

# Market Analysis Platform

## CONTENTS

1. Overview
2. Technology Stack
3. Database Schema
4. Core Tables
5. API Endpoints
6. Market Data
7. Portfolio
8. Alerts
9. Analysis
10. External APIs
11. Development Phases
12. Phase 1: Foundation (Week 1-2)
13. Phase 2: Data Integration (Week 3-4)
14. Phase 3: Analysis Engine (Week 5-7)
15. Phase 4: Portfolio & Alerts (Week 8-9)
16. Phase 5: Polish & Deploy (Week 10)
17. Technical Indicators
18. Sentiment Analysis
19. Deployment
20. Docker Compose
21. Notes

## Market Analysis Platform - Project Plan

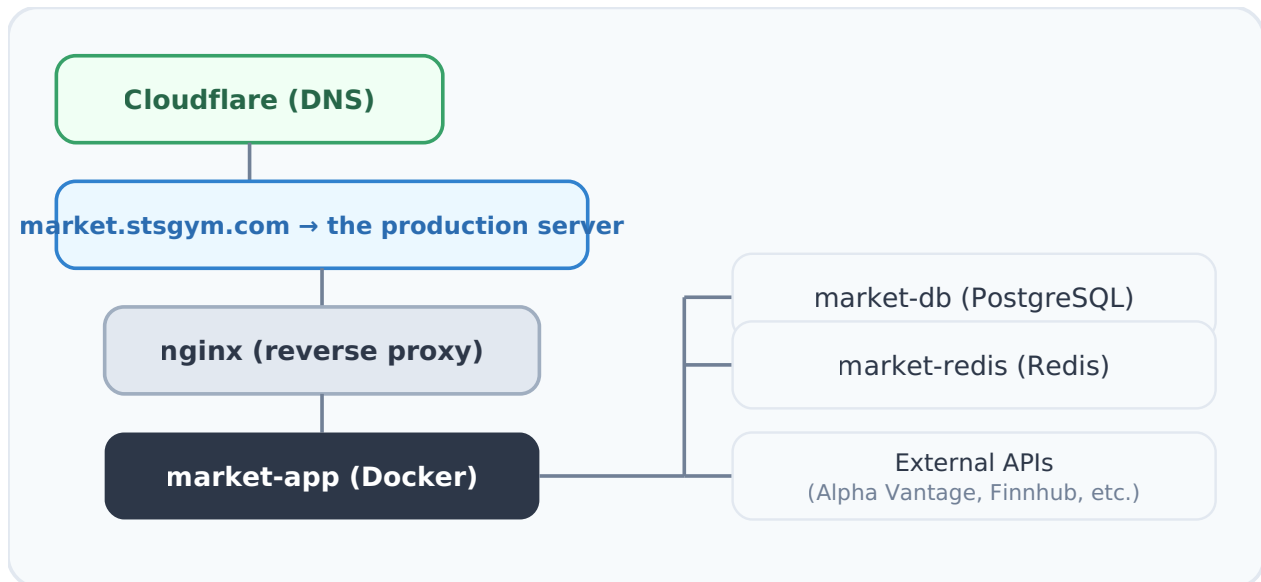
**Created:** 2026-03-26 **Domain:** market.stsgym.com **Port:** TBD (suggest 12770) **Server:** miner (the production server) **Repository:** TBD

---

# Overview

A comprehensive market research and analysis platform providing: - Real-time market data aggregation - Technical indicator analysis - Sentiment analysis from news/social - Portfolio tracking and alerts - Industry comparison reports

---



pha Vantage, Finnhub, etc.)

---

## Technology Stack

### Component Technology

Backend	Node.js/Express (same as 51O8)
Database	PostgreSQL 15
Cache/Queue	Redis + Bull
Frontend	Handlebars + Bootstrap
Charts	Chart.js
AI Analysis	Ollama (gemma3:1b)
SSO	auth.stsgym.com

---

## Database Schema

### Core Tables

```
-- Market data
CREATE TABLE market_symbols (
  id SERIAL PRIMARY KEY,
```

```

symbol VARCHAR(10) UNIQUE NOT NULL,
name VARCHAR(255),
exchange VARCHAR(20),
sector VARCHAR(100),
industry VARCHAR(100),
market_cap BIGINT,
created_at TIMESTAMP DEFAULT NOW()
);

CREATE TABLE market_quotes (
    id SERIAL PRIMARY KEY,
    symbol_id INTEGER REFERENCES market_symbols(id),
    price DECIMAL(18,4),
    change_amount DECIMAL(18,4),
    change_percent DECIMAL(8,4),
    volume BIGINT,
    high_52w DECIMAL(18,4),
    low_52w DECIMAL(18,4),
    pe_ratio DECIMAL(10,2),
    dividend_yield DECIMAL(6,4),
    timestamp TIMESTAMP DEFAULT NOW()
);

-- User portfolios
CREATE TABLE portfolios (
    id SERIAL PRIMARY KEY,
    user_id INTEGER NOT NULL,
    name VARCHAR(100),
    description TEXT,
    created_at TIMESTAMP DEFAULT NOW()
);

CREATE TABLE portfolio_holdings (
    id SERIAL PRIMARY KEY,
    portfolio_id INTEGER REFERENCES portfolios(id),
    symbol_id INTEGER REFERENCES market_symbols(id),
    shares DECIMAL(18,6),
    average_cost DECIMAL(18,4),
    added_at TIMESTAMP DEFAULT NOW()
);

```

```

-- Analysis
CREATE TABLE market_analysis (
    id SERIAL PRIMARY KEY,
    symbol_id INTEGER REFERENCES market_symbols(id),
    analysis_type VARCHAR(50),
-- 'technical', 'fundamental', 'sentiment'
    data JSONB,
    score INTEGER,
    created_at TIMESTAMP DEFAULT NOW()
);

-- Alerts
CREATE TABLE price_alerts (
    id SERIAL PRIMARY KEY,
    user_id INTEGER NOT NULL,
    symbol_id INTEGER REFERENCES market_symbols(id),
    alert_type VARCHAR(20), -- 'above', 'below',
'change_percent'
    target_price DECIMAL(18,4),
    triggered BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT NOW()
);

```

---

## API Endpoints

### Market Data

Endpoint	Method	Purpose
/api/symbols	GET	List all symbols
/api/symbols/:symbol	GET	Symbol details
/api/symbols/:symbol/quote	GET	Current quote
/api/symbols/:symbol/history	GET	Price history
/api/symbols/:symbol/analysis	GET	Technical analysis
/api/sectors	GET	Sector list
/api/sectors/:sector/performance	GET	Sector performance

## Portfolio

Endpoint	Method	Purpose
/api/portfolio	GET	User portfolios
/api/portfolio/:id	POST	Create portfolio
/api/portfolio/:id/holdings	GET	Portfolio holdings
/api/portfolio/:id/add	POST	Add holding
/api/portfolio/:id/remove	POST	Remove holding

## Alerts

Endpoint	Method	Purpose
/api/alerts	GET	User alerts
/api/alerts	POST	Create alert
/api/alerts/:id	DELETE	Delete alert

## Analysis

Endpoint	Method	Purpose
/api/analysis/:symbol/technical	GET	Technical indicators
/api/analysis/:symbol/fundamental	GET	Fundamental analysis
/api/analysis/:symbol/sentiment	GET	News sentiment
/api/analysis/:symbol/prediction	GET	AI prediction

---

## External APIs

API	Purpose	Free Tier
Alpha Vantage	Stock quotes, indicators	5 calls/min
Finnhub	News, company info	60 calls/min
Yahoo Finance	Market data	Free
NewsAPI	News headlines	100 req/day

---

## Development Phases

### Phase 1: Foundation (Week 1-2)

- Create project directory structure
- Initialize Node.js project
- Set up PostgreSQL database
-

- Set up Redis
- Configure SSO integration
- Create Docker Compose
- Basic routes and middleware

## **Phase 2: Data Integration (Week 3-4)**

- Alpha Vantage API integration
- Finnhub API integration
- Quote storage and caching
- Historical data storage
- Sector/industry classification

## **Phase 3: Analysis Engine (Week 5-7)**

- Technical indicators (RSI, MACD, SMA, etc.)
- Fundamental ratios
- News sentiment analysis
- AI prediction model
- Analysis job queue

## **Phase 4: Portfolio & Alerts (Week 8-9)**

- Portfolio CRUD
- Holdings tracking
- Price alerts
- Email notifications
- Dashboard charts

## Phase 5: Polish & Deploy (Week 10)

- UI polish
  - Performance optimization
  - Security audit
  - Documentation
  - Production deployment
- 

## Technical Indicators

Indicator	Type	Description
SMA	Trend	Simple Moving Average
EMA	Trend	Exponential Moving Average
RSI	Momentum	Relative Strength Index
MACD	Momentum	Moving Average Convergence Divergence
Bollinger	Volatility	Bollinger Bands
ATR	Volatility	Average True Range
Stochastic	Momentum	Stochastic Oscillator
ADX	Trend	Average Directional Index

---

## Sentiment Analysis

Sources: - News headlines (NewsAPI, Finnhub) - Social media mentions (Twitter API - limited) - Analyst ratings - Earnings call transcripts

Processing: - Ollama for NLP sentiment - Store sentiment scores over time - Correlate with price movements

---

## Deployment

### Docker Compose

```
version: '3.8'
services:
  app:
    build: .
    container_name: market-app
```

```
ports:
  - "127.0.0.1:12770:12770"
environment:
  - DATABASE_URL=postgresql://market:market123@db:5
market
  - REDIS_URL=redis://redis:6379/0
  - SESSION_SECRET=${SESSION_SECRET}
  - COOKIE_SECRET=${COOKIE_SECRET}
  - ALPHA_VANTAGE_API_KEY=${ALPHA_VANTAGE_API_KEY}
  - FINNHUB_API_KEY=${FINNHUB_API_KEY}
depends_on:
  - db
  - redis

db:
  image: postgres:15
  container_name: market-db
  environment:
    - POSTGRES_USER=market
    - POSTGRES_PASSWORD=market123
    - POSTGRES_DB=market
  volumes:
    - postgres_data:/var/lib/postgresql/data

redis:
  image: redis:7
  container_name: market-redis

volumes:
  postgres_data:
```

---

## Notes

- Reuse 5108 patterns (Ollama integration, job queues)
- Share stock\_symbols database with 5108
- Consider WebSocket for real-time updates
- Rate limit external API calls
- Cache quotes in Redis (5-min TTL)

---

*Last updated: 2026-03-26*