ELECTA Unified Development Log & Execution Plan

This document consolidates prior development logs and the updated version into a single authoritative record for ongoing project tracking.

# Section 1: Original Development Log

ELECTA - Development Log (v1.8) Document Version: 1.8 Date: August 7, 2025 Status: Finalized for Session --- Date: July 25, 2025 Sprint: Sprint 0 Summary: Development commenced. Git repositories created. --- Date: July 28, 2025 Sprint: Sprint 0 Summary: Pivoted to a Docker Compose local development strategy, unblocking core development. --- Date: July 30, 2025 Sprint: Sprint 1 Summary: Stabilized the local development environment after extensive troubleshooting of Docker, database, and migration issues. --- Date: August 5, 2025 Sprint: Sprints 1 & 2 (Backend) Summary: Completed all backend development for Sprints 1 and 2, including OTP registration, Admin creation, and the Charter module API/DB schema. Tasks Completed: BE-3.1, BE-3.2, RATE-1, AUDIT-1, DB-2, BE-8, BE-9. --- Date: August 6, 2025 Sprint: Sprints 1, 2, & 3 (Android & Backend Hardening) Summary: Completed all Android features for Sprints 1-3. Successfully executed Sprint 3 hardening tests (PERF-1, CHAOS-1) on the backend. Tasks Completed: FE-A-2, FE-A-3, FE-A-4, FE-A-5, PERF-1, CHAOS-1. --- Date: August 7, 2025 (New Entry) Sprint: Sprints 1 & 3 (Web Frontend) Summary: Completed all planned hands-on development for the React Admin Panel. After containerizing the application to solve environmental issues, successfully implemented the Admin Login page, a placeholder Dashboard with routing, and the initial UI for managing Charter Ideologies and Party Compliance records. All code has been committed. Tasks Completed: FE-W-1, FE-W-2, FE-W-3. Status Change: All planned hands-on development for the project's foundational features across Backend, Android, and Web is now complete. The project is in a stable state.

# Section 2: Updated Development Log (2025-08-20)

Title: Auth Contract Unification & API Stability Pass  
Date: August 21, 2025 (Asia/Dhaka)  
Author: Saif

Summary

Aligned all clients (Web Admin, Android) and the FastAPI backend to a single auth contract:

Backend returns { "access\_token": "<JWT>", "token\_type": "bearer" }

Clients send Authorization: Bearer <JWT>

Web stores token as localStorage["admin\_token"]

Android stores token via EncryptedSharedPreferences (TokenManager)

Stabilized API startup, docs, CORS, and rate limiting.

Scope of Changes

Backend (FastAPI)

app\api\v1\endpoints\admin.py: fixed JWT creation with create\_access\_token(subject=..., additional\_claims=...); ensured token shape.

app\api\v1\endpoints\auth.py: standardized login/OTP flow; added shared SlowAPI limiter; enforced request: Request on limited routes.

app\api\v1\api.py: aggregator includes admin, auth, ekyc, (optionally ideologies placeholder).

app\main.py: simplified startup; added CORS; wired shared limiter & 429 handler.

app\core\security.py: unified helpers (hash/verify, create/decode token), OAuth2 password bearer for docs.

app\schemas\token.py: Pydantic v2 style (ConfigDict(from\_attributes=True)).

app\schemas\\_\_init\_\_.py: explicit exports (\_\_all\_\_).

app\deps.py: (new) optional helper for resolving current admin (not strictly required today).

app\limits.py: (new) single shared SlowAPI Limiter.

Web (React Admin)

src\lib\axios.ts: single axios instance, interceptor reads admin\_token and sets Authorization header; TS-safe headers handling.

Migrated any axios.get("http://localhost:8000/api/v1/...") to api.get("/...").

Login page stores admin\_token; removed legacy token key.

.env.development: REACT\_APP\_API\_BASE\_URL=http://localhost:8000.

Android

app\build.gradle.kts: Debug BASE\_URL="http://10.0.2.2:8000/"; Release points to prod.

RetrofitClient.kt: adds Authorization: Bearer <token> via interceptor.

TokenManager: secure storage; initialized in Application class.

Debug network security config allows cleartext to 10.0.2.2.

Rationale

One consistent auth shape reduces cross-client bugs.

Interceptors centralize header logic.

CORS configured from .env to match front-end origin(s).

SlowAPI provides basic rate-limit protection on brute-forceable endpoints.

How to Test (performed)

Backend: uvicorn app.main:app --reload → /healthz and /openapi.json load instantly; /docs loads.

Admin login (form): POST /api/v1/admin/login returns { access\_token, token\_type }.

EKYC initiate: POST /api/v1/ekyc/initiate with Authorization: Bearer ... succeeds (mock flow).

React Admin: Login stores only admin\_token; all API calls include Bearer header and hit /api/v1/....

Android (emulator): Base URL 10.0.2.2:8000; login/OTP flows call backend with Bearer header.

Migration Notes

Replace any storage/usage of token with admin\_token (web).

Ensure Android BuildConfig.BASE\_URL is correct for debug/prod.

If docs hang on some networks, we can switch to locally served Swagger assets.

Known Follow-ups (next tasks)

Async SQLAlchemy for auth.py/others (async engine + session; async def routes and queries).

Validation: add mobile format validation (BD numbers), optional email checks where applicable.

Ideologies: swap placeholder with DB-backed CRUD.

Security: refresh tokens / rotation; scope claims for admin vs citizen; stricter CORS per env.

Observability: request logging, structured logs & error IDs.

Risk/Impact

Low risk; behavior aligned to Bearer tokens. Ensure front-ends are rebuilt after .env changes.

References

Files changed: see Git commits in feat/auth-contract-unification branch.

ELECTA - Development Log (v1.8) Document Version: 1.8 Date: August 7, 2025 Status: Finalized for Session --- Date: July 25, 2025 Sprint: Sprint 0 Summary: Development commenced. Git repositories created. --- Date: July 28, 2025 Sprint: Sprint 0 Summary: Pivoted to a Docker Compose local development strategy, unblocking core development. --- Date: July 30, 2025 Sprint: Sprint 1 Summary: Stabilized the local development environment after extensive troubleshooting of Docker, database, and migration issues. --- Date: August 5, 2025 Sprint: Sprints 1 & 2 (Backend) Summary: Completed all backend development for Sprints 1 and 2, including OTP registration, Admin creation, and the Charter module API/DB schema. Tasks Completed: BE-3.1, BE-3.2, RATE-1, AUDIT-1, DB-2, BE-8, BE-9. --- Date: August 6, 2025 Sprint: Sprints 1, 2, & 3 (Android & Backend Hardening) Summary: Completed all Android features for Sprints 1-3. Successfully executed Sprint 3 hardening tests (PERF-1, CHAOS-1) on the backend. Tasks Completed: FE-A-2, FE-A-3, FE-A-4, FE-A-5, PERF-1, CHAOS-1. --- Date: August 7, 2025 (New Entry) Sprint: Sprints 1 & 3 (Web Frontend) Summary: Completed all planned hands-on development for the React Admin Panel. After containerizing the application to solve environmental issues, successfully implemented the Admin Login page, a placeholder Dashboard with routing, and the initial UI for managing Charter Ideologies and Party Compliance records. All code has been committed. Tasks Completed: FE-W-1, FE-W-2, FE-W-3. Status Change: All planned hands-on development for the project's foundational features across Backend, Android, and Web is now complete. The project is in a stable state.

August 20, 2025 (Asia/Dhaka) — Project Audit & Completion Plan Snapshot

Summary of findings:

• Tech stack confirmed: React 18 (CRA) admin panel, FastAPI backend (with SQLAlchemy/Alembic), Kotlin Android app.

• Legacy/unused code present in backend: Express.js (Node) and Maven/Spring scaffold. Recommend deprecating and removing.

• Endpoint alignment issues: Admin Panel calls /api/v1/ideologies (GET/POST) which are not implemented in backend.

• Inconsistent auth on the web: both admin\_token and token are used; axios header initialization inconsistent.

• Android debug BASE\_URL points to 192.168.0.101:8080 while backend defaults to 8000; release points to api.prod.electa.com.

• Alembic migrations folder has no versions; database bootstrap likely incomplete.

Backend FastAPI route inventory (prefix: /api/v1):

• POST /admin/candidates (admin.py)

• POST /admin/login (admin.py)

• POST /login (auth.py)

• POST /register/request-otp (auth.py)

• POST /register/verify-otp (auth.py)

• GET /charter/versions (charter.py)

• POST /charter/clauses (charter.py)

• POST /charter/versions (charter.py)

• POST /compliance/evidence (compliance.py)

• POST /compliance/records (compliance.py)

• GET /ekyc/callback (ekyc.py)

• POST /ekyc/initiate (ekyc.py)

Admin Panel outbound axios calls discovered:

• GET http://localhost:8000/api/v1/ideologies [components/IdeologyManager.tsx]

• POST http://localhost:8000/api/v1/ideologies [components/IdeologyManager.tsx]

• POST http://localhost:8000/api/v1/admin/login [components/LoginPage.tsx]

• GET http://localhost:8000/api/v1/admin/ideologies [pages/IdeologyManager.tsx]

• POST http://localhost:8000/api/v1/admin/ideologies [pages/IdeologyManager.tsx]

• GET /api/users [services/dataService.js]

Android BASE\_URL configuration found:

• \"http://192.168.0.101:8080/\"

• \"https://api.prod.electa.com/\"

Immediate next steps (1–2 days of focused work):

• Remove Node/Maven remnants from backend; keep FastAPI-only. Add Dockerfile service and docker-compose service for API.

• Implement Ideology endpoints (+ SQLAlchemy model CRUD) to match Admin Panel: GET /api/v1/ideologies, POST /api/v1/ideologies.

• Create initial Alembic migration from models; wire startup migration runner.

• Normalize auth on web: use admin\_token consistently; set axios default Authorization: Bearer <token> via interceptor.

• Align Android debug BASE\_URL to FastAPI: http://10.0.2.2:8000/ (emulator) or http://<LAN-IP>:8000/; keep release pointing to production.

• Add CORS origins for web and Android; verify with .env CORS\_ORIGINS.

# Section 3: Unified Execution Plan

This section consolidates the execution strategy, pending task list, and implementation plan.  
  
Key Phases:  
1. Backend API stabilization (auth, OTP, citizen registration, token lifecycle).  
2. Frontend (Android) integration with backend API (SessionManager, TokenManager).  
3. Async SQLAlchemy migration & input validation (mobile/email formats).  
4. Testing: unit tests + integration tests for OTP, auth, and user flows.  
5. Deployment pipeline setup (CI/CD with Docker, Uvicorn/Gunicorn, Nginx).  
6. Benchmark review against ELECTA\_Project\_Benchmark.docx.  
7. Final QA and stakeholder demo.