AKADEMIA GÓRNICZO-HUTNICZA

im. Stanisława Staszica w Krakowie

Projekt zaliczeniowy z przedmiotu 'Studio Projektowe 2'

AGH

Zdecentralizowana aplikacja do tworzenia ankiet oraz głosowania Web3 Voting App

Autorzy:

Kaja Dzielnicka, Szymon Frączek, Piotr Gąsiorek

Rok akademicki 2023/2024

Spis treści

1 Wstęp

Web3 Voting App to nowoczesne rozwiązanie do zarządzania ankietami i głosowaniami, wykorzystujące technologię blockchain. W dobie rosnącej potrzeby transparentności i bezpieczeństwa danych, aplikacja oferuje użytkownikom możliwość tworzenia ankiet oraz oddawania głosów z gwarancją niezmienności i transparentności wyników, dzięki zastosowaniu smart kontraktów.

Jej głównym celem jest umożliwienie bezpiecznych głosowań bez konieczności zaufania do centralnych podmiotów, zapewniając anonimowość uczestników i odporność na manipulacje. Idealna dla społeczności, organizacji i firm, aplikacja wspiera różne scenariusze głosowań, od wyborów wewnętrznych po decyzje biznesowe.

Raport ten przedstawia szczegółowy opis technologii wykorzystanych w projekcie, architekturę systemu, funkcjonalności aplikacji oraz propozycje usprawnień. Omówimy również proces rejestracji i uwierzytelniania użytkowników oraz rozwiązania problemów związanych z przechowywaniem danych na blockchainie. Na koniec zaprezentujemy propozycje ułatwienia korzystania z aplikacji dla użytkowników bez specjalistycznej wiedzy o blockchainie.

2 Technologie oraz architektura systemu

Nasza aplikacja wykorzystuje najnowsze technologie, aby zapewnić wydajność, bezpieczeństwo i niezawodność. Kluczowe technologie zastosowane w projekcie to:

2.1 Backend

Backend aplikacji został zbudowany przy użyciu frameworka Axum w języku Rust. Rust jest znany ze swojej wysokiej wydajności i bezpieczeństwa, co czyni go idealnym wyborem dla aplikacji, która musi obsługiwać wiele równoczesnych żądań i zapewniać bezpieczeństwo danych. Axum to nowoczesny framework webowy dla Rust, który umożliwia łatwe tworzenie skalowalnych i wydajnych serwisów webowych.

- **Rust Axum**: Zapewnia wysoką wydajność i bezpieczeństwo.
- **MongoDB**: Baza danych używana do przechowywania danych użytkowników i ankiet. MongoDB jest skalowalna i elastyczna, co pozwala na efektywne zarządzanie dużymi ilościami danych.

Backend odpowiada za:

- Przechowywanie danych użytkowników i ankiet w MongoDB.
- Uwierzytelnianie użytkowników.
- Obsługę transakcji związanych z ankietami i głosowaniem.
- Komunikację z blockchainem poprzez smart kontrakty.

2.2 Frontend

Frontend aplikacji został zbudowany przy użyciu Next.js, frameworka do React.js, który umożliwia server-side rendering oraz łatwą nawigację i optymalizację SEO.

- **Next.js**: Umożliwia szybkie i efektywne renderowanie stron oraz lepszą optymalizację SEO. Jest to framework oparty na React.js, co zapewnia nowoczesne i responsywne interfejsy użytkownika.

Frontend zapewnia:

- Intuicyjny interfejs użytkownika do tworzenia i zarządzania ankietami.
- Możliwość głosowania w ankietach.
- Real-time notyfikacje o statusie transakcji za pomocą WebSocket.

2.3 Smart Contracty

Smart kontrakty, które zarządzają logiką głosowania i tworzenia ankiet, zostały napisane w języku Solidity i wdrożone lokalnie za pomocą Hardhat, środowiska programistycznego do Ethereum.

- **Hardhat**: Narzędzie do tworzenia, testowania i wdrażania smart kontraktów na Ethereum. Hardhat umożliwia łatwe testowanie kontraktów na lokalnym blockchainie przed wdrożeniem na rzeczywisty blockchain.
- **Solidity**: Język programowania używany do tworzenia smart kontraktów na Ethereum. Jest to dominujący język dla kontraktów inteligentnych ze względu na swoją kompatybilność z EVM (Ethereum Virtual Machine).

Smart kontrakty zarządzają:

- Tworzeniem ankiet.
- Oddawaniem głosów.
- Weryfikacją wyników głosowań.

2.4 Blockchain

Aplikacja wykorzystuje lokalny blockchain do testowania i rozwoju smart kontraktów, co pozwala na szybkie iteracje i debugowanie.

- **Local Hardhat Blockchain**: Lokalne środowisko blockchain umożliwiające testowanie i rozwój smart kontraktów bez potrzeby wdrażania na publiczny testnet (nie zostały tam wdrożone ze względu na ograniczenia czasowe i budżetowe). Pozwala to na szybsze iteracje i mniejsze koszty związane z rozwojem.

2.5 Komunikacja z Blockchainem

Aplikacja komunikuje się z blockchainem za pomocą WebSocket, co umożliwia real-time notyfikacje i aktualizacje statusu transakcji.

- **WebSocket**: Protokół komunikacyjny umożliwiający dwukierunkową komunikację między serwerem a klientem. Używany do natychmiastowego informowania użytkowników o statusie ich transakcji bez konieczności odświeżania strony.

3 Bezpieczeństwo

Bezpieczeństwo jest kluczowym elementem projektu Web3 Voting App. Wszystkie komponenty aplikacji są zaprojektowane z myślą o ochronie danych użytkowników i zapewnieniu integralności procesu głosowania.

- **Uwierzytelnianie**: System uwierzytelniania sprawdza, czy dany publiczny adres jest już zarejestrowany w bazie danych. Nowi użytkownicy muszą zarejestrować się za pomocą adresu e-mail z domeny agh.edu.pl.
- **Transparentność**: Dzięki blockchainowi wszystkie transakcje są transparentne i możliwe do zweryfikowania przez każdego użytkownika

4 Funkcjonalności

Aplikacja Web3 Voting App oferuje szeroki zakres funkcjonalności, które zapewniają użytkownikom pełną kontrolę nad procesem tworzenia i zarządzania ankietami oraz głosowaniem. Poniżej przedstawiono główne funkcje aplikacji:

4.1 Rejestracja i logowanie

Połączenie z portfelem:

- Użytkownik rozpoczyna proces logowania/rejestracji od połączenia się ze swoim portfelem kryptowalutowym, np. MetaMask.
- Aplikacja automatycznie pobiera publiczny adres użytkownika z portfela.

Weryfikacja adresu:

- Backend sprawdza, czy dany publiczny adres istnieje już w bazie danych.
- Jeśli adres istnieje, użytkownik jest automatycznie logowany.
- Jeśli adres nie istnieje, użytkownik jest przekierowywany do formularza rejestracyjnego.

Rejestracja:

- Użytkownik podaje swój nick oraz adres e-mail z domeny agh.edu.pl.
- Po wprowadzeniu wymaganych danych, konto użytkownika jest tworzone, a adres portfela jest zapisywany w bazie danych.
- Po zakończeniu rejestracji użytkownik jest logowany do systemu.

4.2 Tworzenie ankiet

Tworzenie nowej ankiety:

- Zalogowani użytkownicy mogą tworzyć ankiety poprzez interfejs użytkownika.
- Proces tworzenia ankiety obejmuje definiowanie pytaia oraz możliwych odpowiedzi.
- Użytkownik może również określić czas trwania ankiety oraz inne parametry, takie jak liczba możliwych odpowiedzi na pytanie.

Zapis ankiety na blockchainie:

- Po zdefiniowaniu ankiety, jej dane są zapisywane na blockchainie za pomocą smart kontraktu.
- Dzięki temu ankieta staje się niezmienna i publicznie dostępna, co zapewnia jej transparentność.

4.3 Głosowanie

Oddawanie głosów:

- Użytkownicy mogą przeglądać dostępne ankiety i oddawać głosy.
- Każdy głos jest zapisywany na blockchainie, co gwarantuje jego niezmienność i publiczną weryfikację.
- Użytkownik musi być połączony ze swoim portfelem kryptowalutowym, aby oddać głos, co pozwala na jednoznaczne przypisanie głosu do konkretnego publicznego adresu.

Anonimowość:

- System zapewnia anonimowość głosujących poprzez przechowywanie jedynie publicznych adresów portfeli bez dodatkowych danych osobowych.

4.4 Zarządzanie ankietami

Przegląd ankiet:

- Twórcy ankiet mają możliwość przeglądania swoich ankiet oraz śledzenia ich postępów.
- Po opublikowaniu ankiety jej dane stają się niezmienne ze względu na zapis na blockchainie.

Zakończenie głosowania i sprawdzanie wyników:

- Twórcy ankiet mogą w dowolnym momencie zakończyć głosowanie.
- Użytkownicy mogą przeglądać wyniki ankiet poprzez interfejs użytkownika, który pobiera dane z blockchaina.

5 Propozycje ulepszeń

Aby jeszcze bardziej zwiększyć funkcjonalność i użyteczność aplikacji Web3 Voting App, przygotowaliśmy kilka propozycji ulepszeń. Te zmiany mogą znacznie poprawić doświadczenia użytkowników i efektywność systemu, wprowadzając nowe technologie i usprawnienia w zakresie interfejsu użytkownika oraz zarządzania danymi.

5.1 Ulepszenie nawigacji i funkcjonowania bez oczekiwania na zatwierdzenie transakcji

Aby poprawić wygodę poruszania się po stronie i umożliwić użytkownikom funkcjonowanie bez oczekiwania na zatwierdzenie transakcji, wprowadziliśmy już pierwsze kroki w tym kierunku. Proces zatwierdzania transakcji na blockchainie może być czasochłonny i frustrujący dla użytkowników, dlatego kluczowe było stworzenie mechanizmu, który pozwala na kontynuowanie pracy z aplikacją bez blokad

Obecne rozwiązanie:

W celu zwiększenia wygody użytkowników, każda transakcja wykonywana przez użytkownika jest natychmiast przesyłana na serwer backendowy. Backend obsługuje przetwarzanie transakcji w sposób asynchroniczny, co oznacza, że nie blokuje dalszej interakcji użytkownika z aplikacją. Aby zapewnić płynność działania i informować użytkowników o postępach ich transakcji, backend wykorzystuje połączenie WebSocket do komunikacji z frontendem. Dzięki temu użytkownicy są natychmiast powiadamiani o statusie swoich transakcji w czasie rzeczywistym. Mechanizm ten pozwala użytkownikom kontynuować korzystanie z aplikacji, na przykład tworzyć kolejne ankiety czy oddawać głosy, podczas gdy ich wcześniejsze transakcje są przetwarzane w tle. To podejście zapewnia non-blocking user experience, co znacząco poprawia komfort korzystania z aplikacji.

Dalsze kroki do wdrożenia:

1. **Optymalizacja Backend-WebSocket**: Zapewnienie niskich opóźnień i wysokiej niezawodności połączeń WebSocket, aby użytkownicy byli natychmiastowo informowani o statusie transakcji. Zwiększenie wydajności serwera backendowego oraz optymalizacja protokołu komunikacyjnego pozwoli na szybszą i bardziej stabilną wymianę danych pomiędzy serwerem a klientem.

2. **UI/UX Enhancements**: Implementacja wizualnych wskaźników statusu transakcji w interfejsie użytkownika, takich jak paski postępu, ikony ładowania czy powiadomienia. Informacje o statusie transakcji (np. "w toku", "zatwierdzone", "odrzucone") będą wyświetlane w czasie rzeczywistym, co poprawi doświadczenie użytkownika. Dodatkowo, wprowadzenie powiadomień w czasie rzeczywistym, które będą informować użytkowników o zakończeniu transakcji, ewentualnych problemach, czy sukcesach, bez potrzeby aktywnego sprawdzania statusu.

3. **Fallback Mechanisms**: Dodanie mechanizmów awaryjnych, takich jak ponowne połączenie WebSocket w przypadku utraty połączenia, aby zapewnić ciągłość działania. Implementacja alternatywnych kanałów komunikacji, takich jak HTTP polling, jako zapasowego rozwiązania w sytuacjach awaryjnych, gdy WebSocket jest niedostępny.

5.2 Rozwiązanie problemu przechowywania danych dotyczących ankiet w smart kontrakcie

Przechowywanie dużej ilości danych bezpośrednio w smart kontrakcie jest nieefektywne i kosztowne. Koszty związane z operacjami na blockchainie mogą znacząco wzrosnąć, gdy przechowywane są obszerne dane, takie jak szczegóły ankiet czy wyniki głosowań. Aby zminimalizować te koszty i jednocześnie zapewnić efektywne zarządzanie danymi, proponujemy dwa rozwiązania tego problemu:

5.2.1 Wykorzystanie IPFS (InterPlanetary File System)

IPFS to zdecentralizowany system plików, który umożliwia przechowywanie i udostępnianie danych w sposób skalowalny i odporny na cenzurę. Integracja IPFS z naszym systemem może znacząco zmniejszyć koszty związane z przechowywaniem danych na blockchainie.
Link do artykułu na temat IPFS

Implementacja:

- **Przechowywanie danych**: Dane dotyczące ankiet, takie jak pytania, odpowiedzi i wyniki, są przesyłane do IPFS. IPFS generuje unikalny hash (adres) dla każdej przesłanej ankiety, co pozwala na ich skalowalne i zdecentralizowane przechowywanie.

- **Hash Management**: Po przesłaniu danych do IPFS, zwracany jest hash, który jest zapisywany w smart kontrakcie. Smart kontrakt przechowuje jedynie odniesienia do danych, a nie same dane, co znacząco zmniejsza koszty przechowywania na blockchainie.
- **Retrieval**: Podczas odczytywania danych, aplikacja frontendowa pobiera je z IPFS, korzystając z hashy zapisanych w smart kontrakcie. Dzięki temu użytkownicy mogą łatwo uzyskać dostęp do pełnych danych ankiet, bez obciążania blockchaina.

Zalety:

- **Zmniejszenie kosztów**: Przechowywanie jedynie hashy w smart kontrakcie znacząco redukuje koszty transakcji na blockchainie.
- **Skalowalność**: IPFS zapewnia skalowalne przechowywanie danych, co jest szczególnie ważne przy rosnącej liczbie ankiet i głosowań.
- **Decentralizacja**: Dane przechowywane w IPFS są odporne na cenzurę i awarie centralnych serwerów, co zwiększa bezpieczeństwo i niezawodność systemu.

Wady:

- **Zarządzanie dostępem**: Konieczność zarządzania dostępem i bezpieczeństwem danych przechowywanych w IPFS.
- **Integracja**: Potrzeba dodatkowej warstwy integracji z IPFS, co może zwiększyć złożoność systemu.

5.2.2 Wykorzystanie Smart Contract Factory

Smart Contract Factory to wzorzec projektowy, który umożliwia tworzenie wielu instancji smart kontraktów z jednego "fabrycznego" kontraktu.
Link do artykułu na temat Smart Contract Factory

Implementacja:

- **Indywidualne kontrakty**: Zamiast przechowywać wszystkie dane ankiet w jednym kontrakcie, każda ankieta może być reprezentowana przez osobny smart kontrakt utworzony przez fabrykę. Każdy kontrakt ankietowy jest odpowiedzialny za zarządzanie własnymi danymi.
- **Factory Contract**: Fabryczny kontrakt zarządza tworzeniem nowych kontraktów ankietowych i przechowuje jedynie ich adresy. Dzięki temu główny kontrakt pozostaje lekki i efektywny.

- **Modularność**: Każdy kontrakt ankietowy zarządza własnymi danymi, co pozwala na bardziej złożone operacje bez zwiększania kosztów w jednym kontrakcie głównym. To podejście umożliwia również łatwiejsze skalowanie systemu, ponieważ nowe ankiety mogą być dodawane bez obciążania istniejącego kontraktu.

Zalety:

- **Redukcja kosztów**: Dystrybucja danych między wieloma kontraktami zmniejsza obciążenie jednego kontraktu, co może prowadzić do obniżenia kosztów operacyjnych.
- **Skalowalność**: Możliwość łatwiejszej aktualizacji i zarządzania poszczególnymi ankietami, co jest korzystne przy dużej liczbie ankiet i głosowań.
- **Elastyczność**: Modularność systemu pozwala na wprowadzanie zmian i aktualizacji w poszczególnych kontraktach bez wpływu na cały system.

Wady:

- **Złożoność zarządzania**: Konieczność zarządzania wieloma kontraktami może zwiększyć złożoność systemu i wymagać bardziej zaawansowanego monitorowania oraz utrzymania.
- **Liczba transakcji**: Potencjalnie większa liczba transakcji potrzebnych do zarządzania ankietami, co może zwiększyć koszty operacyjne w niektórych przypadkach.

5.3 Ułatwienie korzystania z aplikacji dla użytkowników bez wiedzy o blockchainie

Aby zwiększyć dostępność aplikacji Web3 Voting App dla szerokiego grona użytkowników, w tym osób bez wiedzy o technologii blockchain, zaproponowaliśmy kilka usprawnień. Te rozwiązania mają na celu uproszczenie procesu rejestracji, korzystania z portfela kryptowalutowego oraz opłacania transakcji, aby uczynić aplikację bardziej przyjazną dla użytkownika.

5.3.1 Generowanie portfela na podstawie adresu e-mail

Zamiast wymagać od użytkowników samodzielnego generowania i zarządzania portfelami kryptowalutowymi, możemy zautomatyzować ten proces poprzez generowanie portfela na podstawie adresu e-mail użytkownika. To podejście znacząco upraszcza proces rejestracji i korzystania z aplikacji dla osób, które nie są zaznajomione z technologią blockchain.

Implementacja:

- **Rejestracja użytkownika**: Podczas rejestracji użytkownik podaje swój adres e-mail (np. z domeny agh.edu.pl). Na podstawie tego adresu e-mail generowany jest portfel kryptowalutowy, co eliminuje konieczność samodzielnego tworzenia portfela przez użytkownika.
- **Generowanie portfela**: Wykorzystując deterministyczne algorytmy, takie jak BIP-39 (Mnemonic Phrase), system generuje klucze prywatne i publiczne portfela. Proces ten jest bezpieczny i umożliwia odtworzenie portfela na podstawie danych wejściowych (adres e-mail + hasło).
- **Przechowywanie portfela**: Klucze prywatne mogą być przechowywane bezpiecznie na serwerze backendowym, zapewniając ich dostępność i bezpieczeństwo. Alternatywnie, klucze mogą być zaszyfrowane i przechowywane po stronie klienta (np. w przeglądarce), co dodatkowo zwiększa prywatność użytkowników.

Zalety:

- **Prostota**: Użytkownicy nie muszą zarządzać własnymi portfelami kryptowalutowymi, co znacząco upraszcza proces rejestracji i korzystania z aplikacji.
- **Bezproblemowy onboarding**: Proces rejestracji staje się bardziej intuicyjny, co może przyciągnąć więcej użytkowników bez wiedzy o blockchainie.

Wady:

- **Bezpieczeństwo**: Potrzeba bezpiecznego przechowywania kluczy prywatnych, co może stanowić wyzwanie w kontekście ochrony danych użytkowników.
- **Prywatność**: Potencjalne obawy użytkowników dotyczące prywatności i bezpieczeństwa ich danych.

5.3.2 Przejmowanie opłat za gas (gasPrice) przez stronę

Opłaty za gas są jednym z głównych wyzwań dla użytkowników, którzy nie posiadają doświadczenia z technologią blockchain. Aby uprościć korzystanie z aplikacji i eliminować bariery finansowe, aplikacja może przejąć te opłaty, co uczyni cały proces bardziej przyjaznym dla użytkownika

Implementacja:

- **Sponsorowane transakcje**: Strona może przechwytywać i sponsorować opłaty za gas. Oznacza to, że backend aplikacji będzie zarządzał portfelem, z którego będą opłacane transakcje użytkowników. Dzięki temu użytkownicy nie muszą posiadać kryptowalut na pokrycie kosztów transakcji.

- **Meta-transactions**: Wykorzystanie meta-transakcji pozwala użytkownikom na podpisywanie transakcji bez konieczności posiadania Ethera (lub innej kryptowaluty). Użytkownik podpisuje transakcję off-chain, a backend przesyła ją na blockchain, pokrywając opłaty za gas.
- **Ekonomiczne modele**: Można zaimplementować różne modele ekonomiczne, takie jak sponsorowane transakcje przez reklamodawców, opłaty abonamentowe od użytkowników premium lub inne mechanizmy finansowania, które pokryją koszty gas.

Zalety:

- **Eliminacja barier finansowych**: Użytkownicy nie muszą posiadać kryptowalut, aby korzystać z aplikacji, co znacząco ułatwia dostęp do systemu.
- **Przyjazność dla użytkownika**: Uproszczony proces transakcji zwiększa komfort użytkowania i może przyciągnąć osoby niezaznajomione z technologią blockchain.

Wady:

- **Koszty operacyjne**: Aplikacja musi zabezpieczyć fundusze na pokrycie kosztów gas, co może stanowić wyzwanie finansowe.
- **Złożoność implementacji**: Implementacja mechanizmów sponsorowania transakcji i zarządzania funduszami wymaga dodatkowych zasobów i zaawansowanej infrastruktury.

6 Kamienie milowe projektu

Projekt Web3 Voting App przeszedł przez kilka kluczowych etapów, które pozwoliły na jego realizację w założonym czasie.

1. **27.03 - Wstępny research**: Przedstawienie konceptu aplikacji oraz dopracowanie szczegółów projektu.
2. **10.04 - Pierwsza implementacja**: Stworzenie dokumentu, w którym przeanalizowano oraz porównano dostępne technologie Web3 i blockchain.
3. **24.04 - Prototyp aplikacji**: Wzbogacenie dokumentu oraz przedstawienie prototypu aplikacji.
4. **15.05 - Rozwój aplikacji**: Dołączenie do dokumentu informacji opartych na pracy naukowej "Web3-Based Decentralized Autonomous Organizations and Operations: Architectures, models, and Mechanisms" oraz dalsza praca nad aplikacją.
5. **05.06 - Naprawa błędów**: Rozwój aplikacji i naprawa błędów.
6. **26.06 - Działająca aplikacja**: Finalizacja i uruchomienie działającej wersji aplikacji.

7 Dostęp do kodu źródłowego

Pełny kod źródłowy projektu Web3 Voting App jest dostępny w publicznym repozytorium na GitHubie. Można tam znaleźć wszystkie pliki oraz instrukcje dotyczące uruchomienia projektu lokalnie.

8 Podsumowanie

Web3 Voting App to nowoczesna aplikacja do zarządzania ankietami i głosowaniami, która wykorzystuje technologię blockchain do zapewnienia transparentności i niezmienności wyników. Dzięki zastosowaniu technologii takich jak Rust Axum, MongoDB, Next.js, Hardhat i Solidity, aplikacja jest wydajna i bezpieczna. Propozycje ulepszeń, takie jak optymalizacja nawigacji, wykorzystanie IPFS i Smart Contract Factory oraz uproszczenie korzystania z aplikacji dla użytkowników bez wiedzy o blockchainie, mogą znacząco poprawić doświadczenia użytkowników i obniżyć koszty operacyjne.