

# Product Requirements Document

## Fundamental Analysis Feature Addition

**Version:** 1.0

**Date:** December 15, 2024

**Status:** Draft

**Parent Product:** Terminal.AI - Agentic Stock Analysis App

---

### 1. Executive Summary

#### 1.1 Purpose

Extend the existing Terminal.AI stock analysis platform with comprehensive fundamental analysis capabilities, providing retail traders with deep insights into company financials, growth trends, and ownership structure.

#### 1.2 Background

The current MVP successfully delivers technical analysis. Users have requested fundamental analysis to make more informed investment decisions by combining both technical and fundamental insights.

#### 1.3 Goals

- Provide 5 years of quarterly financial data (EPS, Sales, Other Income)
  - Display shareholding patterns (Promoter/FII/DII/Public holdings)
  - Calculate and visualize growth trends and rolling averages
  - Generate AI-powered fundamental analysis insights
  - Enable comparison between earnings and sales growth
- 

### 2. Product Overview

#### 2.1 Feature Description

A new "Fundamental Analysis" tab that displays:

- Quarterly financial metrics for the last 5 years (20 quarters)
- Visual comparison charts (Earnings vs Sales, Sales vs Other Income)
- Rolling averages (2-quarter) for EPS and Sales
- Yearly EPS breakout with growth rates

- Shareholding pattern and float percentage
- AI-generated fundamental analysis commentary

## 2.2 Target Users

- Retail traders seeking long-term investment opportunities
- Investors who combine technical and fundamental analysis
- Users who want to understand company financial health
- Traders looking for growth stocks based on earnings and sales

## 2.3 User Value Proposition

**Before:** Users only see technical indicators (price, volume, moving averages)

**After:** Users get complete picture with financial performance, growth trends, and ownership structure

---

## 3. User Stories

### 3.1 Primary User Stories

#### US-01: View Quarterly Financial Trends

- **As a** retail trader
- **I want to** see EPS and Sales trends for the last 5 years
- **So that** I can identify consistent growth patterns

#### US-02: Analyze Growth Rates

- **As an** investor
- **I want to** compare earnings growth vs sales growth quarter-by-quarter
- **So that** I can evaluate profit margin trends

#### US-03: Understand Shareholding

- **As a** trader
- **I want to** see the shareholding breakdown (Promoter/FII/DII/Public)
- **So that** I can assess stock float and ownership stability

#### US-04: Identify Trends with Rolling Averages

- **As an** analyst

- **I want to** see 2-quarter rolling averages for EPS and Sales
- **So that** I can smooth out volatility and identify true trends

## **US-05: Get AI Fundamental Insights**

- **As a** retail investor
  - **I want to** receive AI-generated fundamental analysis
  - **So that** I can understand complex financial data easily
- 

## **4. Functional Requirements**

### **4.1 Data Collection & Processing**

#### **FR-F01: Quarterly Income Statement Data**

- System shall fetch quarterly income statements for last 5 years (20 quarters minimum)
- Data points required: EPS, Revenue, Other Income, Date
- Source: Financial Modeling Prep (FMP) API
- Fallback: If 20 quarters not available, fetch maximum available

#### **FR-F02: Shareholding Data**

- System shall fetch current institutional ownership data
- Data points: Promoter %, FII %, DII %, Public %, Float %
- Update frequency: Latest available quarter
- Source: FMP API institutional-holder endpoint

#### **FR-F03: Data Validation**

- System shall validate all numeric data for null/zero values
- Missing data shall be marked as "N/A" with user notification
- Outliers (>3 standard deviations) shall be flagged

### **4.2 Calculations & Metrics**

#### **FR-F04: Rolling Averages**

- System shall calculate 2-quarter rolling average for EPS
- System shall calculate 2-quarter rolling average for Sales

- Formula:  $(Q_n + Q_{n-1}) / 2$

#### **FR-F05: Growth Rate Calculations**

- Quarter-over-Quarter (QoQ) growth:  $((Q_n - Q_{n-1}) / Q_{n-1}) \times 100$
- Year-over-Year (YoY) growth:  $((Q_n - Q_{n-4}) / Q_{n-4}) \times 100$
- CAGR (5-year):  $((\text{Final Value} / \text{Initial Value})^{(1/5)} - 1) \times 100$

#### **FR-F06: Yearly Aggregation**

- System shall aggregate quarterly EPS into yearly totals
- Display yearly EPS for last 5 years (2020-2024)
- Calculate year-over-year growth rates

#### **FR-F07: Comparison Metrics**

- Earnings vs Sales Growth: Side-by-side percentage comparison
- Sales vs Other Income: Percentage contribution of other income to total revenue
- Profit Margin Proxy:  $(\text{EPS} / \text{Sales})$  trend over time

#### **FR-F08: Float Calculation**

- $\text{Float} = 100\% - \text{Promoter \%}$
- Display as percentage of total shares available for trading

### **4.3 Visualization Requirements**

#### **FR-F09: Quarterly EPS Chart**

- Type: Line chart with 20 data points
- X-axis: Quarter dates (e.g., Q1 2020, Q2 2020...)
- Y-axis: EPS value
- Additional line: 2-quarter rolling average (dashed line)
- Color: Primary color for EPS, secondary color for rolling avg

#### **FR-F10: Quarterly Sales Chart**

- Type: Line chart with 20 data points
- X-axis: Quarter dates
- Y-axis: Revenue in millions/billions

- Additional line: 2-quarter rolling average
- Tooltip: Show exact values on hover

#### **FR-F11: Earnings vs Sales Growth Chart**

- Type: Dual-axis line chart or grouped bar chart
- X-axis: Quarter dates
- Y-axis Left: EPS growth %
- Y-axis Right: Sales growth %
- Legend: Clear differentiation between both metrics

#### **FR-F12: Sales vs Other Income Chart**

- Type: Stacked bar chart or percentage area chart
- Show: Other Income as % of Total Revenue per quarter
- Highlight quarters where other income >20% (unusual dependency)

#### **FR-F13: Yearly EPS Breakout**

- Type: Bar chart
- X-axis: Years (2020-2024)
- Y-axis: Annual EPS
- Annotation: Show YoY growth % on top of each bar

#### **FR-F14: Shareholding Pattern**

- Type: Pie chart or donut chart
- Segments: Promoter, FII, DII, Public
- Labels: Percentage + absolute value (if available)
- Color scheme: Distinct colors for each segment
- Callout: Float percentage prominently displayed

### **4.4 AI Analysis Requirements**

#### **FR-F15: AI Fundamental Commentary**

- System shall generate structured fundamental analysis using LLM
- Sections required:
  1. Financial Performance Overview

2. Growth Trajectory Analysis
3. Profitability Assessment
4. Shareholding Pattern Insights
5. Red Flags & Strengths
6. Overall Fundamental Rating (1-10 scale)

#### **FR-F16: AI Prompt Structure**

- Input data: All calculated metrics, growth rates, trends
- Context: Industry averages (if available)
- Output format: Structured markdown with clear sections
- Tone: Educational, balanced (no trading advice)

#### **FR-F17: Combined Analysis (Technical + Fundamental)**

- Optional: AI shall provide combined insights when both analyses are run
- Compare: Technical trend vs Fundamental strength
- Identify: Discrepancies (e.g., strong fundamentals but weak technicals)

### **4.5 User Interface Requirements**

#### **FR-F18: Tab Navigation**

- Add "Fundamental Analysis" tab next to "Technical Analysis"
- Default: Show Technical Analysis on initial load
- State persistence: Remember last viewed tab per session

#### **FR-F19: Loading States**

- Display skeleton loaders for charts while data is fetching
- Show progress indicator: "Fetching financial data... (Step 1/3)"
- Estimated time: "This may take 5-10 seconds"

#### **FR-F20: Error Handling**

- Graceful degradation: Show available data if partial failure
- Clear error messages: "Financial data not available for this stock"
- Retry option: "Try Again" button

#### **FR-F21: Data Freshness Indicator**

- Display: "Last updated: Q3 2024" or "Data as of: Dec 10, 2024"
- Warning: If data is >6 months old, show "Stale Data" warning

## **FR-F22: Responsive Design**

- Charts shall adapt to mobile/tablet screens
  - Stack charts vertically on smaller screens
  - Maintain readability and interactivity
- 

## **5. Non-Functional Requirements**

### **5.1 Performance**

#### **NFR-F01: API Response Time**

- FMP API calls shall complete within 5 seconds
- Total fundamental analysis (including AI) shall complete within 15 seconds
- Parallel processing: Fetch multiple API endpoints concurrently

#### **NFR-F02: Data Caching**

- Cache fundamental data for 24 hours per stock
- Reduce redundant API calls for frequently analyzed stocks
- Cache invalidation: Manual refresh option for users

#### **NFR-F03: Chart Rendering**

- Charts shall render within 2 seconds after data is received
- Smooth animations for data updates
- No janky scroll or lag

### **5.2 Reliability**

#### **NFR-F04: API Failure Handling**

- Retry logic: Up to 3 attempts with exponential backoff
- Timeout: 10 seconds per API call
- Fallback: Display cached data if fresh data unavailable

## **NFR-F05: Data Quality**

- Validate all API responses before processing
- Handle missing quarters gracefully (interpolation or skip)
- Log data quality issues for monitoring

## **5.3 Scalability**

### **NFR-F06: Rate Limiting**

- Respect FMP API rate limits (300 requests/minute on free tier)
- Implement request queuing if limits approached
- Display rate limit warnings to users

### **NFR-F07: Cost Management**

- Monitor FMP API usage to stay within free tier (250 requests/day)
- Optimize queries to fetch multiple data points in single request where possible

## **5.4 Usability**

### **NFR-F08: Accessibility**

- Charts shall support keyboard navigation
- Screen reader compatible labels for all visual elements
- High contrast mode support

### **NFR-F09: Data Export**

- Option to download charts as PNG
- Option to export data as CSV
- Copy AI analysis text to clipboard

---

## **6. Data Sources & API Integration**

### **6.1 Financial Modeling Prep (FMP) API**

#### **Primary Endpoints:**

##### **1. Income Statement (Quarterly)**

- Endpoint: `/income-statement/{symbol}?period=quarter&limit=20`



- Data: EPS, Revenue, Other Income, Date
- Rate Limit: Free tier - 250 requests/day

## 2. Institutional Ownership

- Endpoint: `/institutional-holder/{symbol}`
- Data: Holder name, shares held, % of ownership
- Update: Quarterly

## 3. Key Metrics

- Endpoint: `/key-metrics/{symbol}?period=quarter&limit=20`
- Data: Additional financial ratios if needed

### API Response Format:

```
json
{
  "symbol": "AAPL",
  "date": "2024-09-30",
  "revenue": 94930000000,
  "eps": 1.64,
  "otherIncome": 1200000000,
  "period": "Q3"
}
```

## 6.2 Data Mapping for Indian Stocks

**Challenge:** FMP primarily covers US stocks. For Indian stocks (NSE/BSE):

**Option 1:** Use FMP India-specific endpoints (if available)

- Symbol format: `RELIANCE.NS` or `TCS.BO`
- Verify data availability during implementation

**Option 2:** Integrate secondary API

- Consider: Polygon.io, Alpha Vantage, or Indian-specific providers
- Fallback: Manual data entry or scraping (not recommended)

**Option 3:** User selects market

- US Stocks: Use FMP
- Indian Stocks: Placeholder or alternative API

**Recommendation:** Clarify target market (US vs India) before implementation

---

## 7. Technical Architecture Changes

### 7.1 Backend Changes Summary

#### New Files:

- `app/services/fmp_service.py` - FMP API wrapper
- `app/tools/fundamental_tool.py` - Calculation logic
- `app/models/fundamental.py` - Pydantic models
- `app/agent/nodes.py` - Add `fundamental_analysis_node`

#### Modified Files:

- `app/agent/graph.py` - Add fundamental node to workflow
- `app/routes.py` - Add `/api/analyze/fundamental` endpoint
- `app/config/settings.py` - Add `FMP_API_KEY`

### 7.2 Frontend Changes Summary

#### New Components:

- `FundamentalAnalysis.jsx` - Main container
- `FinancialCharts.jsx` - All financial charts
- `ShareholdingPattern.jsx` - Pie chart component
- `ComparisonCharts.jsx` - Earnings vs Sales
- `TabNavigation.jsx` - Switch between Technical/Fundamental

#### Modified Components:

- `App.jsx` - Integrate tab navigation
- `StockInput.jsx` - Add "Analyze Fundamentals" option

### 7.3 LangGraph Workflow Update

#### Current Flow:

START → Fetch Stock Data → Technical Analysis → END

**New Flow:**

START → Fetch Stock Data → Technical Analysis → Fundamental Analysis → Combine Insights → END

**Node Details:**

- **Fundamental Analysis Node:**
  - Input: Stock symbol, technical data (optional)
  - Process: Fetch FMP data, calculate metrics, generate AI analysis
  - Output: Fundamental data object + AI commentary

## 8. API Contract

### 8.1 New Endpoint: Fundamental Analysis

**Request:**

```
POST /api/analyze/fundamental
Content-Type: application/json

{
  "symbol": "RELIANCE",
  "exchange": "NSE",
  "include_technical": false // Optional: combine with technical
}
```

**Response (Success - 200 OK):**

json

```
{
  "status": "success",
  "data": {
    "symbol": "RELIANCE",
    "company_name": "Reliance Industries Ltd.",
    "data_as_of": "2024-12-15T10:30:00Z",

    "quarterly_data": {
      "dates": ["Q4 2024", "Q3 2024", ..., "Q1 2020"],
      "eps": [65.2, 62.1, 58.3, ...],
      "sales": [250000, 245000, 238000, ...],
      "other_income": [12000, 11500, 11200, ...]
    },

    "rolling_averages": {
      "eps_2q": [63.65, 60.2, ...],
      "sales_2q": [247500, 241500, ...]
    },

    "yearly_eps": {
      "2024": 245.6,
      "2023": 228.4,
      "2022": 198.7,
      "2021": 175.3,
      "2020": 152.1
    },

    "growth_metrics": {
      "eps_cagr_5y": 12.45,
      "sales_cagr_5y": 15.23,
      "latest_quarter": {
        "eps_qoq": 5.0,
        "eps_yoy": 18.5,
        "sales_qoq": 2.0,
        "sales_yoy": 12.8
      },
      "profit_margin_trend": "improving"
    },

    "shareholding": {
      "promoter": 50.11,
      "fii": 23.45,
      "dii": 15.32,
      "public": 11.12,
      "float": 49.89,
      "last_updated": "Q3 2024"
    }
  }
}
```

```

    },

    "comparison_data": {
      "earnings_vs_sales_growth": [
        {"quarter": "Q4 2024", "eps_growth": 5.0, "sales_growth": 2.0},
        {"quarter": "Q3 2024", "eps_growth": 3.8, "sales_growth": 1.5}
      ],
      "other_income_percentage": [
        {"quarter": "Q4 2024", "percentage": 4.8},
        {"quarter": "Q3 2024", "percentage": 4.7}
      ]
    },

    "ai_analysis": {
      "financial_performance": "Reliance has demonstrated consistent growth...",
      "growth_trajectory": "The company shows strong EPS CAGR of 12.45%...",
      "profitability": "Profit margins have been improving consistently...",
      "shareholding_insights": "Strong promoter holding at 50.11%...",
      "strengths": [
        "Consistent revenue growth",
        "Improving profit margins",
        "Strong promoter commitment"
      ],
      "red_flags": [
        "High dependency on other income (if >20%)",
        "Recent deceleration in sales growth"
      ],
      "fundamental_rating": 8.5,
      "rating_explanation": "Strong fundamentals with consistent growth..."
    }
  }
}

```

### Response (Error - 400/404/500):

```

json

{
  "status": "error",
  "error_code": "FMP_DATA_NOT_AVAILABLE",
  "message": "Financial data not available for this stock",
  "details": "FMP API returned no data for symbol: RELIANCE"
}

```

## 8.2 Modified Endpoint: Combined Analysis

### Request:

```
POST /api/analyze/complete
Content-Type: application/json
```

```
{
  "symbol": "RELIANCE",
  "exchange": "NSE"
}
```

## Response:

```
json

{
  "status": "success",
  "data": {
    "symbol": "RELIANCE",
    "technical": { /* existing technical data */ },
    "fundamental": { /* fundamental data as above */ },
    "combined_insights": {
      "overall_signal": "Strong Buy / Hold / Sell",
      "technical_trend": "Bullish",
      "fundamental_strength": "Strong",
      "alignment": "Aligned / Divergent",
      "ai_summary": "Combined analysis shows..."
    }
  }
}
```

---

## 9. Success Metrics

### 9.1 Adoption Metrics

- **Target:** 60% of users who run technical analysis also view fundamental analysis
- **Measurement:** Track tab clicks and API calls

### 9.2 Engagement Metrics

- **Average time on Fundamental tab:** >2 minutes
- **Charts viewed per session:** >3 charts
- **AI analysis read rate:** >80%

### 9.3 Technical Metrics

- **API success rate:** >95%
- **Average load time:** <15 seconds
- **Error rate:** <5%

### 9.4 User Satisfaction

- **NPS Score:** >40 (post-feature launch survey)
  - **Feature usefulness rating:** >4.0/5.0
- 

## 10. Out of Scope (This Release)

The following features are **not included** in this release:

- **✗** Real-time fundamental data updates
  - **✗** Comparison with industry peers
  - **✗** Balance sheet and cash flow statement analysis
  - **✗** Valuation metrics (P/E, P/B, DCF)
  - **✗** Dividend history and yield analysis
  - **✗** Management quality assessment
  - **✗** News sentiment integration
  - **✗** Insider trading alerts
  - **✗** Debt-to-equity ratio tracking
  - **✗** Return on equity (ROE) trends
  - **✗** Custom metric calculator
  - **✗** PDF report generation
  - **✗** Email digest of fundamental changes
- 

## 11. Future Enhancements (Post V1)

### Phase 2 (Next Quarter):

1. Add valuation metrics (P/E, PEG, P/B ratios)
2. Peer comparison (compare with 3-5 similar companies)

- 3. Balance sheet analysis (Assets, Liabilities, Equity)
- 4. Cash flow statement visualization

**Phase 3 (6 months):**

- 5. Management discussion & analysis (MD&A) summary
  - 6. Insider trading tracker
  - 7. Dividend analysis and forecasting
  - 8. Red flag detection (aggressive accounting practices)
- 

**12. Assumptions & Dependencies**

**12.1 Assumptions**

- **FMP API** supports the required stock symbols (US or Indian)
- Users have basic understanding of financial metrics (EPS, Sales, etc.)
- Quarterly data is available for at least 16 quarters (4 years minimum acceptable)
- Shareholding data is updated at least quarterly

**12.2 Dependencies**

- **FMP API:** Active subscription and API key
- **LLM API:** OpenAI or Claude for AI analysis generation
- **Chart.js Library:** For rendering financial charts
- **Backend infrastructure:** Sufficient memory for data processing

**12.3 Risks & Mitigations**

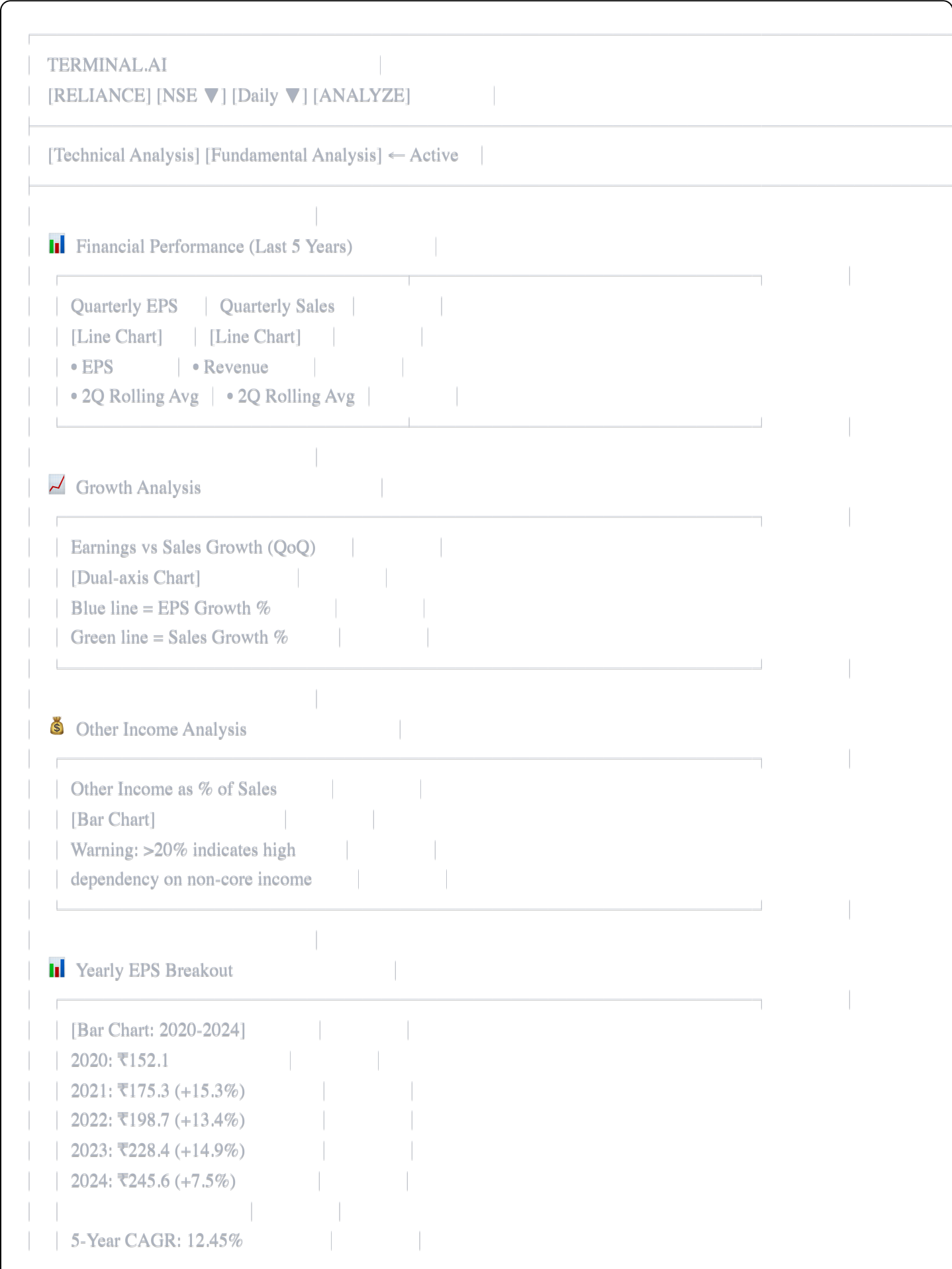
| Risk                                   | Impact | Probability | Mitigation  |
|--|--------|-------------|---|
| FMP data unavailable for Indian stocks | High   | Medium      | Use alternative API or limit to US stocks initially |
| API rate limits exceeded               | Medium | Low         | Implement caching and request throttling            |
| Data quality issues (missing quarters) | Medium | Medium      | Graceful degradation, show available data           |
| Slow chart rendering                   | Low    | Low         | Optimize data payload, use lazy loading             |

---



13. User Interface Mockups

13.1 Tab Layout






|   |                       |  |  |
|---|-----------------------|--|--|
|   |                       |  |  |
|   |                       |  |  |
| 👥 Shareholding Pattern                    |                       |  |  |
|   |                       |  |  |
| [Pie Chart]                               | Breakdown:            |  |  |
|   | • Promoter: 50.11%    |  |  |
| 50.11%                                    | • FII: 23.45%         |  |  |
| Promoter                                  | • DII: 15.32%         |  |  |
|   | • Public: 11.12%      |  |  |
|   |                       |  |  |
|   | Float: 49.89%         |  |  |
|   | (Available for trade) |  |  |
|   |                       |  |  |
| 🤖 AI Fundamental Analysis                 |                       |  |  |
|   |                       |  |  |
| Financial Performance Overview            |                       |  |  |
| Reliance Industries has demonstrated...   |                       |  |  |
|   |                       |  |  |
| Growth Trajectory                         |                       |  |  |
| The company shows strong EPS CAGR of...   |                       |  |  |
|   |                       |  |  |
| Key Strengths:                            |                       |  |  |
| ✓ Consistent revenue growth               |                       |  |  |
| ✓ Improving profit margins                |                       |  |  |
| ✓ Strong promoter holding                 |                       |  |  |
|   |                       |  |  |
| Red Flags:                                |                       |  |  |
| ⚠ Recent deceleration in sales growth     |                       |  |  |
|   |                       |  |  |
| Fundamental Rating: 8.5/10                |                       |  |  |
| [Strong fundamentals with consistent...]  |                       |  |  |
|   |                       |  |  |
|   |                       |  |  |
|   |                       |  |  |
| [Copy Analysis] [Download Charts] [Share] |                       |  |  |





14. Acceptance Criteria

14.1 Data Accuracy





- ✓ All quarterly data matches FMP API response

-  Rolling averages calculated correctly within 0.01% margin of error
-  Growth rates match manual calculations
-  CAGR formula verified against Excel calculations





## 14.2 Visualization

-  All 6 charts render without errors
-  Charts responsive on mobile (320px width)
-  Tooltips show correct values on hover
-  Legend clearly identifies each data series





## 14.3 AI Analysis

-  Analysis generated within 10 seconds
-  Output includes all 5 required sections
-  Fundamental rating provided (1-10 scale)
-  No hallucinations (all statements backed by data)

## 14.4 User Experience

-  Tab switching is instant (<100ms)
-  Loading states visible during data fetch
-  Error messages are user-friendly
-  No layout shift during chart loading

## 14.5 Performance

-  Total load time <15 seconds (including AI)
  -  FMP API calls complete within 5 seconds
  -  Chart rendering <2 seconds after data received
  -  No memory leaks after 10 consecutive analyses
-

## 15. Testing Plan

### 15.1 Unit Tests

- Test FMP service API calls
- Test calculation functions (rolling avg, growth rates, CAGR)
- Test data validation logic
- Test Pydantic models

### 15.2 Integration Tests

- Test LangGraph node execution
- Test end-to-end API flow (request → response)
- Test chart data formatting
- Test error handling for API failures

### 15.3 UI Tests

- Test tab navigation
- Test chart rendering with various data sizes
- Test responsive behavior on mobile
- Test accessibility (keyboard navigation, screen readers)

### 15.4 Load Tests

- Simulate 100 concurrent requests
- Test cache effectiveness
- Monitor API rate limits
- Measure memory usage

### 15.5 User Acceptance Testing (UAT)

- Beta test with 10-20 retail traders
  - Collect feedback on data clarity
  - Validate AI analysis quality
  - Identify edge cases
-

## 16. Documentation Requirements

### 16.1 User Documentation

- Help article: "Understanding Fundamental Analysis"
- Video tutorial: "How to interpret EPS and Sales trends"
- FAQ: Common questions about financial metrics
- Glossary: Define EPS, CAGR, Float, etc.

### 16.2 Technical Documentation

- API documentation for `/api/analyze/fundamental`
  - FMP integration guide
  - Chart.js implementation notes
  - Troubleshooting guide for developers
- 

## 17. Deployment Plan

### 17.1 Rollout Strategy

- **Phase 1 (Week 1):** Deploy to staging environment
- **Phase 2 (Week 2):** Beta test with 20 users
- **Phase 3 (Week 3):** Fix bugs, optimize performance
- **Phase 4 (Week 4):** Production deployment (feature flag enabled)
- **Phase 5 (Week 5):** Monitor metrics, collect feedback

### 17.2 Feature Flag

- Enable "Fundamental Analysis" tab via feature flag
- Gradual rollout: 10% → 50% → 100% of users
- Ability to disable instantly if issues arise

### 17.3 Monitoring

- Track API success/failure rates
- Monitor average load time
- Alert if error rate >5%

- Dashboard: Daily active users viewing fundamental tab

## 18. Appendix

### 18.1 Sample FMP API Response

#### Income Statement (Quarterly):

```
json
{
  "symbol": "AAPL",
  "date": "2024-09-30",
  "calendarYear": "2024",
  "period": "Q3",
  "revenue": 94930000000,
  "costOfRevenue": 54770000000,
  "grossProfit": 40160000000,
  "otherIncome": 1200000000,
  "eps": 1.64,
  "epsdiluted": 1.64
}
```

#### Institutional Ownership:

```
json
[
  {
    "holder": "Vanguard Group Inc",
    "shares": 1270000000,
    "dateReported": "2024-09-30",
    "change": 5000000,
    "percentHeld": 8.2
  }
]
```

### 18.2 Calculation Examples

#### 2-Quarter Rolling Average:

Q1 EPS = 50

Q2 EPS = 55

Q3 EPS = 58

Rolling Avg at Q2 =  $(50 + 55) / 2 = 52.5$

Rolling Avg at Q3 =  $(55 + 58) / 2 = 56.5$

**CAGR (5 years):**

Initial EPS (2020) = 152.1

Final EPS (