

NewsPulse: Global News Trend Analyzer

Project Statement

The project NewsPulse is an AI-powered platform designed to automatically collect, analyze, and visualize trending topics from global news sources in real-time. By leveraging Natural Language Processing (NLP) and machine learning techniques, the system extracts key insights such as trending entities, topics, sentiments, and geographical patterns, enabling users to understand emerging global narratives at a glance.

Outcomes

- Automated Global News Aggregation
- AI-Powered Trend Detection Engine
- Advanced NLP Analysis Skill Gap Identification
- Interactive Data Visualization Dashboard
- Geo-tagged News Mapping
- Scalable and Modular System Architecture
- Knowledge and Research Utility
- Reduced Information Overload
- Foundational Product for Future Expansion

Modules to be Implemented

- Introduction and System Design
- News Data Collection
- Text Preprocessing & Cleaning
- NLP – Topic & Keyword Extraction
- Trend Detection Engine
- Sentiment & Emotion Analysis
- Geo-tagging & News Heatmaps
- Interactive Dashboard Development
- Testing, Optimization & Documentation
- Deployment & Final Presentation

Milestone 1: Module 1 – Introduction & System Design

Focus:

- Understand the problem domain: global news, trend analysis, information overload
- Define project goals and scope
- Plan architecture and data flow
- Choose tech stack (APIs, NLP libraries, backend/frontend)

Chosen Tech Stack:

- Backend: Python (FastAPI or Flask)

- Frontend: React.js or Streamlit
- Database: PostgreSQL / MongoDB
- AI/ML: Hugging Face transformers, spaCy, BERT, etc.
- News APIs: NewsAPI.org, GNews, RSS feeds

Outcomes:

- Project charter and system architecture diagram
- Tech stack setup (GitHub repo, environment)

Milestone 2: Module 2 – News Data Collection

Focus:

- Integrate news APIs (NewsAPI, GNews, RSS)
- Schedule and automate data fetching
- Store data in structured format (PostgreSQL / MongoDB)

Outcomes:

- Working news ingestion pipeline
- Basic database schema (source, title, date, content)

Purpose: Collect news data from global news sources (APIs, RSS feeds, or scraping)

Output: Structured news data including title, source, summary for further AI analysis

Example Code: News Data Collection using NewsAPI

```
import requests
```

```
API_KEY = 'your_newsapi_key'
```

```
def fetch_news(query, language='en', page_size=5):
```

```
    url =
```

```
    f'https://newsapi.org/v2/everything?q={query}&language={language}&pageSize={page_size}&apiKey={API_KEY}'
```

```
    response = requests.get(url)
```

```
    if response.status_code == 200:
```

```
        news_data = response.json()
```

```
        articles = news_data.get('articles', [])
```

```
        results = []
```

```
        for article in articles:
```

```
            news_item = {
```

```
                "title": article['title'],
```

```
                "source": article['source']['name'],
```

```
                "publishedAt": article['publishedAt'],
```

```
                "url": article['url'],
```

```
                "description": article['description']
```

```

    }
    results.append(news_item)
return results
else:
    print(f"Error fetching news: {response.status_code}")
    return []

if __name__ == "__main__":
    query = "global trends"
    news_items = fetch_news(query)
    for i, news in enumerate(news_items, 1):
        print(f"{i}. Title: {news['title']}")
        print(f"   Source: {news['source']}")
        print(f"   Published At: {news['publishedAt']}")
        print(f"   Description: {news['description']}")
        print(f"   URL: {news['url']}\n")

```

Sample Output (Simulated)

1. Title: Global markets show mixed signals amid economic concerns
 Source: Reuters
 Published At: 2025-09-22T10:00:00Z
 Description: Stocks in Europe and Asia showed volatility as inflation data triggered uncertainty...
 URL: <https://www.reuters.com/article/global-markets-idUSL4N3R32Z5>

2. Title: Climate change summit highlights new policy shifts
 Source: BBC News
 Published At: 2025-09-22T08:30:00Z
 Description: World leaders agreed on new emission targets in an effort to combat climate change...
 URL: <https://www.bbc.com/news/world-58512345>

Frontend and Backend Summary

Frontend (React.js / Streamlit):

- Displays the collected news dynamically.
- Provides a clean and user-friendly interface.
- Allows searching and filtering of news topics.

Backend (Flask / FastAPI):

- Handles API requests and data aggregation.
- Manages connections between frontend and database.
- Ensures scalability and modular integration with AI models.