1. Product Overview

Objective:
Build a **real-time, interactive dashboard** that monitors global supply chain disruptions by aggregating and analyzing news articles.
The dashboard will:

- Detect and classify disruptions (physical, cyber, trade, etc.).
- Score events on a **Supply Chain Severity Index (SCSI)**.
- Display global impact through **geospatial visualization**.
- Allow users to filter by risk type, region, and severity.

Primary Users:

- Supply chain managers, risk analysts, operations planners.
- Industry watchers (logistics, trade, manufacturing).
- Strategic decision-makers.

2. Problem Statement

Current risk monitoring is fragmented, with no single, real-time view combining multiple risk categories from global events.
Users need:

- **Comprehensive coverage** across all disruption types.
- **Consistent severity scoring** that accounts for scale, reach, and impact.
- **Geographic mapping** for quick situational awareness.

3. Scope

In-Scope

1. **Data Ingestion**
    - Pull news from **NewsAPI** with curated disruption keywords.
    - Refresh dataset hourly/daily.
    - Support multi-language expansion (phase 2).
2. **Event Classification**
    - Categories:
        - Physical (natural disaster, strikes, port closures).
        - Cybersecurity.
        - Trade/Geopolitical.

        - Infrastructure/Transport.
    - Tag each article with **country/region**, disruption type, and date.
3. **Severity Index Calculation**
    - Parameters:

1. **Geographic scope** (local vs. international).
2. **Economic importance of impacted node** (e.g., major port vs. rural road).
3. **Type risk multiplier** (cyber > physical > local infra).
4. **Trade dependency impact** (value of trade routes affected).
5. **Population or # countries impacted**.

    - Weighted scoring to produce **0–10 SCSI**.

4. **Visualization (Looker Studio)**
    - **Global map**: color-coded severity by region.
    - **Top disruptions table**: ranked by SCSI.
    - **Trend chart**: disruptions over last X days/weeks.
    - **Filter controls**: date range, category, severity level.
5. **Interactivity**
    - Click on a map region to view top events there.
    - Drill-down into event details (headline, source link, severity breakdown).

Out-of-Scope (for MVP)

- Predictive modelling (forecasting disruptions).
- Automated sentiment analysis of news articles.
- Integration with private trade databases.

4. Data Flow

1. **NewsAPI Query** → curated Boolean search (disruption keywords + supply chain/logistics).
2. **Data Cleaning** → remove irrelevant results.
3. **Classification** → assign risk category, affected region.
4. **Severity Scoring** → apply weighted index.
5. **Google Sheets** → store cleaned/scored dataset.

6. **Looker Studio** → connect live feed, update visuals.

5. Severity Index Framework (Sample Weights)

Parameter	Weight
Geographic Scope (local → global)	25%
Economic Importance of Node	25%
Risk Type Multiplier	20%
Trade Dependency Impact	20%
Population / Countries Impacted	10%

Example:

- Local highway strike: small scope (low score), low trade importance → **SCSI ~ 2.0**.
- Russia–Ukraine conflict escalation: global scope, high trade importance, high dependency → **SCSI ~ 9.5**.

6. Success Metrics

- **Coverage**: % of major global disruptions captured within 24 hrs.
- **Relevance**: ≥ 80% of fetched articles classified correctly.
- **Engagement**: Avg. dashboard session time > 3 min (for portfolio/demo, measure via clicks/filters).

7. Tech Stack (Low-Cost, No-Code First)

- **Data Ingestion**: NewsAPI (Free tier for MVP).
- **Processing**: Google Sheets + App Script (for classification/scoring logic).
- **Visualization**: Looker Studio (Free).
- **Geocoding**: Google Maps API (optional for precise mapping).

Prompt to LLM

You are an analyst extracting structured supply-chain risk signals from news articles.

Return STRICT JSON only (no prose). Do not invent facts. If unknown, use null.

Task:

Given a news article (title, description, content, publishedAt, source, url),

1) Classify disruption type(s):
["Physical","Cyber","Trade/Geopolitical","Infrastructure/Transport"].

2) Extract impacted nodes (ports, canals, rail corridors, factories, borders) with country.

3) Map countries to ISO-3166-1 alpha-2 codes. If multiple countries, return all.

4) Assess scope: Local / Regional / Global (pick one).

5) Assess economic importance of the main node: High / Medium / Low.

6) Identify trade dependency context: High / Medium / Low (e.g., chokepoint, top-10 port, major corridor).

7) Estimate population/countries impacted (bucketed, not precise counts):

PopImpactBucket ∈ ["Minimal","City","State/Province","National","Multi-country"].

8) Provide short 25-word impact summary focused on supply chain consequences.

Additionally produce normalized sub-scores (0–10) using these rubrics:

- ScopeScore: None/Irrelevant=1, Local=1, State/Province=3, Regional(multi-province)=5, Multi-country(2–3)=7 Global(≥4 countries)=9 or 10 (allow 0–10 if article clearly implies smaller/larger).

- NodeImportanceScore: High(critical/global hub)=9 or 10, Medium(Regional Facility)=5, Low(minor local facility)=1, None/Unknown=0.

- RiskTypeScore (pick max if multiple): Cyber=9 or 10, Trade/Geopolitical=8, Infrastructure/Transport=5, Physical=3, None/Unknown=0.

(Rationale: cyber can scale quickly; trade policy can ripple globally.)

- TradeDependencyScore: High=9 or 10, Medium=5, Low=1, None/Unknown=0 (e.g., Suez/Panama/marquee ports=High).

- PopulationImpactScore: Minimal(few people)=0, City=2, State/Province=4, National=7, Multi-country=9 or 10.

Also compute:

- SuggestedModelScore (0–10): weighted preview using the following weights:

0.25*Scope + 0.25*NodeImportance + 0.20*RiskType + 0.20*TradeDependency + 0.10*PopulationImpact.

(This is advisory; final score will be computed downstream.)

- Confidence (0–1): confidence in extraction & scoring.

- EvidenceFieldsUsed: which fields you relied on ["title","description","content"].

Constraints:

- Be conservative: if the article is speculative or outdated, lower ScopeScore or add "is_speculative": true.

- If the event is minor (e.g., local road protest) but headline is sensational, cap SuggestedModelScore ≤ 4.

- Never include text outside JSON. No markdown.

Example A — Local highway blockade (low severity)

Input headline:

“Truckers block city ring road over toll hike; deliveries face minor delays”

Expected model JSON (abridged):


```
{
 "risk_types": ["Physical"],
 "scope": "Local",
 "impacted_nodes": [{"name":"City Ring Road","type":"road","country":"India","iso2":"IN"}],
 "countries_impacted": ["IN"],
 "economic_importance": "Low",
 "trade_dependency": "Low",
 "population_impact_bucket": "City",
 "impact_summary": "Local blockade on ring road delays last-mile deliveries; limited effect on intercity freight or ports.",
 "scores": {
  "scope_score": 3,
  "node_importance_score": 3,
  "risk_type_score": 6,
  "trade_dependency_score": 3,
  "population_impact_score": 3,
  "suggested_model_score": 3.9
 },
```

"confidence": 0.82,

"is_speculative": false,

"evidence_fields_used": ["title","description"]

}

Example B — Major port strike (high severity)

Input headline:

“Nationwide dockworkers strike shuts down Los Angeles/Long Beach terminals, vessels queue offshore”

Expected (abridged):

{

"risk_types": ["Infrastructure/Transport","Trade/Geopolitical"],

"scope": "Regional",

"impacted_nodes": [{"name":"Port of Los Angeles/Long Beach","type":"port","country":"United States","iso2":"US"}],

"countries_impacted": ["US"],

"economic_importance": "High",

"trade_dependency": "High",

"population_impact_bucket": "National",

"impact_summary": "Strike at major US container hub halts flows; import backlogs likely ripple across national retail and manufacturing.",

"scores": {

"scope_score": 6,

"node_importance_score": 9,

"risk_type_score": 7,

"trade_dependency_score": 9,

"population_impact_score": 7,

"suggested_model_score": 7.5

},

"confidence": 0.88,

"is_speculative": false,

"evidence_fields_used": ["title","description","content"]

}

Severity Calculation

1. Base SCSI (Supply Chain Severity Index)

Purpose: Create a **composite score** that combines multiple impact dimensions into a single metric.
Logic:

- Use weighted contributions from 5 key sub-scores that the LLM generates:
    - **Scope** → 25% weight (0.25)
    - **Node Importance** → 25% weight (0.25)
    - **Risk Type Severity** → 20% weight (0.20)
    - **Trade Dependency** → 20% weight (0.20)
    - **Population Impact** → 10% weight (0.10)
- Formula:

Base SCSI = 0.25*Scope + 0.25*NodeImportance + 0.20*RiskType + 0.20*TradeDependency + 0.10*PopulationImpact

Reason: Weighted scoring lets you balance factors by importance — scope & node importance get the most weight because they directly influence operational and economic impact.

2. Age Decay Multiplier

Purpose: Reduce the severity score as events get older.
Logic:

- News less relevant over time, so we decay the score based on days since published_at:
    - **≤ 2 days old** → full weight (1.0)
    - **3–7 days old** → −10% (0.9 multiplier)
    - **> 7 days old** → −20% (0.8 multiplier)
      Reason: Older disruptions are less likely to still be active or relevant for real-time decision-making.

3. Confidence Cap

Purpose: Avoid over-scoring when the LLM's classification confidence is low.
Logic:

- If confidence < 0.6, the score is **capped at 5**, regardless of the calculated value.
  Reason: A low confidence means the AI might be unsure about classification, so we treat it conservatively.

4. Speculative Cap

Purpose: Reduce the score for speculative or unconfirmed reports.
Logic:

- If is_speculative = TRUE, the score is **capped at 4**, regardless of prior adjustments.
  Reason: Prevents rumors or early unverified reports from skewing the rankings too high.

5. Spread Bonus

Purpose: Boost the score if an event impacts multiple countries.
Logic:

- If COUNTUNIQUE(countries_impacted for this event_id) ≥ 3, add **+0.5 points**.
- After adding, cap the final score at **10**.
  Reason: Disruptions affecting multiple countries usually have broader trade and economic implications.

6. Final SCSI

Purpose: Produce a final, audit-friendly severity score between **0 and 10**.
Logic:

- Apply all adjustments **in order**:
  1. Base SCSI from weighted sum
  2. Apply **Age Decay Multiplier**
  3. Apply **Confidence Cap**
  4. Apply **Speculative Cap**
  5. Apply **Spread Bonus**
  6. Apply final **cap at 10**
     Reason: This sequence ensures the score starts from a weighted base, is reduced for quality/age factors, and only then receives a small boost for broader spread — maintaining a logical, risk-sensitive flow.