QuantivaHQ — Mobile Handoff Documentation

Last updated: 2026-01-05

Table of Contents

---

1) System Overview

QuantivaHQ is a **multi-service trading intelligence platform** composed of:

- **Frontend (Next.js 14 – App Router)**
- User-facing web experience: onboarding, KYC, dashboard, AI insights, portfolio, community, and settings.
- **NestJS Backend (`quantiva_backend/q_nest`)**

- Core API, orchestration, cronjobs, Prisma/PostgreSQL data layer, exchange connectivity, auth, KYC, strategies, and task scheduling.

- **Python Backend (`quantiva_backend/q_python`)**

- FastAPI microservice housing ML/AI engines (sentiment, technical, fundamental, liquidity, event-risk, fusion, confidence), FinBERT sentiment, Finnhub stock batching, and KYC CV placeholders.

- **Database**

- PostgreSQL accessed via Prisma (NestJS) with expanded schemas for strategies, signals, KYC, trending assets, news, sessions, and more.

- **External Providers**

- Binance Testnet, Alpaca, LunarCrush, StockNewsAPI, Finnhub, CoinGecko, SendGrid (email), Interactive Brokers assets, and optional LLM for explanations.

---

2) Repository Layout

```
PROJECT_DOCUMENTATION.md   # This file
quantiva_backend/
├─ Docs (architecture, flows, KYC, signup, deletion, trading)
├─ Scripts & tests
├─ q_nest/                 # NestJS backend
└─ q_python/               # Python FastAPI + ML engines

QuantivaHQ-frontend/       # Next.js 14 frontend
```

Key Files & Directories

- **PROJECT_DOCUMENTATION.md** — Root onboarding reference
- **quantiva_backend/q_nest/** — NestJS service (Prisma, modules, cronjobs, controllers)
- **quantiva_backend/q_python/** — FastAPI service (ML engines, batching, KYC CV placeholders)
- **QuantivaHQ-frontend/** — Next.js App Router frontend

---

3) Backend — NestJS (`quantiva_backend/q_nest`)

Architecture

- Layered design:

- **Controllers → Services (business logic, caching, orchestration) → Integrations (raw HTTP/SDK, signing, retries)**
- Prisma ORM for PostgreSQL
- Task scheduling via `@nestjs/schedule`
- Storage abstraction (`src/storage`) with filesystem implementation (S3-ready)

See `SERVICE_ARCHITECTURE_EXPLANATION.md` for a Binance Testnet reference implementation.

Key Domains & Flows

Auth & Sessions

- Signup/login with **mandatory email-based 2FA** (SendGrid)
- JWT access + refresh tokens
- Sessions persisted in `user_sessions`
- Email verification field exists but is **not enforced** (`email_verified` remains false unless manually updated)

Signup Flow (Implemented)

1. `POST /auth/register`
2. Creates user, hashes password, initializes 2FA secret
3. `POST /auth/login`
4. Validates credentials, sends 6-digit 2FA code via email
5. `POST /auth/verify-2fa`
6. Verifies code, issues access/refresh tokens, creates session
7. `GET /auth/profile` + `POST /users/profile`
8. Collects personal and professional data

See **SIGNUP_FLOW_COMPLETE.md** for field-level mappings.

Account Deletion (Production-Ready)

- **Two-step destructive flow**:
- `POST /auth/request-delete-account-code`
- `DELETE /auth/delete-account`

- **Four-layer security**:

- JWT
- Password confirmation
- 2FA deletion code

- Business checks (active orders, positions, subscriptions)

- Deletes **26 entity types across 10 phases** in a single DB transaction

- Cleans stored files (profile images, KYC docs)
- Extended DB timeout (60s)

See **ACCOUNT_DELETION_COMPLETE_FLOW.md**.

KYC Module

- Upload documents and selfie, submit for review, poll status
- Admin review: approve / reject / resubmit
- Decision engine with configurable thresholds
- Guard: `kyc-verified.guard.ts`

Stored fields include: - Liveness result & confidence - Face match score - Document authenticity score - MRZ data, document metadata

See **KYC_IMPLEMENTATION.md**.

Binance Testnet (Paper Trading)

- Controller / Service / Integration separation
- HMAC-SHA256 request signing
- Retries (3x), rate-limit handling
- Short-term caching (3–5s TTL)
- Parallel symbol fetching with fallback defaults

See: - `BINANCE_TESTNET_FLOW_EXPLANATION.md` - `SERVICE_ARCHITECTURE_EXPLANATION.md`

Strategies & Signals — Crypto

- Cron: every **10 minutes**
- Fetches 50 trending assets (LunarCrush)
- Runs Python ML engines
- Stores results in `strategy_signals` & `signal_details`
- Optional LLM explanations for top signals

Engine weights: - Sentiment: 35% - Technical: 25% - Fundamental: 15% - Event Risk: 15% - Liquidity: 10%

Thresholds: - BUY > +0.3 - SELL < -0.3 - High event risk → HOLD override

See **CRYPTO_TRADING_ENGINE_FLOW.md**.

Strategies & Signals — Stocks

- Dedicated cron every **10 minutes**
- 50 trending stocks via Finnhub batching
- Batch processing (3 assets at a time)
- FinBERT sentiment + 7 engines
- Alpaca for OHLCV and quotes
- LLM explanations for top 10 signals

Thresholds: - BUY > +0.5 - SELL < -0.5

See **STOCK_IMPLEMENTATION_SUMMARY.md**.

Configuration & Environment

- PostgreSQL connection
- API keys (Binance, Alpaca, Finnhub, LunarCrush, CoinGecko, SendGrid)
- Python API URL
- KYC thresholds
- Storage root path

Run:

```
cd quantiva_backend/q_nest
npm install
npm run start:dev
```

Prisma migrations:

```
npm run prisma:migrate:dev
```

Testing

- `test-stocks-workflow.js` — Node-based E2E stock tests
- `test-python-backend.py` — Python API health & engine tests
- `test-all.sh` — Aggregated backend tests

---

4) Backend — Python (`quantiva_backend/q_python`)

Purpose

FastAPI service powering ML/AI scoring, sentiment analysis, stock batching, and KYC CV placeholders.

Engines

- Sentiment (FinBERT, social metrics, EMA smoothing)
- Technical (multi-timeframe indicators)
- Fundamental (market & on-chain metrics)
- Event Risk (30-day events)
- Liquidity (order book depth, slippage)
- Fusion (weighted final score)
- Confidence (position sizing, risk caps)

Stocks Support

- Finnhub batch APIs (fundamentals, earnings, sentiment, trending)
- Reduces ~150 calls to 3
- Asset-type aware thresholds (`asset_type='stock'`)

KYC CV Placeholders

Endpoints (stubbed): - `/api/v1/kyc/ocr` - `/liveness` - `/face-match` - `/document-authenticity`

Planned: EasyOCR, MediaPipe, DeepFace, OpenCV

Run & Environment

```
cd quantiva_backend/q_python
pip install -r requirements/base.txt
python run.py
```

Windows scripts available (`run.bat`, `run.ps1`).

---

5) Frontend — Next.js (`QuantivaHQ-frontend`)

Tech Stack & Structure

- Next.js 14 (App Router)
- TypeScript + Tailwind v4
- Zustand (state), Zod (validation)
- 40-screen scaffold across auth & dashboard

Key folders: - `components/layout` — App shell - `config/navigation.ts` — Sidebar routes - `state/` — Zustand stores - `lib/api` — API placeholders

---

Recent Fixes / Notes

- IBKR logo integration
- Navigation and sidebar fixes
- Global CSS appearance rule

See: - `IBKR_Integration_Summary.md` - `Navigation_Fix_Summary.md`

Running

```
cd QuantivaHQ-frontend
npm install
npm run dev
```

Open: http://localhost:3000

---

6) Data Model Highlights (Prisma / DB)

- `users` — auth, profile, KYC status, 2FA secret
- `two_factor_codes` — 6-digit codes with expiry & usage
- `user_sessions` — refresh tokens & devices
- `kyc_*` tables — documents, face matches, scores
- `trending_assets`, `trending_news`
- `strategies`, `strategy_signals`, `signal_details`, `signal_explanations`
- `orders`, `order_executions`, `portfolios`

---

7) External Integrations

- **Crypto**: Binance, Bybit, LunarCrush, CoinGecko

- **Stocks**: Finnhub, Alpaca, StockNewsAPI
- **AI/ML**: FinBERT, optional LLM
- **Comms**: SendGrid
- **Storage**: Local FS (S3 planned)

---

8) Operational Notes for Mobile Team

- Mandatory email-based 2FA
- KYC is multi-step with polling
- Paper trading via Binance Testnet
- Signals auto-refresh every 10 minutes
- Account deletion is a strict two-step flow

---

9) Getting Started (All Services)

1. Install dependencies
2. Run Python backend
3. Run NestJS backend
4. Run frontend
5. Verify health endpoints

---

10) Key References

- SIGNUP_FLOW_COMPLETE.md
- ACCOUNT_DELETION_COMPLETE_FLOW.md
- KYC_IMPLEMENTATION.md
- CRYPTO_TRADING_ENGINE_FLOW.md
- STOCK_IMPLEMENTATION_SUMMARY.md
- SERVICE_ARCHITECTURE_EXPLANATION.md

---

11) Open Gaps / TODOs

- Email verification not implemented
- KYC ML models pending
- Role-based access for KYC review
- Rate limiting for KYC uploads
- LLM provider finalization

This document is the primary onboarding reference for the mobile development team.