Voter Authentication Service with Microservices

This document outlines the architecture and implementation of a voter authentication service based on a microservice architecture. It focuses on the voter registration and authentication process, leveraging modern technologies for image recognition, facial verification, and security.

# Main Users

The primary users of the system include:

- Voters: Individuals who will use the application to register, authenticate, and vote securely.

- Politicians/Candidates: Users who will maintain social media-like profiles where they can post campaign content such as videos, articles, and updates.

- Administrators: System administrators responsible for managing elections, user permissions, and ensuring system integrity.

# Microservice-Based Architecture

The application is designed using a microservice architecture, where different services handle distinct functionalities. Each service communicates via REST APIs and can be independently scaled or maintained. The key microservices include:

## Voter Authentication Service

The voter authentication service is responsible for validating users before they can participate in elections. It involves a two-step authentication process:

1. \*\*Upload Identity Card Photo\*\*: The user uploads a photo of their identity card. This image is processed using Optical Character Recognition (OCR) to extract details such as name, date of birth, and ID number.

2. \*\*Upload Selfie\*\*: The user uploads a selfie, which is compared to the photo on their ID card using a facial recognition service like AWS Rekognition or Azure Face API.

If the identity is validated, the service issues a JSON Web Token (JWT) to authenticate the voter in future actions.

# Database Design

The database is divided into multiple tables to manage various aspects of the application. Key tables include:

- \*\*Users\*\*: Stores information about the voters and candidates, including personal details, account status, and profile information.

- \*\*Elections\*\*: Contains data related to each election, such as election dates, participating candidates, and results.

- \*\*Votes\*\*: Stores voting records, ensuring that each voter can cast only one vote per election.

- \*\*Campaign Posts\*\*: Allows politicians to post videos and other campaign content, similar to a social media feed.

## ID Card Processing

The ID card processing service receives an image of the user's identity card and performs the following tasks:

- \*\*Validation\*\*: Ensures the uploaded file meets the required format (JPEG, PNG) and quality standards (resolution, size).

- \*\*OCR\*\*: Optical Character Recognition is applied to extract key details from the card. Tools like Tesseract OCR or Google Vision API can be used.

## Selfie Verification

The selfie verification service receives a selfie and compares it against the photo from the ID card using a facial recognition API. Services like AWS Rekognition or Azure Face API are capable of performing this comparison with a defined similarity threshold.

## Token Issuance

Once the user's identity is verified, a JWT is generated using a service such as jjwt in Java. This token will be used for authentication in future interactions with the app.

# Election and Campaign Management

This service allows politicians to manage their campaign profiles, similar to social media accounts. They can upload campaign videos and posts, which are visible to voters. Voters can view campaign materials before casting their vote.

# Security Considerations

Given the sensitivity of election data and voter authentication, several security measures must be implemented:

- \*\*Encryption\*\*: All communication between services and user data must be encrypted using HTTPS and secure protocols.

- \*\*Token Expiration\*\*: JWTs must have a defined expiration time to ensure session security.

- \*\*Multi-Factor Authentication\*\*: Optionally, multi-factor authentication (MFA) can be added for an extra layer of security during login.

# Conclusion

This voter authentication service is designed to ensure secure and reliable participation in elections, leveraging modern technologies like OCR, facial recognition, and JWTs in a microservice-based architecture.