CSE 312 SOFTWARE ENGINEERING AND PROJECT MANAGEMENT

Project title: Election System

22 August 2024

Roles:

2022BCS0118 - Ali Shaik (Backend)

2022BCS0184 - Manoj (Database)

2022BCS0217 - Shoury (Frontend)

2022BCD0058 - Yeshwanth (Backend)

FEASIBILITY STUDY REPORT:

1.Technical Feasibility:

a) Technology Stack
b) React (Front-end)
c) FastAPI (Back-end)
d) PostgreSQL (Database)
e) Scalability and Load Balancing Tools
    - Celery: For handling background tasks and distributed workloads, especially useful in processing-intensive scenarios.
    - Redis: As an in-memory data store, useful for caching and session management to reduce load on the database.
    - Horizontal Scaling: Deploy multiple instances of FastAPI and PostgreSQL, using load balancers like NGINX or HAProxy to manage traffic distribution.
    - Database Sharding/Replication: Implement sharding or replication strategies for PostgreSQL to handle large datasets and high read/write traffic.
    - Caching: Use Redis for caching frequently accessed data to reduce load on the database.
    - Asynchronous Processing: Utilize Celery for background tasks, such as processing votes, which can be distributed across multiple workers

2.Operational Feasibility:

a) Security:
    - **Data Integrity**: The system must ensure that votes cannot be tampered with or altered. Secure databases can be used to maintain the integrity of vote data.
    - **Voter Authentication**: Robust methods to verify voter identity are crucial to prevent fraud, such as multi-factor authentication (MFA) or biometric verification.
b) Scalability:
    - **Handling High Traffic**: The system should be able to handle large numbers of users simultaneously, especially during peak voting times. Cloud infrastructure or distributed systems can help with scalability.
    - **Load Balancing**: Implement load balancing to distribute user requests across multiple servers, preventing overload.
c) User accessibility:
    - **Ease of Use**: The interface should be user-friendly and accessible to all.
d) Public Trust and Adoption
    - **Transparency**: The system must be transparent, with mechanisms for public and independent verification of results. Blockchain technology can enhance transparency by providing an immutable record of votes.
    - **Voter Confidence**: Building public trust in the system is essential. This can be achieved through rigorous testing, pilot programs, and independent audits.
e) Technical Reliability:

- **Downtime and Failures**: The system must have minimal downtime, with backup systems in place to handle any technical failures. Regular stress testing and disaster recovery plans are crucial.

3.Economic Feasibility:

Development Costs

Personnel:

Front-end Developers (React): developers for building and maintaining the user interface.

Back-end Developers (FastAPI): developers for API development, integration, and server-side logic.

Database Administrators (PostgreSQL): developers for database design, optimization, and maintenance.

DevOps Engineers: engineers for managing deployment, and scaling strategies.

QA Engineers: testers for ensuring the system is bug-free and secure.

Objectives:

1. Ensure Security and Integrity

Implement robust security measures to protect the election process from fraud, unauthorized access, and tampering. This includes encryption, secure authentication, and audit trails.

2. Enhance Accessibility and Usability

Make the voting process more accessible and user-friendly for all eligible voters, including those with disabilities or limited technological skills. This may involve providing alternative voting methods, such as mail-in ballots or electronic voting machines.

3. Optimize Usability

Design an intuitive user interface that simplifies the voting process for all users, reducing the chances of user errors and ensuring a smooth voting experience.

4. Maintain Transparency

Implement features that ensure transparency in the voting process, such as real-time vote counting, and provide voters with a means to verify their votes have been correctly registered.

5. Enable Secure Remote Voting

Explore and integrate secure remote voting options for absentee voters, ensuring their ability to participate in the election from different locations.

Functionalities:

User Registration and Authentication

1. Voter Registration:

Allow eligible users to register with the system, providing necessary details such as identification, age verification, and residential information.

Administrator Login: Provide a secure login portal for election officials to manage and monitor the election process.

Multi-factor Authentication: Implement additional layers of security for both voters and administrators, such as OTP (One-Time Password) or biometric verification.

2. Candidate Management:

Candidate Registration: Allow candidates to register and submit their profiles, including their platform, party affiliation, and any relevant documents.

Approval/Verification Process: Enable administrators to verify and approve candidate registrations.

3. Voting Process

Ballot Creation and Distribution: Automatically generate and distribute ballots to registered voters based on their eligibility (e.g., region, constituency).

Vote Casting: Allow voters to securely cast their votes. Ensure that votes are encrypted and stored securely to prevent tampering.

Vote Confirmation: Provide a confirmation receipt to voters after their vote is successfully cast.

4. Audit and Reporting

Audit Trails: Maintain detailed logs of all actions performed within the system, including vote casting, candidate registration, and administrative actions, to ensure transparency and accountability.

Reporting Tools: Generate comprehensive reports on various aspects of the election, such as voter turnout, demographic breakdowns, and election outcomes.

5. Data Encryption and Security

Secure Data Storage: Ensure all data, including voter information and votes, is stored securely with encryption.

Data Backup and Recovery: Implement mechanisms for regular data backups and recovery to prevent data loss in case of system failures.