**Biometric Sequential Authentication System for Secure, Transparent & Digitized Elections**

**Table of Contents**

1. Project Overview
2. Workflow and Process Flow

2.1 Detailed Process Flow

2.2 Flowchart

3. Fingerprint Registration & Digital Slip Simulation

4. Operational Voting Process

5. Circuit Interfacing and Hardware Setup

6. Traditional System vs. New System Budget Comparison

7. Web3 Implementation and DID-based Vote Count

8. Risk Analysis & Mitigation Strategies

9. Future Improvements & Scalability

10. Final Review and Conclusion

1. **Project Overview**

**Objective:** This project aims to modernize the electoral process by integrating digital communication, dynamic shift allocation, robust biometric authentication, and decentralized identity (via Web3). The system replaces physical voter slips with digital messages, allocates voting shifts dynamically using a real-time clock (RTC), and uses dual-stage fingerprint verification to ensure that only verified voters cast their vote. In addition, vote records (including vote counts stored as strings whose hashes are converted into decentralized identifiers) are registered on a blockchain to provide an immutable audit trail, ensuring transparency and security.

**Key Features:**

**Digital Slip Distribution:** Voters receive SMS-based notifications (or simulated messages) that specify their assigned voting slot.

**Dynamic Shift Allocation:** Voters initially choose the first slot; if full, non‑voters are re‑allocated to subsequent shifts with continuous reminder messages.

**Dual-Stage Biometric Verification:** Utilizes two fingerprint sensors—one for entry (employee and voter index fingerprint) and one for vote confirmation (voter thumb fingerprint).

**Push Button Voting:** Voters select their desired political party using dedicated push buttons.

**Web3 Integration:** Unique fingerprint IDs (hashed) are stored on a blockchain as decentralized identifiers (DIDs), and vote counts (stored as strings) are hashed to form part of the DID, ensuring an immutable and transparent record.

**Cheating Detection:** The system flags duplicate or fraudulent voting attempts.

**Government Incentives:** Only voters who cast a valid vote receive benefits (e.g., tax waivers/subsidies).

1. **Workflow and Process:**

**Registration Phase:**

**Fingerprint Enrollment:** Employees and voters have their fingerprints captured twice using an AS608 sensor. Each fingerprint is stored with a unique ID (e.g., 101 for Employee 1, 201 for Voter 1).

**Digital Slip Simulation:** After successful registration, a digital slip message is simulated (via Serial Monitor) prompting voters to select their preferred voting slot. In our prototype, all initially choose Slot 1.

**Voting Phase:**

**Digital Slip & Reminders:** On election day, voters receive their digital slip message indicating their assigned slot. Non‑voters receive periodic reminders and are re‑allocated to subsequent shifts.

**Shift Allocation:**

Shift 1 (9:00–11:00 AM): Voters who selected Slot 1 vote normally.

Shift 2 (11:00–1:00 PM): Voters who didn’t vote in Shift 1 are re‑allocated; if a cheating case occurs, it is flagged.

Shift 3 (1:00–3:00 PM): Additional re‑allocation occurs; for example, the system detects and flags cheating.

Shift 4 (3:00–4:00 PM): The remaining valid voters updated address vote.

**Polling Station Process:**

**Entry Verification:** The employee (via Sensor 1) and the voter (via index fingerprint on Sensor 1) are authenticated.

**Vote Selection:** The voter selects a political party using push buttons.

**Vote Confirmation:** The voter confirms by scanning their thumb using Sensor 2.

**Error and Cheating Handling:** Any discrepancies or duplicate votes are flagged, and cheating cases are logged (leading to potential arrest in simulation).

**Post-Voting:** Voters who have not cast a vote continue to receive reminders and ultimately forfeit government incentives.

**Election Conclusion:**

After 4:00 PM, the election ends. The system tallies the final vote counts, declares the winning party, and displays which voters receive government incentives.

1. **Coding Area:**

**Registration & Digital Slip Simulation Code**

**Purpose:** This code is designed to enroll fingerprints for voters and employees and then simulate sending a digital slip message. It is run prior to the voting process, ensuring that every user’s biometric data is captured and stored with a unique ID.

**Key Components and Workflow:**

**Fingerprint Enrollment:**

**Initialization:** The code initializes the AS608 fingerprint sensor using the SoftwareSerial library (on pins 2 and 3). It begins by setting the sensor’s baud rate (57600) and verifies communication with the sensor.

**User Input:** Through the Serial Monitor, the user is prompted to enter a unique ID (e.g., 101 for an employee, 201 for a voter). This unique ID will be associated with the fingerprint template.

**Fingerprint Capture:** The code then asks the user to place their finger on the sensor. It captures the fingerprint image and converts it into a digital template using image2Tz(1).

**Re-capture for Accuracy:** After the first capture, the user is prompted to remove their finger, and after a short delay, to place it again. The second capture is used to confirm the fingerprint by converting it using image2Tz(2).

**Model Creation and Storage:** The sensor compares the two captured templates using createModel(). If they match, the template is stored in the sensor’s memory under the unique ID using storeModel(id). This ensures that only a correctly captured fingerprint is registered, reducing the risk of errors during later authentication.

**Digital Slip Simulation:** Once the fingerprint is successfully stored, the code simulates sending a digital slip by printing a message on the Serial Monitor. The simulated digital slip asks the voter to select their preferred voting slot. For our prototype, all voters are assumed to choose the first slot.

**Operational Voting Process Code**

**Purpose:** This code manages the entire voting process on election day. It handles dynamic shift allocation based on time, verifies both employee and voter identities using two fingerprint sensors, facilitates vote selection via push buttons, confirms votes, and finally tallies the vote counts. It also integrates reminder messages and basic cheating detection.

**Key Components and Workflow:**

**RTC-Based Shift Allocation:**

**Timekeeping:** The DS1307 RTC module (connected via I2C to A4 and A5) is used to obtain the current time.

**Shift Determination:** The function getCurrentShift() compares the current hour against predefined intervals (Shift 1: 9–11 AM, Shift 2: 11–1 PM, Shift 3: 1–3 PM, Shift 4: 3–4 PM) to decide which shift is active.

**Digital Slip/Reminder Messaging:**

The code simulates sending a digital slip or reminder message (via the Serial Monitor) based on the current shift. This reminds voters about their assigned slot and encourages them to cast their vote if they haven't already.

**Entry Verification (Fingerprint Sensor 1):**

**Employee Verification:** The designated employee for the current shift (with a pre-assigned ID, e.g., 101 for Shift 1) verifies their identity by scanning their fingerprint.

**Voter Verification:** After successful employee verification, the voter scans their fingerprint (using an index finger) on the same sensor. The function entryVerification() checks if the scanned voter fingerprint matches the expected ID for that shift.

**LED Feedback:** LEDs (green, yellow, and red on pins 6, 7, and 8) provide immediate visual feedback—green for successful verification, yellow for employee errors, and red for mismatches or illegal attempts.

**Vote Selection & Confirmation:**

**Push Button Selection:** After entry verification, the voter selects their political party using push buttons (connected to pins 12, 13, A0, A1). The selected party is captured once a button is pressed.

**Second Fingerprint Verification:** Before finalizing the vote, the voter confirms their identity by scanning their thumb using the second AS608 sensor (connected via SoftwareSerial on pins 4 and 5). The function voteVerification(selectedParty) compares the live thumb fingerprint with the expected voter ID.

**Cheating Detection:** The code contains logic to flag cheating attempts (e.g., duplicate voting or impersonation) by comparing scanned IDs with expected values.

**Vote Tally:** Once the vote is confirmed (indicated by a green LED on pin 10), the vote count for the selected party is incremented. Votes are stored as numbers, and later, a further Web3 module (not in this code) can convert these counts to strings and hash them for DID-based recordkeeping.

**End of Election & Final Results:**

When the RTC indicates that the election period has ended (after 4 PM), the system stops accepting votes. The final vote counts are printed on the Serial Monitor, the winning party is declared, and a message indicates that voters who cast their vote will receive government incentives.

1. **Web3 Implementation Using Unique Fingerprint ID and DID-based Vote Count**

**Overview:**

To further enhance security and transparency, the project integrates Web3 technology:

**Decentralized Identity (DID):** Each voter’s unique fingerprint ID is hashed (using keccak256, for example) off-chain and registered as a decentralized identifier (DID) on a blockchain. This ensures that the voter’s identity is tamper-resistant and verifiable.

**Immutable Audit Trails:** When a vote is cast, the system creates a blockchain transaction that includes a cryptographic hash of the vote data along with the voter’s DID. Vote counts for each party are stored as strings whose hashes are computed and recorded on-chain. This creates an immutable audit trail that anyone can verify without compromising privacy.

**Smart Contracts:** Smart contracts enforce the rule that each DID can vote only once, record votes securely, and trigger incentive distributions upon election completion.

**Benefits**: This integration prevents fraud, offers public verifiability, and automates critical election processes such as vote tallying and incentive distribution.

1. **Final Review:**

**Uniqueness and Novelty:**

The system uniquely integrates digital slip distribution, dynamic shift allocation, dual-stage biometric verification, and blockchain-based Web3 integration. Its use of DIDs for secure, decentralized identity management sets it apart from traditional voting systems.

**Relevance to the Real World:**

Addressing critical issues like voter fraud, inefficient slip distribution, and manual verification errors, the system is highly relevant to current electoral challenges.

**Future Scope – Web3 Integration:**

The Web3 component not only enhances security through immutable audit trails and smart contract enforcement but also positions the system for future scalability and wider adoption in digital governance.

**Design:**

With a modular hardware and software architecture, the system is user-centric and easy to interface with. It offers robust error handling and dynamic re‑allocation, ensuring a smooth voter experience.

**Social Impact:**

By ensuring transparent, secure, and accessible voting, the system builds public trust, encourages civic participation, and offers tangible economic incentives to voters, ultimately strengthening democratic processes.

1. **Conclusion:**

The Biometric Sequential Authentication System for Secure, Transparent & Digitized Elections is a comprehensive, future‑ready solution that transforms traditional electoral processes. By combining advanced biometric verification, dynamic shift management, digital communication, and decentralized blockchain technology, it ensures secure and fraud‑resistant voting. With added features like risk mitigation, scalability planning, and a decentralized audit trail, this system not only addresses current challenges but also sets a strong foundation for the evolution of digital democracy. This project is poised to significantly enhance voter trust, operational efficiency, and overall democratic integrity.