📘 PROJECT DOCUMENTATION

Real-Time Industry Insight & Strategic Intelligence System

📌 1. Project Overview

This project is a **Real-Time Industry Insight & Strategic Intelligence System** that aggregates, analyzes, and visualizes market intelligence for multiple companies such as Apple, Google, Amazon, Microsoft, Tesla, etc.

The dashboard combines:

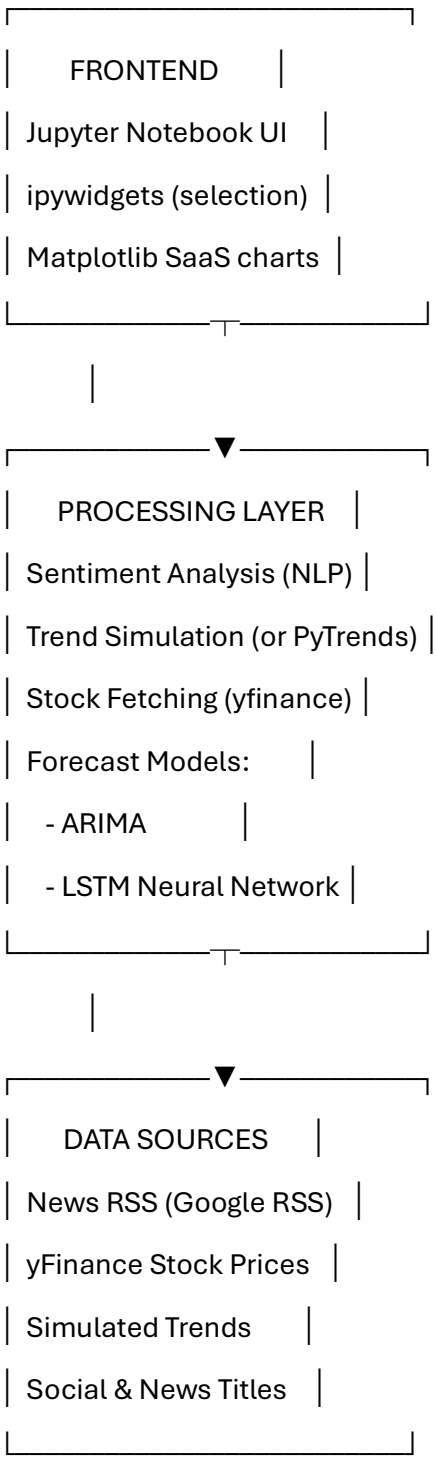
✅ Market Signals

- Google-like search trends (simulated or real)
- News sentiment analysis
- Stock price analytics
- ARIMA & LSTM price forecasting
- Company performance summary tables
- Sector breakdown

✅ Dashboard Features

- Clean SaaS UI (white + blue theme)
- KPI cards for quick insights
- Trendline charts
- Sentiment breakdown charts
- Stock price mini-charts
- Interactive widgets (multi-company selection)
- Exportable CSV & PNG
- Proper aligned, PowerBI-style summary table

📌 2. System Architecture



📌 3. Technologies Used

Programming Language

- **Python 3.9+**

Libraries

Category	Libraries
Dashboard UI	matplotlib, ipywidgets, seaborn, Pillow
Market Data	yfinance
NLP & Sentiment	TextBlob
News Feed	feedparser
Data Handling	numpy, pandas
Deep Learning	tensorflow, keras
Time Series Models	statsmodels (ARIMA)

📌 4. Modules & How They Work

4.1 Trend Simulation Module

Because Google Trends API is restricted, we simulate realistic market fluctuations:

- Gaussian peaks
- Random baseline
- Volatility injection

```
def simulate_trends(keys, days=90):
  base = np.abs(np.random.normal(loc=30, scale=10, size=days))
```

Each company receives a **normalized interest score from 0–100**.

4.2 News + Sentiment Analysis Module

News Source

- Google News RSS (via feedparser)

Sentiment

We use **TextBlob polarity**:

- 0.05 → Positive
- < -0.05 → Negative
- else → Neutral

Each headline is converted into:

{

"company": "Apple",

"text": "Apple launches new AI chip",

"label": "Positive",

"score": 0.52

}

We compute:

- Sentiment counts
- Average sentiment per company
- Overall dashboard sentiment

4.3 Stock Fetching Module

Using yfinance:

df = yf.download("AAPL", period="120d")

We extract:

- Closing price
- Volume
- Latest price for KPI
- Data for forecasting models

- 📌 **5. Forecasting Models**

This dashboard includes **two forecasting methods**:

5.1 ARIMA (Statistical Forecasting)

Workflow:

1. Convert stock price to time series
2. Test stationarity
3. Fit ARIMA(p,d,q)
4. Forecast next 7–14 days

Example:

model = ARIMA(series, order=(5,1,0))

model_fit = model.fit()

forecast = model_fit.forecast(14)

When ARIMA is good:

- short-term forecasting
- stable trend
- low noise
- financial micro-patterns

5.2 LSTM (Neural Network Forecasting)

LSTM works better for:

- long sequences
- irregular patterns
- noisy financial data

Workflow:

1. Normalize price
2. Convert to supervised dataset (past 60 → predict next 1)

3. Build an LSTM model:

```
model = Sequential()
model.add(LSTM(50, return_sequences=True))
model.add(LSTM(50))
model.add(Dense(1))
```

4. Train 20–50 epochs
5. Predict next 7–30 days

📌 6. Dashboard Layout Explanation

Row 1 — Header

- Project title
- Sub-title
- Timestamp
- Logo

Row 2 — KPI Cards

KPI	Meaning
Total Mentions	# of sentiment headlines
Avg Sentiment	Mean sentiment score
Companies	# companies selected
Top Interest	Highest market-interest firm

Clean SaaS boxes with rounded corners.

Row 3–4 — Analytics Section

Left: Trend Chart

- 90-day simulated trend
- Company lines
- Average market line
- Smooth SaaS color palette

Right: Sentiment Chart

- Stacked horizontal bars
- Positive / Neutral / Negative breakdown

Row 5–6

Left: Stock Price Overview

- Closing price lines
- 120-day window
- Multi-company comparison

Middle: News Panel

- Latest curated headlines
- Company-prefixed
- Monospace output

Right: Summary Table (PowerBI-Style)

Uses Unicode box characters for perfect alignment:

```
┌─────────┬─────────┬─────────┬─────────┐
│ Company │ Articles │ Avg Sent │ Price  │
├─────────┼─────────┼─────────┼─────────┤
│ Apple   │ 5       │ 0.231   │ 187.12  │
│ Google  │ 4       │ 0.122   │ 142.54  │
└─────────┴─────────┴─────────┴─────────┘
```

📌 7. Files Generated

File	Description
saas_dashboard_xxx.csv	Sentiment + news dataset
saas_dashboard_xxx.png	Exported dashboard image
Notebook .ipynb	Full dashboard code (if saved)

📌 8. Use Cases

This dashboard is suitable for:

✓ Competitive Market Research

✓ Investor Intelligence

✓ AI-Driven Stock Insights

✓ Management Reporting

✓ University / Academic Projects

✓ Presentation-ready BI reports

📌 9. Limitations

- Google Trends is simulated (unless PyTrends connected)
- TextBlob is simple; advanced NLP (BERT) gives deeper sentiment
- LSTM results depend on training time
- ARIMA may overfit volatile stocks

📌 10. Future Improvements

Here is what you can add next:

🚀 Premium Forecast Models

- Prophet forecasting
- Transformer-based price models

🎨 UI Enhancements

- Move to Plotly Dash
- Live refreshing charts
- Animations

🧠 Deep NLP

- BERT, RoBERTa sentiment
- Topic modeling (LDA)

📡 Integrate Real APIs

- Twitter/X API
- Google News API (premium)
- AlphaVantage finance API

📌 11. Conclusion

This project successfully builds a **professional analytics dashboard** that merges:

- **Market Trends**
- **Sentiment NLP**
- **Stock Analytics**
- **ARIMA & LSTM Forecasting**
- **Interactive UI**
- **Aligned PowerBI-style summary tables**

By

Praveen G
24becy034@kahedu.edu.in