

AIAIgoTradeHits.com

Complete Architecture Document

Generated: January 12, 2026
Version 4.0

Table of Contents

1. Executive Summary
2. System Architecture Overview
3. Folder & File Structure
4. Complete Site Map
5. Frontend Components Analysis
6. Backend API Structure
7. Cloud Functions Inventory
8. Database Schema
9. Redundancy Analysis
10. NLP Development Roadmap
11. Environment Configuration
12. Refactoring Recommendations

1. Executive Summary

AIAlgoTradeHits.com is a comprehensive AI-powered fintech trading platform that provides:

Core Capabilities:

- Multi-timeframe market analysis (Daily, Hourly, 5-Minute)
- AI/ML predictions with 66.2% UP accuracy using Nested Multi-Timeframe Model
- Natural Language Processing (NLP) for intelligent search
- Real-time data from 5 APIs: TwelveData, Kraken, FRED, CoinMarketCap, Finnhub
- 6 asset classes: Stocks, Crypto, Forex, ETFs, Indices, Commodities
- 24 technical indicators per masterquery.md specifications

Technology Stack:

- Frontend: React 18 + Vite + TradingView Lightweight Charts
- Backend: Python Flask API on Google Cloud Run
- Database: Google BigQuery (1.2B+ records across 113 tables)
- ML: BigQuery ML (XGBoost) + Vertex AI
- Infrastructure: GCP Cloud Functions, Cloud Scheduler, Cloud Storage

Metric	Value	Notes
Total Data Records	1.2B+	Across all timeframes
BigQuery Tables	113	Data warehouse
Cloud Functions	33	Data pipeline
Cloud Schedulers	54	Automation
React Components	42	Frontend UI
API Endpoints	80+	REST API
ML Model Accuracy	68.4%	Nested Multi-TF
UP Prediction	66.2%	6.2x improvement

2. System Architecture Overview

High-Level Architecture:

Data Layer (BigQuery)

- crypto_trading_data dataset (main)
- ml_models dataset (ML features/models)
- 113 tables with partitioning

API Layer (Cloud Run)

- trading-api service
- Flask-based REST API
- 80+ endpoints

Data Collection (Cloud Functions)

- bulletproof-fetcher (primary)
- 33 specialized functions
- Multi-source integration

Frontend (Cloud Run)

- React 18 SPA
- 42 components
- TradingView charts

ML Pipeline (BigQuery ML + Vertex AI)

- Nested Multi-Timeframe Model
- Walk-Forward Validation
- Feature engineering

2.1 Data Flow Architecture

Data Collection Flow:

- Cloud Scheduler triggers Cloud Functions (hourly/daily)
- bulletproof-fetcher calls TwelveData API (800 calls/min)
- Parallel fetching from Kraken, FRED, CMC, Finnhub
- Data processed with 24 indicators per masterquery.md
- Uploaded to BigQuery tables (_clean suffix)
- ML features computed and stored
- Predictions generated and cached

User Request Flow:

- User interacts with React frontend
- API calls to trading-api (Cloud Run)
- API queries BigQuery (read-only)
- Results transformed and returned as JSON
- Frontend renders data with TradingView charts

3. Folder & File Structure

3.1 Root Directory Structure

```
C:\1AITrading\Trading\  
    stock-price-app/ # React Frontend Application  
    cloud_function_api/ # Main REST API (Flask)  
    cloud_function_bulletproof/ # Primary Data Fetcher  
    cloud_function_*/ # 33 Cloud Functions  
    cloud_run_trading_agent/ # AI Trading Agent  
    shared_ai_modules/ # Reusable AI Components  
    agentic_trading_system/ # Multi-Agent Framework  
    models/ # ML Model Artifacts  
    utils/ # Utility Scripts  
    documents/ # Generated PDFs  
    masterquery.md # PRIMARY SPECIFICATION  
    CLAUDE.md # AI Assistant Instructions  
    *.py # 349 Python Scripts
```

3.2 Frontend Structure (stock-price-app/)

```
stock-price-app/  
    src/  
        components/ # 42 React Components  
            TradingDashboard.jsx # Main dashboard  
            SmartDashboard.jsx # Search-enabled dashboard  
            Navigation.jsx # Side menu + top bar  
            NestedSignals.jsx # ML signals display  
            MultiTimeframeTrader.jsx # Multi-TF analysis  
            WalkForwardValidation.jsx # ML validation  
            ... (39 more)  
        services/  
            api.js # API client service  
            marketData.js # Market data utilities  
            aiService.js # AI integration  
            monitoringService.js  
        utils/  
            technicalAnalysis.js  
            App.jsx # Main app + routing  
            App.css # Global styles  
            main.jsx # Entry point  
            index.css # Base CSS  
            theme.js # Theme configuration  
        public/ # Static assets  
        dist/ # Build output  
        package.json # Dependencies  
        vite.config.js # Vite configuration  
        Dockerfile # Container build
```

3.3 Backend API Structure (cloud_function_api/)

```
cloud_function_api/  
    main.py # Flask app + 80+ endpoints  
    nlp_query_engine.py # NLP search engine
```

```
■■■ walk_forward_endpoints.py # ML validation API  
■■■ ai_endpoints.py # AI/ML endpoints  
■■■ ai_trading_service.py # Trading signal service  
■■■ trading_alerts.py # Alert system  
■■■ ml_ensemble_endpoints.py # Ensemble ML  
■■■ requirements.txt # Python dependencies  
■■■ Dockerfile # Container build  
■■■ deploy_api.py # Deployment script
```

4. Complete Site Map

4.1 Public Routes (All Users)

Route	Component	Description	Status
/dashboard	SmartDashboard	Main dashboard with NLP search	ACTIVE
/trade-analysis	MultiTimeframeTrader	Daily/Hourly/5-Min analysis	ACTIVE
/opportunity-report	OpportunityReport	Daily ranked opportunities	ACTIVE
/weekly	WeeklyDashboard	Weekly analysis all assets	ACTIVE
/weekly/stocks	WeeklyAnalysis	Weekly stocks analysis	ACTIVE
/weekly/cryptos	WeeklyAnalysis	Weekly crypto analysis	ACTIVE
/weekly/etfs	WeeklyAnalysis	Weekly ETF analysis	ACTIVE
/weekly/forex	WeeklyAnalysis	Weekly forex analysis	ACTIVE
/weekly/indices	WeeklyAnalysis	Weekly indices analysis	ACTIVE
/weekly/commodities	WeeklyAnalysis	Weekly commodities	ACTIVE
/ai-signals/predictions	AIPredictions	ML price predictions	ACTIVE
/ai-signals/patterns	AIPatternRecognition	Chart pattern detection	ACTIVE
/ai-signals/sentiment	ComingSoon	Sentiment analysis	PLANNED
/ai-signals/signals	AITradeSignals	Buy/sell signals	ACTIVE
/ai-signals/anomalies	ComingSoon	Anomaly detection	PLANNED
/portfolio	PortfolioTracker	Portfolio management	ACTIVE
/alerts	PriceAlerts	Price alert system	ACTIVE
/strategies	StrategyDashboard	Trading strategies	ACTIVE
/market-movers	MarketMovers	Top gainers/losers	ACTIVE
/fundamentals	FundamentalsView	Company fundamentals	ACTIVE
/etf-analytics	ETFAnalytics	ETF analysis	ACTIVE
/ai-training	AITrainingDocs	ML training docs	ACTIVE
/documents	DocumentsLibrary	System documentation	ACTIVE
/settings	Settings (inline)	Account settings	ACTIVE

4.2 Admin Routes (Admin Only)

Route	Component	Description	Status
/admin	AdminPanelEnhanced	User management	ACTIVE
/table-inventory	TableInventory	BigQuery table browser	ACTIVE
/data-warehouse-status	DataWarehouseStatus	Data collection monitor	ACTIVE
/database-summary	DatabaseSummary	XLSX export	ACTIVE

/data-export	DataExportDownload	97-field CSV export	ACTIVE
/ml-test-data	MLTestDataDownload	ML test datasets	ACTIVE
/walk-forward	WalkForwardValidation	ML validation system	ACTIVE

5. Frontend Components Analysis

5.1 Component Inventory (42 Components)

Component	Lines	Purpose	Category
TradingViewChart.jsx	1798	Full TradingView integration	CORE
SmartDashboard.jsx	1255	NLP-enabled main dashboard	CORE
TableInventory.jsx	1011	BigQuery table browser	ADMIN
OrganizationChart.jsx	969	Org structure display	UTILITY
WeeklyDashboard.jsx	966	Weekly analysis hub	CORE
MultiTimeframeTrader.jsx	944	Multi-TF trading analysis	CORE
TradingDashboard.jsx	888	Main trading dashboard	CORE
DataDownloadWizard.jsx	885	Data export wizard	ADMIN
SmartearchBar.jsx	827	NLP search component	CORE
EnhancedDashboard.jsx	821	Enhanced dashboard	REDUNDANT
AIPatternRecognition.jsx	761	Pattern detection UI	AI
OpportunityReport.jsx	783	Daily opportunities	CORE
Navigation.jsx	740	Side menu + top bar	CORE
AITrainingDocs.jsx	643	ML training documents	AI
NLPSearch.jsx	628	NLP search page	REDUNDANT
WeeklyReconciliation.jsx	620	Data reconciliation	REDUNDANT
MLTestDataDownload.jsx	600	ML test data export	ADMIN
WeeklyAnalysis.jsx	584	Weekly analysis view	CORE
DataExportDownload.jsx	576	97-field export	ADMIN
AIAlgoTradeHits.jsx	564	Landing page	REDUNDANT
AIAlgoTradeHitsReal.jsx	542	Alternate landing	REDUNDANT

Note: Components marked REDUNDANT should be reviewed for removal.

6. Backend API Structure

6.1 API Endpoint Categories

Market Data Endpoints (30+):

- /api/twelveData/{asset_type}/{timeframe} - Generic TwelveData
- /api/stocks/history, /api/stocks/hourly, /api/stocks/5min
- /api/crypto/daily, /api/crypto/hourly, /api/crypto/5min
- /api/forex/history, /api/forex/hourly
- /api/etfs/history, /api/etfs/hourly
- /api/indices/history, /api/indices/hourly
- /api/commodities/history, /api/commodities/hourly

Analysis Endpoints (15+):

- /api/analysis/stocks/weekly - Weekly stock analysis
- /api/analysis/cryptos/weekly - Weekly crypto analysis
- /api/analysis/nlp-search - NLP natural language search
- /api/analysis/active-list - Active trading list

ML/AI Endpoints (10+):

- /api/nested/signals - Nested multi-TF signals
- /api/nested/alignment - Timeframe alignment
- /api/nested/performance - Model performance
- /api/nested/features - Feature importance
- /api/walk-forward/* - Walk-forward validation

Admin Endpoints (10+):

- /api/admin/scheduler-status - Scheduler monitoring
- /api/admin/trigger-scheduler - Manual trigger
- /api/admin/table-counts - BigQuery stats
- /api/users - User management CRUD

Auth Endpoints (4):

- /api/auth/login - User login
- /api/auth/verify - Token verification
- /api/auth/change-password - Password change
- /health - Health check

7. Cloud Functions Inventory

7.1 Active Cloud Functions (33 Total)

Function Name	Purpose	Trigger	Status
bulletproof-fetcher	Primary data fetcher (all APIs)	HTTP/Scheduler	PRIMARY
twelvedata-fetcher	TwelveData backup fetcher	HTTP/Scheduler	BACKUP
gap-detector	Data gap detection & fill	Scheduler	ACTIVE
data-deduplication	Remove duplicate records	Scheduler	ACTIVE
ml-daily-analysis	ML feature calculation	Scheduler	ACTIVE
ml-daily-inference	ML predictions	Scheduler	ACTIVE
walk-forward-validation	ML validation system	HTTP	ACTIVE
etf-analytics-fetcher	ETF data collection	Scheduler	ACTIVE
market-movers-fetcher	Top gainers/losers	Scheduler	ACTIVE
opportunity-report-generator	Daily reports	Scheduler	ACTIVE
smart-search	NLP search engine	HTTP	ACTIVE
pdf-report-generator	PDF generation	HTTP	ACTIVE

7.2 Redundant Functions (Candidates for Removal)

Function	Reason	Recommendation
cloud_function_5min	Replaced by bulletproof-fetcher	REMOVE
cloud_function_daily	Replaced by bulletproof-fetcher	REMOVE
cloud_function_hourly	Replaced by bulletproof-fetcher	REMOVE
cloud_function_stocks_daily	Duplicate of bulletproof	REMOVE
cloud_function_stocks_hourly	Duplicate of bulletproof	REMOVE
cloud_function_stocks_5min	Duplicate of bulletproof	REMOVE
cloud_function_daily_stocks	Duplicate functionality	REMOVE
cloud_function_max_quota	Replaced by bulletproof	REMOVE
cloud_function_all_assets	Merged into bulletproof	REMOVE
cloud_function_multi_source	Merged into bulletproof	REMOVE

8. Database Schema (BigQuery)

8.1 Dataset Structure

Project: aialgotradehits

Primary Dataset: crypto_trading_data

ML Dataset: ml_models

Total Tables: 113

Total Records: 1.2B+

Core Tables (Clean Schema):

- stocks_daily_clean (4M+ records) - Daily stocks
- stocks_hourly_clean - Hourly stocks
- stocks_5min_clean - 5-minute stocks
- crypto_daily_clean (2.7M records) - Daily crypto
- crypto_hourly_clean - Hourly crypto
- crypto_5min_clean - 5-minute crypto
- etfs_daily_clean - ETF daily
- forex_daily_clean - Forex daily
- indices_daily_clean - Indices daily
- commodities_daily_clean - Commodities daily

Supplemental Tables:

- fred_economic_data (247K records) - Economic indicators
- cmc_crypto_rankings (2.6K records) - CMC rankings
- finnhub_recommendations (1.3K records) - Analyst ratings
- kraken_buy_sell_volume - Buy/sell pressure

ML Tables:

- nested_daily, nested_hourly, nested_5min - Feature tables
- nested_alignment_final - Alignment features
- nested_predictor_v1 - XGBoost model
- v_nested_signals_final - Prediction view

9. Redundancy Analysis

9.1 Redundant Frontend Components

Component	Lines	Reason	Action
EnhancedDashboard.jsx	821	Superseded by SmartDashboard	DELETE
NLPSearch.jsx	628	Integrated into SmartDashboard	DELETE
WeeklyReconciliation.jsx	620	Not used in routing	DELETE
DataReconciliation.jsx	384	Commented out in App.jsx	DELETE
AIAlgoTradeHits.jsx	564	Duplicate landing page	DELETE
AIAlgoTradeHitsReal.jsx	542	Duplicate landing page	DELETE
AdminPanel.jsx	453	Superseded by AdminPanelEnhanced	DELETE
StockPriceWindow.jsx	143	Legacy component	DELETE
SchedulerMonitoring.jsx	506	Moved to Admin panel	REVIEW
DatabaseMonitoring.jsx	539	Duplicate of TableInventory	REVIEW
BillingDashboard.jsx	367	Not implemented	DELETE
DataDownloadControl.jsx	406	Superseded by DataExportDownload	DELETE
AdvancedTradingChart.jsx	17	Empty wrapper	DELETE

Estimated Lines to Remove: 5,373 lines (18% reduction)

9.2 Redundant Python Scripts (Root Directory)

The root directory contains 349 Python scripts. Many are one-time migration/setup scripts that should be archived:

Categories for Cleanup:

- backfill_*.py (15 files) - Historical data migration
- create_*.py (40 files) - Table/schema creation
- migrate_*.py (18 files) - Data migration
- check_*.py (15 files) - Validation scripts
- fix_*.py (12 files) - Bug fix scripts
- convert_*.py (7 files) - Document conversion
- generate_*.py (10 files) - Report generation
- setup_*.py (12 files) - Initial setup

Recommendation: Move to /archive folder, keep only active utility scripts.

10. NLP Development Roadmap

10.1 Current NLP Capabilities

Implemented Features:

- Basic keyword search across all asset types
- Filter parsing (oversold, overbought, high volume)
- Asset type detection (stocks, crypto, forex, etc.)
- Timeframe extraction (daily, weekly, hourly)
- Voice search via Web Speech API

Search Engine Location: cloud_function_api/nlp_query_engine.py

Integration: /api/analysis/nlp-search endpoint

10.2 NLP Enhancement Roadmap

Phase	Feature	Description	Timeline
1	Intent Classification	Detect user intent (search, analyze, compare)	Q1
1	Entity Extraction	Extract symbols, amounts, dates	Q1
2	Conversational Memory	Remember context across queries	Q2
2	Gemini 3 Integration	LLM-powered understanding	Q2
3	Strategy Generation	Generate trading strategies from NLP	Q3
3	Code-Free Customization	Build dashboards via NLP	Q3
4	Autonomous Agents	Self-executing trade analysis	Q4
4	Voice Commands	Full voice-controlled trading	Q4

10.3 Target: Zero-Code Administration

Vision: All system administration via natural language

Example Commands:

- "Show me oversold tech stocks with rising volume"
- "Create an alert when AAPL crosses above \$250"
- "Backtest RSI strategy on crypto for last 6 months"
- "Generate weekly report for my portfolio"
- "Add new data source for options data"
- "Deploy new ML model for sector rotation"

Implementation:

- Gemini 3 Pro for intent understanding
- Function calling for action execution
- BigQuery integration for data queries
- Vertex AI for ML operations

11. Environment Configuration

11.1 Current Environment Variables

Frontend (.env files):

- VITE_API_BASE_URL - API endpoint URL
- VITE_GOOGLE_CLIENT_ID - OAuth client ID

Backend (Secret Manager):

- TWELVEDATA_API_KEY - TwelveData API
- FRED_API_KEY - FRED economic data
- CMC_API_KEY - CoinMarketCap API
- FINNHUB_API_KEY - Finnhub API
- KRAKEN_API_KEY - Kraken API
- GCP_PROJECT_ID - Google Cloud project
- BIGQUERY_DATASET - Dataset name

Recommendation: Centralize all configuration into a single config service that can be modified via admin panel without code changes.

11.2 Proposed Config Architecture

Config Service Design:

1. Create BigQuery table: system_configuration
2. Admin UI for config management
3. API endpoint: /api/admin/config
4. Real-time config updates without deploy
5. Feature flags for gradual rollouts
6. Environment-specific overrides (dev/staging/prod)

Config Categories:

- api_keys: External API credentials
- feature_flags: Enable/disable features
- rate_limits: API throttling settings
- ml_params: Model hyperparameters
- ui_settings: Dashboard customization
- alerts: Alert thresholds and rules

12. Refactoring Recommendations

12.1 Immediate Actions (High Priority)

1. Remove Redundant Components (5,373 lines)

- Delete 13 unused components identified in Section 9.1
- Update imports in App.jsx
- Test all routes after removal

2. Consolidate Cloud Functions

- Keep only bulletproof-fetcher as primary
- Archive 10 redundant functions
- Update scheduler configurations

3. Archive Root Python Scripts

- Move 150+ one-time scripts to /archive
- Keep only active utilities
- Document what each script does

4. Clean Deploy Folders

- Remove deploy-temp/, deploy-clean/, etc.
- Keep only dist/ for production builds

12.2 Short-Term Improvements (1-2 Months)

1. Component Library

- Extract common UI patterns into shared components
- Create consistent button, card, table styles
- Implement proper TypeScript types

2. State Management

- Implement React Context for global state
- Add Zustand or Redux for complex state
- Cache API responses properly

3. API Layer Cleanup

- Consolidate duplicate endpoints
- Add proper error handling
- Implement request caching

4. Testing Infrastructure

- Add Jest + React Testing Library
- Write tests for critical paths
- Set up CI/CD pipeline

12.3 Long-Term Vision (3-6 Months)

1. Full NLP Integration

- Gemini 3 Pro integration
- Conversational interface
- Code-free administration

2. Real-Time Features

- WebSocket for live data

- Push notifications
- Live trading integration

3. Mobile App

- React Native port
- Native charts
- Biometric auth

4. Enterprise Features

- Multi-tenant support
- Role-based access
- Audit logging
- White-label capability