PROJECT DESIGN DOCUMENT: VOTING APPLICATION

GROUP MEMBERS:

PAUL SANSAH GYREYIRI

SIVANEE SANTHALINGAM SIVAKUMAR

1. INTRODUCTION

The voting application is a web-based software that allows members of an organization, club, or student association to vote on various topics or polls. The application will ensure the voting process is secure and anonymous and automatically count votes in real-time. Additionally, the usernames and IDs attached to votes will be encrypted while detecting tampered ballots.

2. COMPONENTS

The voting application comprises the following components:

i. User Authentication Component
ii. Voting Component
iii. Vote Counting Component
iv. Data Storage Component

2.1 User Authentication Component

This component is responsible for user registration, login, and user session management. The authentication component described in this section is assumed to be the same authentication system used for managing user access for the organization and the entire institution. The system is supposed to have been used for signing up members during membership registration into the organization. The passwords will be stored in a hashed form using a secure hashing algorithm. The component will check the user's credentials during login and issue session tokens that will be used to authenticate the user's subsequent requests.

2.1.1 Informal Contract

i. The user authentication component must ensure that user passwords are hashed securely.
ii. The component must prevent unauthorized access by ensuring users provide valid credentials to access protected resources, including the voting component.
iii. Before permitting access to secured resources, the component must ensure that session tokens are current and have not expired.

2.2 Voting Component

This component creates new polls and allows users to vote on existing ones. The system administrator will be responsible for creating new polls, which will be visible to all registered users. Users can only vote once and only once on a particular election. Also, the component will make sure that each vote is assigned a unique ID and the hashed user details such as username. The voting component must include a MAC to the encrypted voted user details to detect any tampering with votes.

2.2.1 Informal Contract

i. The voting component should ensure users can only vote once on a particular poll.
ii. The voting component must ensure that user votes are secure and anonymous.
iii. The usernames and IDs attached to votes should be encrypted, not who or what they voted for.
iv. The voting component must include a MAC to the encrypted voted user details to detect any tampering with votes.

2.3 Vote Counting Component

This component is responsible for counting the votes in real-time for each poll. This component must present the number of votes cast, and the results, both in numbers and percentages, throughout and after the voting process. This component is also responsible for periodically generating MACs for votes and comparing them with stored MACs to detect tampered ballots.

2.3.1 Informal Contract

i. The vote-counting component must ensure that each poll's results are accurate, reliable, and up-to-date.
ii. The vote-counting component must detect tampered votes and raise flags for those votes.

2.4 Data Storage Component

This component would store user credentials, vote data, and other relevant application data. The data would be securely stored in a database to ensure the privacy and security of user information. This component would also provide that information in the votes table is read-only, adding an extra layer of data confidentiality and integrity.

2.4.1 Informal Contract

i. The data storage component must ensure that user data is securely stored and protected from unauthorized access.

ii. The database must be backed up regularly to ensure data availability and reliability.
iii. The storage component must also ensure that data in the votes table are not editable.

2.5 Integration of Components

The components of the voting application integrate as follows:

The user authentication component authenticates users before allowing them to access the voting component. The voting component interacts with the user authentication component to verify that users are authenticated before allowing them to vote. The voting component allows users to vote, hash user details, generate a MAC, and store the hashed details along with the vote and the corresponding MAC. The vote-counting component retrieves the votes and calculates each poll's total number of votes and other statistical information while detecting tampered ballots. In comparison, the data storage component would store user credentials, vote data, and other relevant application data.

Furthermore, a type-first-driven design pattern would be adapted to ensure written codes are less buggy and simple. Input from users should be validated with an express-validator to ensure that the application only accepts data of a specific type. Results pulled from the data would be mapped to their respective defined types.

3. SECURITY CHALLENGES

The proposed voting application faces several security challenges, such as:

i. Ensuring that user votes are anonymous and cannot be traced back to the user. This threat still exists only when an attacker gets the encryption key that was used to encrypt the hasvoted table.
ii. Detecting any attempts to tamper with votes and reverting to the original. (This threat has been addressed by preventing update or deleting of vote table.

4. ASSUMPTIONS

We assume the following while designing the application:

i. The users' passwords will be hashed before being stored in the database.
ii. The application will be accessed via a secure HTTPS connection.
iii. The users' devices are free of malware and viruses that could compromise their data.

5. CONCLUSION

The proposed voting application would allow users to participate in an online poll securely and anonymously. The application's user authentication, voting, vote counting, and data storage components would work together seamlessly to provide a smooth user experience. The security concerns of the application must be addressed to a reasonable extent. While there still exists a slight chance that an attacker can access the .env file and decrypt user details, and watch incoming traffic and whom they voted for, protecting user details is out of this project's scope since we assume user management is the duty of the mother institution.