Project 2: Testing a Software System

Prepared by Group 23 - Immersive Real Estate System Team

Project being Tested: DemocracyNow, a capability to securely and anonymously cast your vote for candidates on the ballot that corresponds to your voting precinct, via your smartphone.

Test Scope

This section defines which functional areas of the DemocracyNow system that will be tested, which items are out of scope, and how non-functional requirements will be addressed.

1. Summary of DemocracyNow's Purpose

DemocracyNow is a secure, accessible digital voting system designed to modernize elections and increase voter participation. It allows eligible citizens to cast their ballots electronically through web and mobile platforms while ensuring privacy, authenticity, and verifiable integrity. The system combines multi-factor authentication, encryption, and blockchain technology to guarantee that every vote is both secure and anonymous, maintaining public trust in a fully verifiable election process.

2. In-scope Targets:

1. **Authentication and Anonymous Credential Issuance**
   Verify secure three-factor authentication, token generation, and identity separation between authentication and voting components.

2. **Ballot Delivery and Four-step Vote Confirmation**
   Confirm that ballots are correctly assigned by jurisdiction, accessible through all platforms, and follow the deterministic four-stage submission workflow. This workflow is described as "select" → "review" → "confirm" → "submit".

3. **Encryption and Blockchain Ledger**
   Validate end-to-end encryption, blind signature implementation, and immutable blockchain storage to ensure vote confidentiality and tamper-proof records.

4. **Verification Portal and Vote Receipts**
   Test that cryptographic receipts are issued correctly and can be verified independently through zero-knowledge proofs without revealing vote contents.

5. **Accessibility and Performance Requirements**

Confirm compliance with WCAG 2.1 AA accessibility standards (screen readers, high contrast, text scaling, multi-language support) and system performance expectations (availability, scalability, and response times during peak voting).

3. Out-of-scope Areas:

The following areas are intentionally excluded from this test plan based on assignment scope:

1. **Unit tests:** Developer-level tests for individual modules or classes are outside the integration and validation focus.

2. **Expected results or execution data:** Specific test data, pass/fail criteria, and actual outcomes will not be included.

3. **Resource and schedule details:** Personnel assignments, tools, and test scheduling are not part of this plan.

4. Scope boundaries

This section describes what is being verified at system level and what's not.

At the system level, the test plan will verify that DemocracyNow functions as a cohesive, secure, and accessible platform, meaning all major modules (authentication, ballot management, encryption, ledger, verification) integrate correctly and meet their specified functional and performance goals.

It will not verify internal code logic, developer-specific unit functionality, or infrastructure deployment performance beyond defined system requirements. The goal is to ensure that the integrated system behaves correctly and securely as a whole, validating trust, integrity, accessibility, and auditability in real-world election use scenarios.

Test Plan

This section outlines how the testing will be approached along with the rationale for the given approach.

1. Integration Test Strategy

We recommend a bottom-up integration approach. DemocracyNow's risk is concentrated in backend cryptography, data separation, and immutable recording. By bringing up the platform from the edge/gateway to identity, then to the preparation of ballots, then to casting

ballots, encrypting the transmission and recording the transaction on the ledger, we minimize the number of UI stubs we'd otherwise need and prove the identity-vote separation early. This also lets us validate opaque tokens and no-PII crossing before and end-to-end flows.

2. Integration Phases (bottom-up)

This section demonstrates how the bottom-up test plans will be broken into phases and sequenced accordingly.

Phase 1: Backend (Gateway) Integration Testing

Phase one will include testing of the API gateway and observability of the transmissions, including TLS, routing, security headers, rate limits, passive logging.

Phase 2: Identity and Anonymous Credential Boundary

Phase two will address identity and anonymous credential boundary isolation, screen lockdown, and signaling

Phase 3: Ballot Delivery and Four-Stage Confirmation
FR-5 to FR-9 and FR-8.1-8.4 Deterministic workflow across PWA/iOS/Android with WCAG behaviors.

Phase 4: Cast Ballot, Encrypt, Ledger and Audit
FR-10, FR-12 to FR-16, FR-13/FR-19/FR-20 key rotation and immutable record.

Phase 5: Vote Tally, Verification Portal, and Pollbook Sync
FR-17, FR-18, FR-25, and independent verification (FR-20/FR-16)

Phase One: Backend (Gateway) Integration Testing

Goal: Establish a correct, observable network edge with mock downstream; no PII leakage.

Test Case ID	Test Case Objective	Test Case Description
IT-1a	Verify system uses proper GET, POST, PUT, and DELETE request methods, adhering to standard REST procedures	The system exposes endpoints with a sensible URL hierarchy. Query parameters are only used for optional fields. GET requests should never contain any data in their body. The system must remain stateless - the server must not have any memory of the user's session, but can still store information about the user.

Test Case ID	Test Case Objective	Test Case Description
IT-1b	Keep an audit log of all user and system activity to monitor for security breaches, or attempts to break in.	Verify that a well-known and reliable logging system is in use and logs to accessible endpoints, such as a text file left on the filesystem, or a syslog entry. All user actions, whether they fail or succeed, must be entered into this log. System operations that were not initiated by a user must also be kept in this log. Check that the log is denoted with 5 levels: DEBUG, INFO, WARN, ERROR, FATAL. DEBUG logs are minute details useful for debugging, Ensure that this is only shown in development builds, or when a specific flag is set. INFO logs will contain user and system operation notices, such as a successful or unsuccessful login. WARN will alert that conditions may not be ideal for a certain situation. ERROR denotes something that went wrong, but was recoverable, such as a bad request or timeout. FATAL must only be logged right before the system crashes and goes offline.
IT-1c	Keep security breaches localized to reduce data leak volume	Verify that all outside data is sanitized and has sensible restrictions to prevent common attacks such as SQL-injection or buffer overflows. For example, a username does not need to exceed 20 characters.
IT-1d	Validate authentication session is terminated after authentication is successful	User authenticates as normal. Attempting to navigate back to previous authentication pages or closing browser should terminate session and not allow user to go back
IT-1e	Keep security breaches localized to reduce data leak volume	Verify that all outside data is sanitized and has sensible restrictions to prevent common attacks such as SQL-injection or buffer overflows. For example, a username does not need to exceed 20 characters.
IT-1f	Ensure identity never co-resides with ballot artifacts in Phase-5 subsystems.	Inspect payloads, ledger records, receipts, proofs, and audit logs; attempt to pass identity claims downstream and confirm strict rejection; verify only opaque tokens/communities are present.
IT-1g	Ensure voting workflow is efficient enough to be completed in a timely manner	Verify that users using the system take, on average, 5 minutes to sign up for an account and submit a vote to the system.
IT-1h	Verify the system maintains full functionality when operating at extremely low bandwidth.	Throttle network speed to ≤56 Kbps and test ballot loading, selection, submission, and confirmation steps to ensure no failures or timeouts occur.
IT-1i	Ensure reliable and very high uptime	Verify that deployment spans multiple regions, and that each region has fallbacks in case any of the systems go down. Also ensure that regions can fallback onto other regions. These systems must be able to communicate with each other to ensure consistency across data.
IT-1j	Ensure high concurrent user count does not affect usability	Verify that deployment automatically scales to multiple nodes to compensate for large bursts of users, and just

Test Case ID	Test Case Objective	Test Case Description
		also automatically downscales to fewer nodes to prevent unnecessary cost for periods of low traffic.
IT-1k	Ensure all vote selections and confirmation data are reliably stored and synced to the backend without loss.	Perform ballot submission and monitor the database for correct write operations, sync operations, retry handling, and resilience against intermittent connectivity.
IT-1l	Verify authentication, ballot management, and vote processing modules can be hot-swapped	Ensure that, while the system is still running, the authentication, ballot management, vote processing modules can be swapped with newer or alternative versions, with minimal effect on downtime (<10 seconds)
IT-1m	Ensure microservices automatically scale depending on demand	Verify that authentication, ballot management, and vote processing microservices are appropriately and automatically scaled up and down according to load, with no effect on downtime.
IT-1n	Ensure separation of concerns	Verify that microservices do not depend on one another, and could effectively perform their task without any other microservice being available. Ensure that each microservice does not take in data that is irrelevant to its operation. For example, ballot management does not need the user's password.
IT-1o	Comply with state and federal regulations	Ensure that the system complies with requirements of the Help America Vote Act and guidelines set by the Election Assistance Commission Voting System Standards. Additionally, verify that the system does not infringe on state regulations for electronic voting systems.

Phase 1 tests cover the following functional and non-functional requirements:

Requirement ID	Requirement Description
FR-21	RESTful API Gateway
FR-23	Passive logging microservice
FR-24	Defense-in-depth layered security
NFR-4	24-hour credential expiration
NFR-6	Defense-in-depth layered security
NFR-8	No PII storage with vote data
NFR-15	5-minute completion time (95% users)
NFR-16	Low-bandwidth support (56 Kbps)
NFR-17	99.99% uptime
NFR-18	10M concurrent users
NFR-19	3-second response time

Requirement ID	Requirement Description
NFR-21	Independent module modification
NFR-22	Independent scaling by load pattern
NFR-23	Clear separation of concerns
NFR-27	HAVA and EAC compliance

Phase Two: Identity and Anonymous Credential Boundary

Goal: Prove three-factor authentication and issuance of time-limited opaque credentials; ensure identity never crosses into vote flow.

Test Case ID	Test Case Objective	Test Case Description
IT-2a	Ensure the process to authenticate voter identity grants access only after 3 stage authentication	Run and pass digital signature, 2FA, and secure identification tests in sequence using the same test user. An eligible citizen should pass all three stages, and an ineligible person (e.g: noncitizen or felon) should fail and be denied access.
IT-2b	Given a citizen ID, ensure that system correctly verifies eligible voters	Upload multiple forms of valid government IDs. The system should check the authenticity of the ID, and if user is an eligible voter, digital signature verification should pass. Test using expired ID, unauthentic ID, and ID for ineligible person, these cases should prevent the user from progressing to the next authentication step.
IT-2c	Ensure the two factor authentication process works correctly	Initiate 2FA process. Verify that the system generates a OTP and delivers it through the configured channel to the user's second device. Ensure that entering said OTP approves authentication. Entering incorrect/expired/used OTP should fail and prevent user from accessing voting.
IT-2d	Ensure biometric scanning process works	Authorized user scans fingerprint and face to access the app. System should verify identity and allow user to access the app. Attempt login with unauthorized users using face and fingerprint. App should not allow entry.
IT-2e	Voter registration database validation	Attempt to authenticate eligible voter using normal process, voter database should return eligible and user should be allowed to continue authentication. Non-existent voters and ineligible people should be tested in the same way and denied.

Test Case ID	Test Case Objective	Test Case Description
IT-2f	Opaque token authentication verification	Upon completion of all authentication steps, verify that the system generates a time-limited anonymous credential that contains no personally identifiable information. Verify that token is not JSON, and the token enables access to ballot but does not expose any identity attributes. Any attempts to reuse a token should fail the test.
IT-2g	Validate authentication session is terminated after authentication is successful	User authenticates as normal. Attempting to navigate back to previous authentication pages or closing browser should terminate session and not allow user to go back
IT-2h	Automatically enters lockdown mode when an active voting session is detected and prevents unauthorized actions.	Simulate an active voting session and confirm that the device disables all non-voting functions, blocks exit attempts, prevents external navigation, and restricts OS-level features until the session ends.
IT-2i	Submission Phase Validation	(i) Verify that the token is not JSON (i.e., is opaque) (ii) Verify that the correct token can be used to access the data (iii) Verify that an incorrect token cannot be used to access the data
IT-2j	Retain authentication between modules, without exposing PII	Verify that opaque tokens are passed between internal systems and can be authenticated without exposing any user PII. Any part of the system that receives an invalid token simply halts, throwing an exception to be handled by the API.
IT-2k	Submission Phase Validation	(i) Verify that the key of the identity of the voter in the voter file is distinct from the voter identifier in the file of ballots cast. (ii) Attempt to run the identity cryptographic key on the ballot identifier and confirm that the key does not decrypt the ballot identifier (iii) Attempt to run the ballot cryptographic key on the voter identity identifier and confirm that the key does not decrypt the identity identifier
IT-2l	Ensure separation of concerns with regards to PII	Verify that modules are only exposed to information strictly necessary for their operation. Ensure use of opaque tokens to hide information from modules that do not have authorization.
IT-2m	Submission Phase Validation	(i) Verify that the key of the identity of the voter in the voter file is distinct from the voter identifier in the file of ballots cast. (ii) Attempt to run the identity cryptographic key on the ballot identifier and confirm that the key does not decrypt the ballot identifier (iii) Attempt to run the ballot cryptographic key on the voter identity identifier and confirm that the key does not decrypt the identity identifier

Phase 2 tests cover the following functional and non-functional requirements:

Requirement ID	Requirement Description
FR-1	Three-factor authentication process
FR-1.1	Verify digital signatures from government IDs
FR-1.2	Two-factor authentication (password + OTP)
FR-1.3	Biometric verification (fingerprint/facial)
FR-1.4	Integration with state voter databases
FR-2	Issue anonymous credentials (tokens)
FR-3	Terminate authentication sessions
FR-4	Activate device lockdown mode
FR-11	Opaque token exchange between components
FR-22	Token-based communication
NFR-2	Blind signature cryptography
NFR-3	Information hiding principle
NFR-7	Mathematically untraceable votes

Phase Three: Ballot Delivery and Four-Stage Confirmation

Goal: Validate jurisdiction-specific ballots, unbiased rendering, accessibility, and the deterministic FR-8 confirmation sequence.

Test Case ID	Test Case Objective	Test Case Description
IT-3a	Loads and displays the correct ballot style based on jurisdiction and voter registration data.	Authenticate a voter with a known jurisdiction profile and check that the ballot matches the assigned precinct, including contests, ordering, and jurisdiction-specific rules.
IT-3b	Delivers a consistent ballot-marking experience across all supported platforms.	Access the ballot from PWA, iOS, and Android devices and verify consistency in UI layout, navigation, candidate lists, and interaction behavior across channels.
IT-3c	The candidate list is displayed in a randomized or non-biased manner in accordance with election guidelines.	Open multiple ballot sessions and confirm that candidate ordering follows the jurisdiction's fairness rules (e.g., rotation, random ordering) and displays correct information without omissions.
IT-3d	Four-stage confirmation	When casting a vote, verify that the confirmation process takes all four stages: selection, review, confirmation, submit

Test Case ID	Test Case Objective	Test Case Description
IT-3e	Selection phase validation	(i) Verify that the app opens to the Selection Phase (ii) Verify selection phase enables users to select one and only one candidate for each open office or seat (iii) Verify that the Review button appears after at least on selection is made
IT-3f	Review Phase Validation	(i) Verify the app opens the Review page after clicking the Review button on the Selection page (ii) Verify that the Review page has a Change button that returns the user to the Selection page when clicked (iii) Verify that the user's votes are not changed from the Selection page to/from the Review page (iv) Verify that the user navigates to the Confirmation page when the Confirm button is clicked
IT-3g	Confirmation Phase Validation	(i) Verify that the Confirmation page opens with the Confirm button is clicked on the Review Page (ii) Verify that the user's vote has not changed when navigating from the Review Page to/from the Confirm page (iii) Verify that the Confirmation page has a large button with wide margins (iv) Verify that user is navigated to the Submission page when the Confirm button is clicked
IT-3h	Submission Phase Validation	(i) Verify the user navigated from the Confirmation page to the Submission page when the Confirm button is clicked on the Confirmation page (ii) Verify the user's vote selection is unchanged from the Confirmation page to the Submission page (iii) Confirm that a count-down clock is visible and counts down from 3 to 1 (iv) Verify that the app automatically cancels the user's submission if the count-down clock reaches zero (v) Verify that the user's vote is submitted if the Submit button is clicked before the clock reaches zero. (vi) Verify that the user can stop the clock by clicking the Back button (vii) Verify that the clock restarts at 3 when the user returns to the Submission page
IT-3i	All major accessibility features remain available during the voting experience.	Test screen reader navigation, list scanning, keyboard navigation, audio cues, zoom, and high-contrast mode throughout the entire voting flow to ensure uninterrupted accessibility support.
IT-3j	Voting interface complies with WCAG 2.1 AA criteria for perceivable, operable, and understandable content.	Review UI elements for proper labels, contrast ratios, focus order, error messaging, and alternative input methods, ensuring compliance with WCAG checkpoints.

Test Case ID	Test Case Objective	Test Case Description
IT-3k	Four-stage confirmation	(i) Attempt to controvert the submission workflow by pausing for a long time between actions, using the back button, attempt to skip forward using the forward button and verify that the confirmation workflow cannot be put off-track

Phase 3 tests cover the following functional and non-functional requirements:

Requirement ID	Requirement Description
FR-5	Deliver jurisdiction-specific ballots
FR-6	Multi-channel ballot access (PWA, iOS, Android)
FR-7	Display unbiased candidate lists
FR-8	Four-stage confirmation workflow
FR-8.1	Selection stage
FR-8.2	Review stage with "Change" option
FR-8.3	Explicit confirmation stage
FR-8.4	Final submission with countdown timer
FR-9	Comprehensive accessibility features
FR-9.1 - 9.6	Screen readers, high-contrast, keyboard nav, audio, scaling, multi-language
NFR-20	Deterministic sequence workflow

Phase Four: Cast Ballot, Encrypt, Ledger and Audit

Goal: Establish unlinkability and immutability across vote cast/encrypt/record/receipt.

Test Case ID	Test Case Objective	Test Case Description
IT-4a	Submission Phase Validation	(i) Verify that the key of the identity of the voter in the voter file is distinct from the voter identifier in the file of ballots cast. (ii) Attempt to run the identity cryptographic key on the ballot identifier and confirm that the key does not decrypt the ballot identifier (iii) Attempt to run the ballot cryptographic key on the voter identity identifier and confirm that the key does not decrypt the identity identifier

Test Case ID	Test Case Objective	Test Case Description
IT-4b	Four-stage confirmation	(i) Using a took like chrome://webrtc-internals/, confirm that all internet connections are using 256-bit AES encryption
IT-4c	Verify encrypted votes are recorded on a permission ledger with append only, tamper-evident properties.	Submit one or more encrypted ballots through the cast path; query the ledger for corresponding entries; attempt invalid modifications (e.g., replay/alteration via simulated node) and confirm rejection; verify block metadata (hash, timestamp, jurisdiction tag) and inclusion proof are available to downstream audit tools.
IT-4d	Verify that rotating signing/encryption keys does not break verification, receipt checks, or ledger validation.	Execute a rotation via the key-management service; cast votes before and after rotation; confirm receipts and ledger validation succeed for both generations; ensure old keys are retired but verifiable, and new keys are active and discoverable; check rotation events are logged.
IT-4e	Ensure voters receive a cryptographic receipt that proves inclusion without revealing vote content.	Cast a ballot and capture the issued receipt; use the receipt against the verification interference to confirm ledger inclusion; validate that no ballot choices or identity data are revealed during verification.
IT-4f	Validate the portal can verify a receipt and tally inclusion using ZK proofs without exposing the vote.	Present a valid receipt to the portal; confirm it returns a positive proof of inclusion; exercise negative cases (malformed/expired receipts) and verify denial with safe error messaging; confirm no identity or selection data is exposed in responses or logs.
IT-4g	Confirm all Phase-5 operations write tamper-evident, privacy-preserving audit entries.	Perform cast → ledger → rotation →tally →verification flows; inspect audit trail for each step (IDs, timestamps, nonces); attempt to alter or delete entries and confirm detection (hash mismatch/chain break); ensure logs contain no PII or ballot content.
IT-4h	Validate that an external verifier can reproduce results using public proofs and ledger state.	Export the required public artifacts (header, commitments, proofs); run a verifier (black-box harness) to recompute totals; confirm match with system tally; introduce a controlled discrepancy in a staging snapshot and verify the verifier flags it.
IT-4i	Confirm encryption remains end-to-end through storage and during tally prep.	Trace a ballot from client encryption through ledger persistence; validate ciphertext only at rest/in transit; ensure decryption occurs only within threshold-tally flow with required quorum.
IT-4j	Validate policy and mechanics of periodic rotation and distributed escrow.	Trigger a scheduled rotation; verify keys are escrowed with threshold access; confirm old keys verify historical artifacts; ensure monitoring/alerts fire for rotation milestones.

Test Case ID	Test Case Objective	Test Case Description
IT-4k	Demonstrate that no entity can infer how a specific voter voted.	Correlate Phase-5 artifacts (receipts, ledger entries, proofs, tally outputs) and confirm no linkage to identities; attempt linkage attacks (timing, metadata) and verify mitigation (batching/mixing/indistinguishability).
IT-4l	Ensure compatibility with leading screen readers across platforms (JAWS, VoiceOver, NVDA).	Run the ballot interface with each supported screen reader, verifying correct semantic structure, announced labels, navigable components, and accurate reading of candidate/contest information.
IT-4m	Validate correct system performance across network, connectivity, and device variability reflective of rural and urban deployment contexts.	Execute voting sessions under different bandwidth, latency, and device capability profiles; confirm the interface and data submission behave correctly under each scenario.
IT-4n	Four-stage confirmation	(i) Verify that the application makes a reassuring chime when a candidate is selected and different chimes for each of the 4-phase confirmation. (ii) When the vote is finally submitted, issue a "Your vote has been recorded" visual message and audio message (iii) Verify that the message is voiced in the language preference of the user
IT-4o	Four-stage confirmation	(i) Visually inspect buttons for sufficiently large size for finger-touches (ii) Visually inspect text to confirm font size of at least 14 points (iii) Confirm count-down timer appears on each screen and is voiced for users who set that option in their profile
IT-4p	Verify audit logs are immutable and cross-checked against ledger anchors.	Compare audit logs chains with on-chain anchors; simulate log truncation/alteration and confirm detection; verify retention controls are enforced by policy (pointer to retention system if separate).
IT-4q	Confirm the system stores audit logs in a format and location that supports mandatory 22-month retention.	Submit sample ballots and verify that logs (timestamps, selections, events) appear in the long-term storage location with correct retention policy metadata applied.
IT-4r	Validate that the system generates and verifies cryptographic ZKP proofs for election integrity without exposing voter identity.	Trigger a ballot submission through the verification pipeline and confirm that a valid ZKP is produced, validated, logged, and stored without revealing identifying voter information.

Phase 4 tests cover the following functional and non-functional requirements:

Requirement ID	Requirement Description
FR-10	Blind signature cryptography

Requirement ID	Requirement Description
FR-12	End-to-end encryption (256-bit AES)
FR-13	Blockchain immutable storage
FR-14	Automated cryptographic key rotation
FR-15	Generate cryptographic vote receipts
FR-16	Verification portal with zero-knowledge proofs
FR-19	Tamper-evident audit logs
FR-20	Independent third-party verification
NFR-1	End-to-end encryption (256-bit AES)
NFR-5	Quarterly key rotation
NFR-10	Ballot secrecy compliance
NFR-11	WCAG 2.1 Level AA compliance
NFR-12	Scenario-based design (rural/urban)
NFR-13	Explicit visual/audio feedback
NFR-14	Anti-misclick design
NFR-24	Tamper-evident audit logs
NFR-25	22-month log retention
NFR-26	Zero-knowledge proof verification

Phase Five: Vote Tally, Verification Portal, and Pollbook Sync

Goal: Validate threshold decryption, end-to-end verifiability, and duplicative-vote prevention across channels.

Test Case ID	Test Case Objective	Test Case Description
IT-5a	Ensure tallying requires M-of-N officials and cannot proceed with an incomplete quorum.	Close the election window; attempt tally with insufficient keys (expect block); perform tally with a valid quorum and verify aggregated results are produced; confirm partial decrypts/signatures are recorded for audit.
IT-5b	Verify post-submission status is synchronized to prevent duplicates across all channels.	After a successful submission, immediately attempt additional digital and in-person (simulated pollbook) votes; confirm rejection with correct reason; verify cross-channel sync events and conflict handling under brief network partition/recovery.
IT-5c	Ensure data consistency	Verify that votes are not duplicated or lost across nodes, and that new vote tallies are available within 5 minutes of submission.

Test Case ID	Test Case Objective	Test Case Description
IT-5d	Separation of concerns between microservices	Ensure that the authentication module does not take in more data than absolutely necessary for authenticating users. Verify that only opaque tokens are passed to the ballot and voting modules to confirm authentication.

Phase 5 tests cover the following functional and non-functional requirements:

Requirement ID	Requirement Description
FR-17	Vote tallying with threshold decryption
FR-18	Update voter status in database
FR-25	Real-time voter status synchronization
NFR-9	Separation of concerns (authentication vs. voting)

Validation Testing

End-to-end validation confirms the system meets functional and non-functional requirements without exposing user identities or vote content. Below are validation test skeletons mapped to requirements. Included are format and steps only; no specific test data nor expected results are described.

Validation Test ID	Validation Test Objective	Validation Test Description	Functional and Non-Functional Requirement ID
VT-1	**End-to-End Secure Voting Session:** Validate that a voter can complete a full voting session (authentication → ballot → confirmation → cast→receipt) securely and anonymously	Execute a complete user journey: perform three-factor authentication to obtain an anonymous credential; access the correct jurisdiction ballot; navigate the four-stage confirmation workflow (select → review→ confirm → submit); cast the vote so it is encrypted, recorded on the ledger, and a cryptographic receipt is issued. Throughout the flow, confirm that only opaque tokens are used beyond the authentication boundary and no personally identifiable information appears in downstream requests or logs.	FR-1, FR-1.1, FR-1.2, FR-1.3, FR-1.4, FR-2, FR-3, FR-5, FR-6, FR-8, FR-8.1–FR-8.4, FR-12, FR-13, FR-15, FR-22; NFR-1, NFR-3, NFR-7, NFR-9, NFR-10

Validation Test ID	Validation Test Objective	Validation Test Description	Functional and Non-Functional Requirement ID
VT-2	**Multi-Channel Ballot Delivery & Accessibility:** Verify that voters across devices and connectivity conditions receive the correct ballot and can use all accessibility features.	Using the same eligible voter profile, access the system from PWA, iOS, and Android clients under both normal and low-bandwidth network conditions. Confirm that each client delivers the correct jurisdiction-specific ballot and that screen reader navigation, high-contrast display, keyboard-only navigation (where applicable), audio guidance, text scaling up to 200%, and multi-language options remain available and usable throughout the full voting flow.	FR-5, FR-6, FR-9, FR-9.1–FR-9.6; NFR-11, NFR-12, NFR-13, NFR-14, NFR-15, NFR-16
VT-3	**Deterministic Four-Stage Confirmation Workflow:** Validate that the entire confirmation workflow is deterministic, resistant to user navigation tricks, and prevents accidental submission.	From ballot access through submission, exercise the four-stage sequence (selection, review, confirmation, submission) while introducing navigation variations: long pauses, use of browser back/forward, repeated attempts to skip stages, and repeated confirmation clicks. Confirm the sequence cannot be bypassed or reordered, and that the countdown-based submission stage always enforces the cancel window and multi-step confirmation behavior.	FR-8, FR-8.1, FR-8.2, FR-8.3, FR-8.4; NFR-20
VT-4	**Anonymity, Encryption, and Ledger Integrity:** Confirm that votes remain encrypted and unlinkable to voters while being immutably stored on the ledger.	Complete multiple voting sessions from different voters and jurisdictions. For each session, trace the flow from anonymous credential to encrypted ballot, blind-signing, and ledger recording. Inspect stored artifacts to verify only ciphertext and non-identifying metadata are persisted, then attempt correlation between authentication records and ledger entries using timing and metadata. Confirm that blind signatures, key separation, and ledger design prevent linking a specific voter to a specific encrypted ballot.	FR-10, FR-12, FR-13, FR-19; NFR-1, NFR-2, NFR-3, NFR-7, NFR-8, NFR-10, NFR-24
VT-5	**Receipt Verification and Zero-Knowledge Proofs:** Validate that voters can verify inclusion of their vote through the verification portal using cryptographic receipts and zero-knowledge proofs.	For multiple completed voting sessions, collect the issued cryptographic receipts and submit them to the verification portal. Confirm that valid receipts yield successful inclusion verification while malformed, expired, or forged receipts are rejected with safe, non-revealing error messages. Ensure that, at no point in the verification process, the portal exposes ballot content or identity information in responses or logs.	FR-15, FR-16, FR-20; NFR-2, NFR-3, NFR-24, NFR-26
VT-6	**Threshold Tally and Independent Re-Verification:** Ensure that final election tallies require threshold decryption by multiple officials and can be independently re-verified.	After an election window closes, attempt to run the tally with fewer than the required number of decryption key holders and confirm the operation is blocked. Then run the tally with a valid quorum and export the public proofs and ledger metadata. Use an independent verification tool or process to recompute aggregate results from the public artifacts and confirm they match the system's reported tallies.	FR-17, FR-20; NFR-5, NFR-24, NFR-26

Validation Test ID	Validation Test Objective	Validation Test Description	Functional and Non-Functional Requirement ID
VT-7	**Duplicate-Vote Prevention Across Channels:** Validate that a voter cannot successfully cast more than one vote across digital and in-person channels.	Have an eligible voter cast a digital ballot to completion. Immediately afterward, attempt to submit another digital ballot with the same identity, followed by a simulated in-person pollbook vote for the same voter. Introduce a short simulated network partition and recovery while these attempts occur. Confirm that all subsequent attempts are rejected and that voter status remains synchronized and consistent across systems after connectivity is restored.	FR-18, FR-25; NFR-9, NFR-22
VT-8	**System Availability, Performance, and Completion Time:** Validate that the system maintains required availability, scalability, and completion time envelopes during peak voting periods.	Apply a simulated peak-load traffic pattern that approximates millions of concurrent users during a compressed election time window. Observe system behavior for service availability, horizontal scaling of microservices, and typical end-to-end vote completion time for representative users. Focus on whether the system continues to accept and process votes within configured response time and completion time envelopes, without specifying exact numeric metrics in this test definition.	FR-21, FR-23; NFR-15, NFR-17, NFR-18, NFR-19, NFR-21, NFR-22, NFR-23
VT-9	**Audit Trail Completeness and Retention:** Confirm that all critical election events are captured in tamper-evident audit logs and retained for the required period.	Execute a representative subset of election activities (authentication, ballot access, vote casting, key rotation, tallying, verification checks). Inspect the audit logs to ensure each step is recorded with appropriate identifiers and timestamps, and that the logs are anchored or cross-checked against ledger metadata. Validate that audit entries are stored in the designated long-term location with retention controls appropriate to the legal retention period.	FR-19; NFR-24, NFR-25, NFR-27
VT-10	**Legal and Policy Compliance Check:** Validate that the deployed system's observable behavior aligns with applicable election regulations and standards.	Review the full end-to-end voting, tally, and verification process as exercised in earlier validation tests against Help America Vote Act (HAVA), Election Assistance Commission (EAC) Voting System Standards, and relevant state electronic voting rules. Confirm that ballot secrecy, auditability, accessibility, and voter eligibility behaviors observed in the system conform to these standards, and that no observed operational behavior contradicts required policies.	NFR-10, NFR-11, NFR-24, NFR-25, NFR-27