# **📊 Market Pulse – Stock Pulse Analyzer**

Market Pulse is a full-stack application that provides \*\*real-time bullish, bearish, or neutral pulse signals\*\* for a given stock ticker by collecting:

- 🟢 Price Momentum (from Alpha Vantage / Twelve Data)

- 📰 Latest News (from NewsAPI)

- 🧠 LLM-powered analysis (Google Gemini)

> A clean single-page React app lets users enter a ticker (e.g., MSFT), and shows:

> - Momentum trend

> - News sentiment

> - Gemini-generated summary + pulse

> - Full JSON data (collapsible view)

---

## ✅ Features Implemented

| Feature | Description |

|--------|-------------|

| 🔄 REST API | `GET /api/v1/market-pulse?ticker=MSFT` |

| 📈 Price Momentum | Fetch last 5 trading-day returns using Alpha Vantage / Twelve Data |

| 📰 News Feed | Latest 5 news headlines using NewsAPI |

| 🤖 LLM Pulse | Gemini API decides bullish/neutral/bearish and gives explanation |

| 🌐 Frontend | React SPA with a chat-style UI |

| 📦 CORS-enabled | Backend supports requests from React frontend |

| 🔒 `.env` config | API keys securely managed via environment variables |

| 📊 Chart | Sparkline chart of returns using Chart.js |

| ⚠️ Error Handling | Fallback UI for API errors / limits |

| 📄 Docs | Complete project structure, setup, usage |

---

## 🧠 Data Flow & Project Flow

1. User enters a \*\*stock ticker\*\* (e.g., `MSFT`) in the frontend.

2. React app calls:

`GET /api/v1/market-pulse?ticker=MSFT`

3. FastAPI server:

- Fetches price data from Alpha Vantage or Twelve Data.

- Fetches news from NewsAPI.

- Computes 5-day returns and momentum score.

- Sends data to \*\*Google Gemini LLM\*\* to classify "pulse" and explain.

4. Returns a unified JSON response:

- `momentum`, `news`, `pulse`, `llm\_explanation`

5. React app shows:

- Pulse summary

- Returns chart

- News

- Explanation

- Raw JSON toggle

---

## 🔗 API Details

### `GET /api/v1/market-pulse?ticker=MSFT`

#### 📥 Query Param:

- `ticker`: e.g., `MSFT`, `AAPL`, `NVDA`

#### 📤 Sample Response:

```json

{

"ticker": "MSFT",

"as\_of": "2025-07-17",

"momentum": {

"returns": [-0.3, 0.4, 1.1, -0.2, 0.7],

"score": 0.34

},

"news": [

{

"title": "Microsoft unveils AI chips",

"description": "The new chip will compete with Nvidia",

"url": "https://example.com"

}

],

"pulse": "bullish",

"llm\_explanation": "Momentum is moderately positive (0.34) and 4 of 5 headlines highlight product launches and strong earnings; hence bullish.",

"news\_sentiment": {

"avg\_score": 0.3,

"label": "positive",

"summary": "News is mostly positive based on sentiment analysis."

},

"llm\_sentiment\_meta": {

"summary": "..."

}

}

**🔐 API Keys Used**

| **Key** | **Purpose** |  |
| --- | --- | --- |
| ALPHAVANTAGE\_KEY | For price data |  |
| TWELVEDATA\_KEY | Backup for price data |  |
| NEWSAPI\_KEY | For latest news |  |
| GEMINI\_API\_KEY | Google Gemini LLM |  |

Keys are stored in .env file.

**📁 Code Structure**

**market-pulse/**

**├── backend/**

**│ ├── main.py**

**│ ├── services/**

**│ │ ├── price\_service.py**

**│ │ ├── news\_service.py**

**│ │ └── llm\_service.py**

**│ └── .env**

**├── frontend/**

**│ ├── App.js**

**│ ├── App.css**

**│ └── index.js**

**└── README.md**

**🧪 Sample curl Test**

**curl "http://localhost:8000/api/v1/market-pulse?ticker=MSFT"**

## **🚀 How to Run the Project**

### **🔧 1. Setup Backend (FastAPI)**

**cd backend**

**python -m venv venv**

**source venv/bin/activate # or venv\Scripts\activate on Windows**

**pip install -r requirements.txt**

**Create .env file:**

**ALPHAVANTAGE\_KEY=Q7SDPE19RWWUI6XS**

**TWELVEDATA\_KEY=6784927a16e24775b814a4cdd97e8e35**

**NEWSAPI\_KEY=635e0026d72a499c9268b5c466b86cc6**

**GEMINI\_API\_KEY=AIzaSyBRxxdViO8DzNK4D4uA5WS08cVXHCdrmrk**

**Run FastAPI server:**

**python -m uvicorn main:app --reload**

**🖥️ 2. Setup Frontend (React)**

**cd frontend**

**npm install**

**Npm install chart.js**

**npm start**

**🖼️ Screenshots**

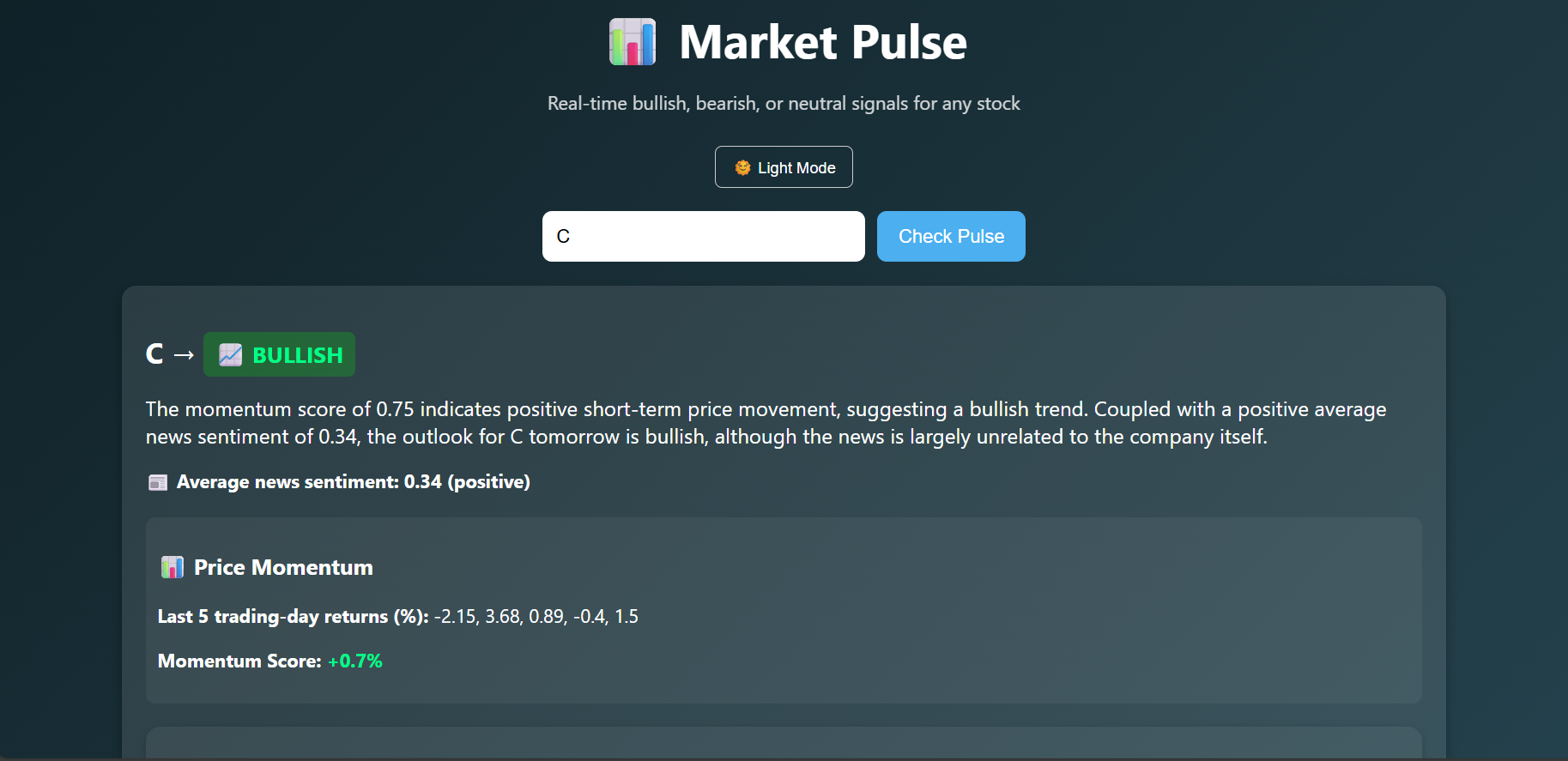


Fig.1 Ticker Entered

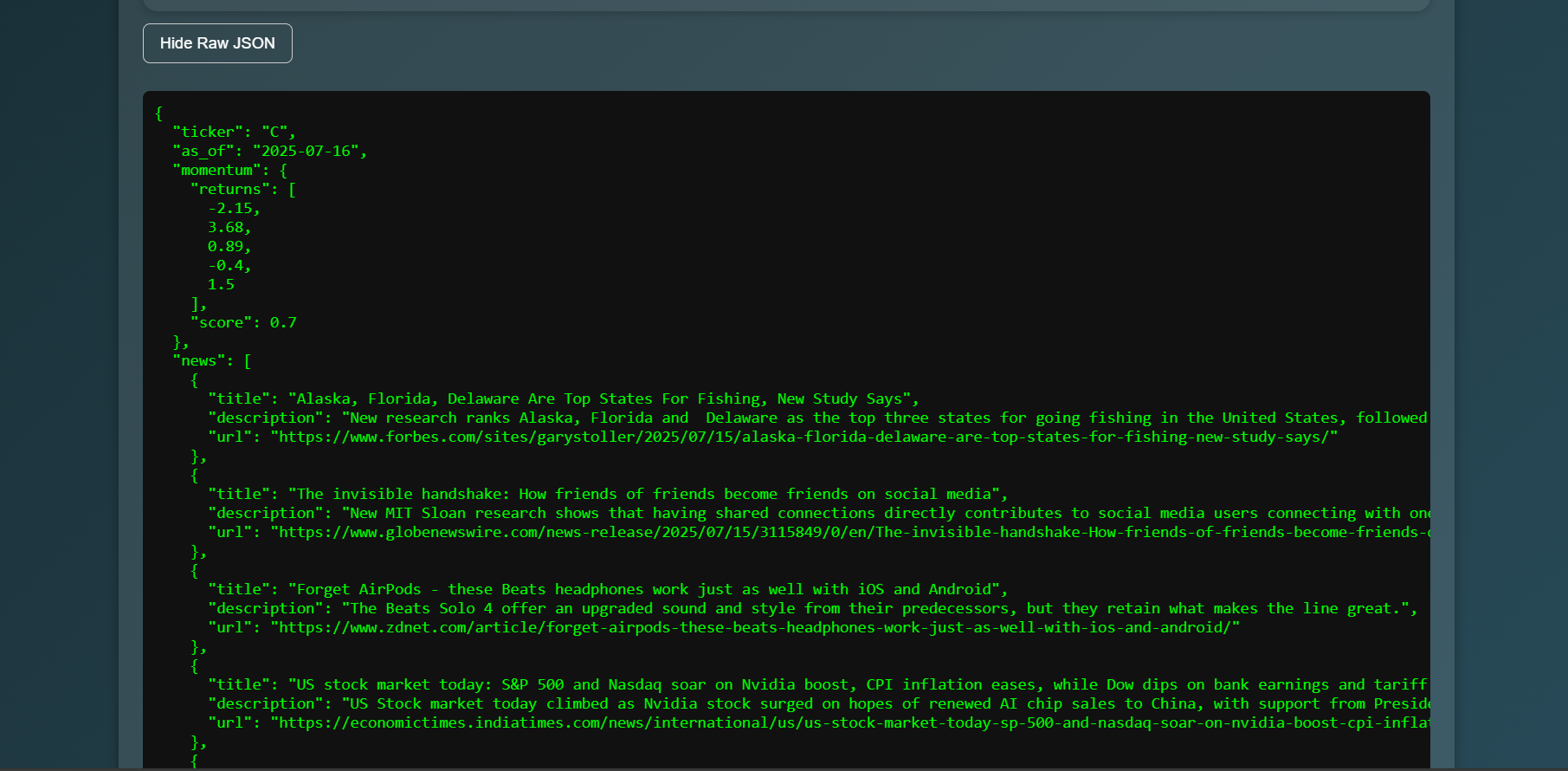
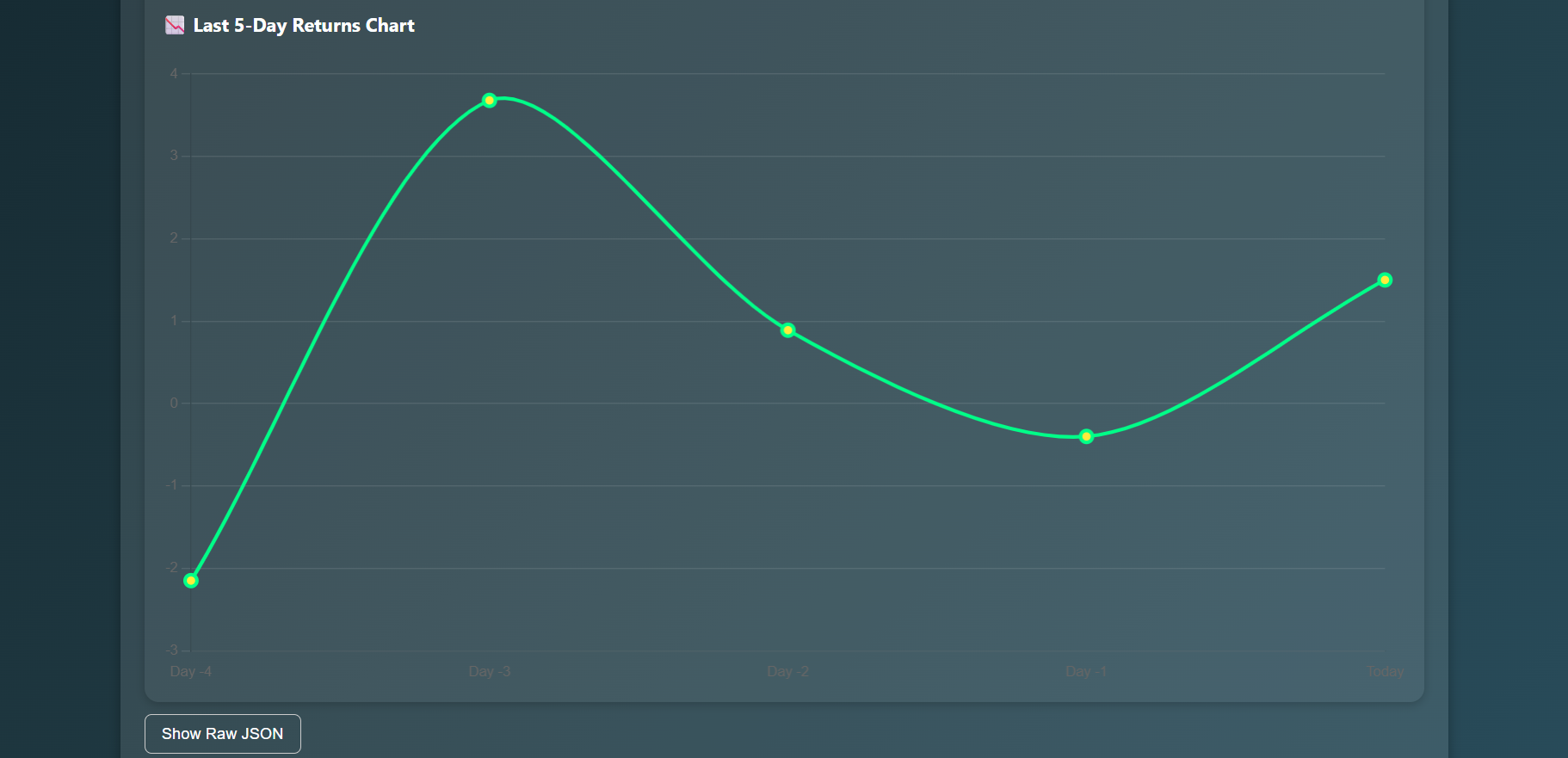


Fig. 3 Ticker’s JSON File

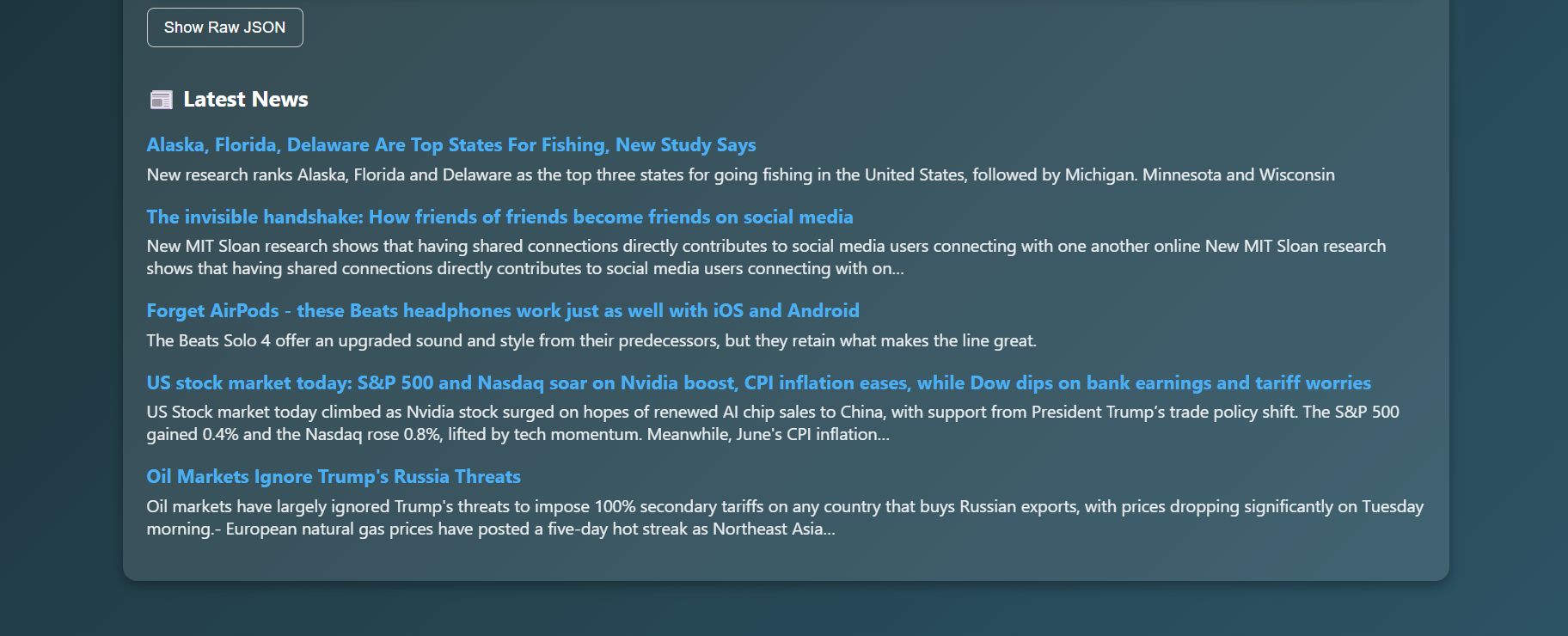


Fig. 4 Ticker’s Latest News

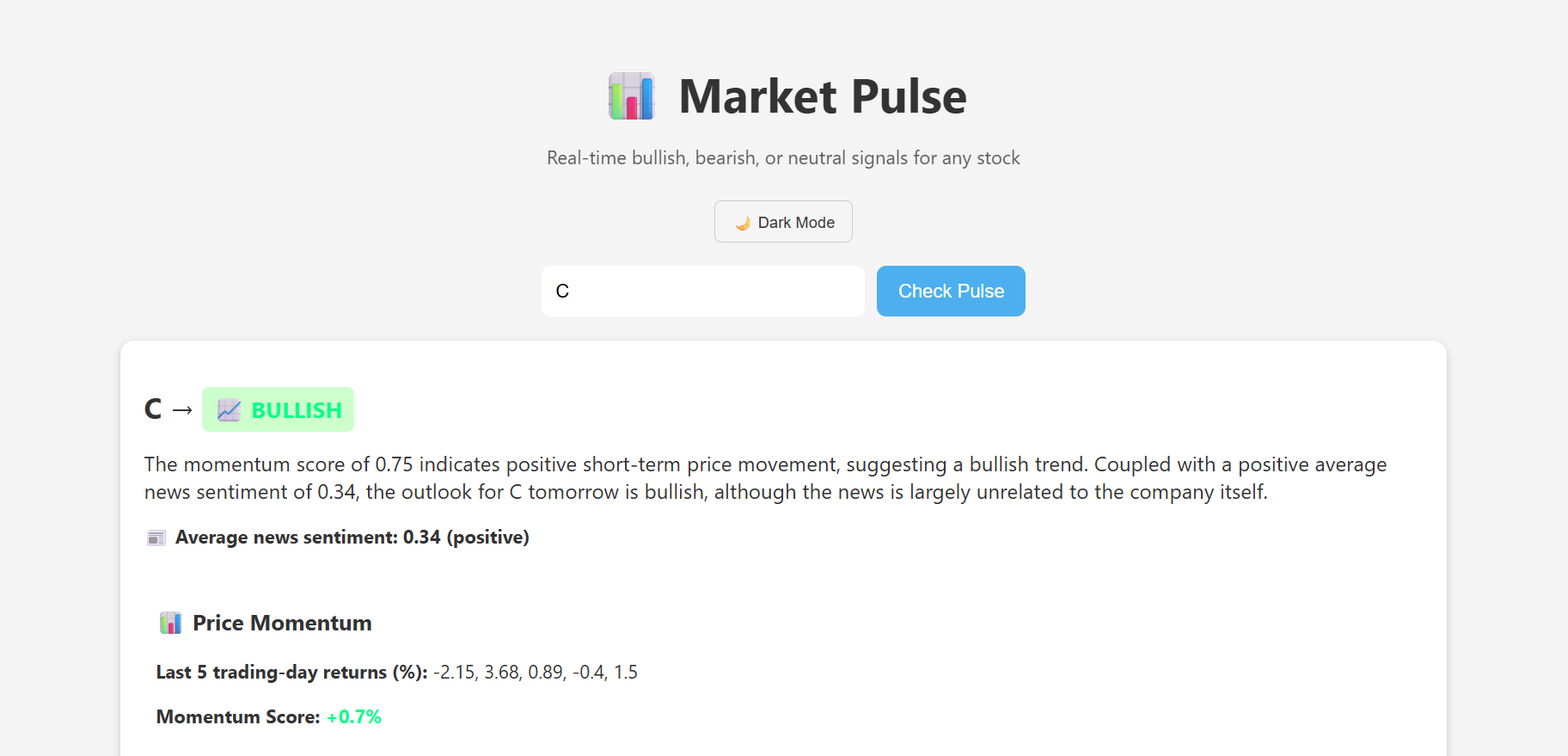


Fig. 5 Ticker’s Mode Changed

## **✨ Design Decisions**

* Used **Google Gemini API** over OpenAI for free tier.
* Used **Alpha Vantage** (fallback: **Twelve Data**) to fetch price data.
* LLM prompt structure includes both news + momentum score.
* All services are modular in services/ for clean separation.
* React chart built with chart.js for rich data visualization.
* Handled API limits gracefully with fallback message.

## **✅ Status: ✅ Project Complete & Working 🚀**

* 💯 Pulse analysis works
* 💡 Gemini LLM explanation working
* 📉 Momentum score computed
* 📰 News headlines shown
* 🧪 API testable with curl or browser