ELECTA Project Benchmark (Consolidated)

This document merges narrative detail from ELECTA\_18-Project\_Benchmark with the structured, measurable benchmarks from ELECTA\_Project\_Benchmark. It serves as the master reference for tracking and reporting.

# Benchmark Sections

# Full Narrative Version (ELECTA\_18)

ELECTA Project – Comprehensive Benchmark of Achievements

1. Foundation & Infrastructure

✅ Repo & Git Strategy: Multi-service repo structure created (backend, web, android).

✅ Local Dev Environment: Docker Compose for PostgreSQL, Backend FastAPI running with migrations, Android app builds and runs.

⚠️ Pending: Automated CI/CD pipeline (GitHub Actions or GitLab CI).

Benchmark: A fully bootstrapped developer can clone → run docker-compose → run migrations → launch backend + frontend in under 10 minutes.

2. Backend (FastAPI)

✅ Core App Setup: FastAPI app with routers, rate limiting, health checks.

✅ Security: JWT auth, password hashing, rate limiting on login/OTP.

✅ User Management: OTP-based citizen registration, login with mobile + password.

⚠️ Pending: Ideologies CRUD API, Async SQLAlchemy refactor, Alembic migrations, Refresh token support.

Benchmark: Auth endpoints stable, at least one business feature (Ideologies) live, migrations reproducible.

3. Database

✅ Citizen schema implemented with password hashing.

⚠️ Pending: Alembic baseline + migrations, schema normalization, seed data.

Benchmark: `alembic upgrade head` builds DB schema, seed script inserts test data.

4. Mobile App (Android)

✅ Auth Flow: Registration (OTP + password), login with JWT storage.

✅ Networking: Retrofit with TokenInterceptor.

⚠️ Pending: Align BASE\_URLs with backend, implement Ideologies list + submission UI, improve error handling.

Benchmark: End-to-end register/login + fetch/post Ideologies from Android emulator against backend.

5. Web Admin (React)

✅ Admin login flow (with token), placeholder dashboard.

⚠️ Pending: Wire Ideologies API, implement CRUD panels.

Benchmark: Admin can login, view, and manage Ideologies via CRUD panels.

6. Observability & Security

✅ Basic health checks + rate limiting.

⚠️ Pending: Request/response logging, error correlation, refresh tokens, role-based scopes.

Benchmark: Logs with trace IDs, structured error JSON, at least two roles enforced.

7. DevOps & Deployment

✅ Local Dev: Docker Compose setup.

⚠️ Pending: Alembic auto-run, unified `.env`, production Dockerfile, staging deployment.

Benchmark: `docker-compose up` builds migrations + runs backend; deployable to staging.

8. Documentation

✅ Dev logs maintained.

⚠️ Pending: API docs cleanup, service-specific READMEs, architecture diagram.

Benchmark: New dev runs full stack locally in <30min, OpenAPI matches live endpoints.

🎯 Final MVP Benchmark

Backend: Auth + Ideologies API live with migrations.

Mobile: Android app register/login + Ideologies API integration.

Web: Admin login + CRUD Ideologies.

Database: Alembic migrations reproducible.

Infra: Docker-based local dev fully reproducible.

# Full Structured Benchmarks (ELECTA\_Project\_Benchmark)

ELECTA Project Benchmark

This benchmark defines the measurable achievements and milestones for the ELECTA project. It serves as a reference point to evaluate progress and readiness for release.

1. Core API

- ✅ Auth routes implemented: register (OTP + password), login, JWT issuance.

- ✅ Rate limiting per route using SlowAPI.

- ✅ Health/Ready endpoints available for monitoring.

- 🔄 Async SQLAlchemy migration (pending).

2. Security & Validation

- ✅ Passwords hashed with industry-standard algorithm.

- ✅ JWT tokens signed with environment-based secret.

- 🔄 Email and mobile number format validation (pending).

- 🔄 Redis OTP store hardened (expiry, replay protection).

3. Frontend Integration

- ✅ Android client integrated with backend session handling.

- ✅ TokenManager + SessionManager implemented.

- 🔄 UI/UX OTP entry and validation polish (pending).

4. Infrastructure & DevOps

- ✅ Project structured with FastAPI routers and modular services.

- ✅ CORS configured for local frontend integration.

- 🔄 CI/CD pipeline (Docker + GitHub Actions/GitLab CI) pending.

- 🔄 Production-ready server stack (Uvicorn + Gunicorn + Nginx).

5. Testing & QA

- ✅ Basic endpoint smoke tests working (/healthz, /readyz, /openapi.json).

- 🔄 Unit tests for auth flows, OTP service, and SMS provider.

- 🔄 Integration tests (frontend ↔ backend).

6. Final Milestone

The project achieves benchmark completion when:  
- All async DB migrations and validations are in place.  
- CI/CD pipeline reliably builds, tests, and deploys.  
- OTP and login flows pass end-to-end tests.  
- Frontend and backend fully integrated.  
- Stakeholder demo and QA sign-off completed.