White Paper: Anonymous and Secure Voting System

Pôle IT: chrldb, pythack

30 novembre 2024

Résumé

This document details the creation of an anonymous and secure voting system designed for simplicity, transparency, and reliability. By leveraging modern technologies such as UUID-based anonymization, session validation, and responsive user interfaces, the system ensures both privacy and integrity. Additionally, the project emphasizes transparency by publishing its core logic as open source, inviting external audits and fostering user trust.

Table des matières

1 Introduction

In the digital age, voting systems must strike a delicate balance between security, anonymity, and ease of use. This project arose from the need for a trustworthy and efficient voting system that guarantees user privacy while maintaining the integrity of the voting process.

Our system addresses these challenges through a combination of robust architectural design and secure programming practices. The system is structured into three main components : the frontend, backend, and database. Each plays a critical role in ensuring the seamless functioning of the voting process while adhering to strict privacy standards.

This document outlines the design choices, implementation strategies, and future improvements for this voting system, providing a comprehensive understanding of its operation.

2 System Architecture

The architecture is built to ensure a clear separation of responsibilities, enhancing both security and scalability. Figure 1 illustrates the high-level architecture.

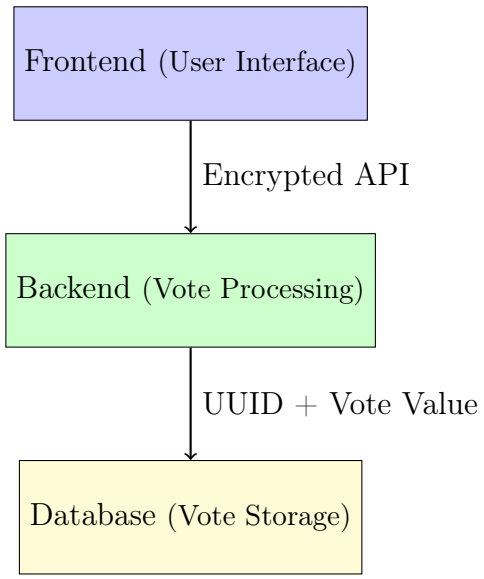

FIGURE 1 – System Architecture

2.A Frontend

The frontend provides an intuitive user interface for casting votes. Designed to be responsive, it ensures compatibility with devices of all screen sizes. A confirmation popup is integrated to prevent accidental submissions, enhancing the user experience.

2.B Backend

The backend is responsible for processing votes securely. It generates unique UUIDs for each vote, ensuring anonymity. It also validates user sessions to prevent duplicate voting without storing identifiable data.

2.C Database

The database stores only anonymized vote data, separated from user information. This ensures that no vote can be traced back to an individual, even in the event of a data breach.

3 Implementation Details

3.A Anonymity with UUIDs

Anonymity is achieved through the use of UUIDs, which are randomly generated identifiers unique to each vote. The following code illustrates the generation of a UUID :

```
function uuidv4() {
    $data = random_bytes(16);
    $data[6] = chr(ord($data[6]) & 0x0f | 0x40); // version 4
    $data[8] = chr(ord($data[8]) & 0x3f | 0x80); // variant
    return bin2hex($data);
}
```

Listing 1 – UUID Generation

This UUID is stored alongside the vote value in the database, with no association to user data.

3.B Session Validation

To prevent duplicate voting, the backend flags a user's session once a vote is submitted. This ensures that each user can vote only once, without compromising anonymity.

3.C Responsive Design

The frontend is designed with a mobile-first approach, ensuring that the voting experience is seamless on devices of all sizes. The confirmation popup adds an extra layer of user interaction to avoid unintentional votes.

4 Challenges and Solutions

4.A Ensuring Anonymity

One of the primary challenges was ensuring that votes remained completely anonymous. This was addressed by separating user sessions from vote records and using UUIDs for identification.

4.B Preventing Duplicate Votes

Session validation was implemented to prevent multiple votes from the same user. This ensures fairness without storing sensitive user data.

4.C Building User Trust

Transparency was prioritized by publishing the core logic as open source. This allows external audits and fosters confidence in the system's integrity.

5 Conclusion

This voting system effectively balances privacy, security, and usability. By utilizing UUIDs for anonymization, session validation to prevent duplicate votes, and a responsive design for user interaction, it provides a robust solution for secure digital voting. The open-source nature of the project further enhances transparency and trust, inviting scrutiny and improvements from the community.

References

— GitHub Repository: Full Source Code
— MDN: HTTPS Overview
— PHP Documentation: random_bytes()