum: A Secure Decentralized Generalized Transaction Ledger. URL: https://ethereum.org/en/whitepaper/
3. Swan M. Blockchain: Blueprint for a New Economy. O'Reilly Media, 2015.