Questions

Open Questions:

- Does the final solution must be deployed to the cloud or just a prototype?
    - Docker Container?
    - API that is deployed?

（ I am not sure but seems Docker should be sufficient in such short time preparation, or even a prototype, we could say then our product for cloud is launchable in 10 months, etc ）

- Should there be a video as a submission? A Power Point? (I would suggest that we prepare a pitch (fancy power points slides for 10-15mins?) and better have a demo with video, like Product Launch, the video will be super helpful to explain the details and highlights to the non-tech people
- How do we have access to data? Are we given API keys? Public APIs?
- Any Latency Requirements? => Efficiency?
- Error Handling? Stable?
- Is the idea or the solution more important? => **Evaluation Criteria? (Several comes to my mind are: innovative, launchable, user-friendly (especially for non-tech users), accuracy and time efficiency, which latest tech did we apply?)**
- Do we have access to last year’s submissions?
    - Repos? Videos? …
- Should this work for a larger user group?
- Who is the end customer? Who will consume the dashboards? (This is a good question, we can also say, compared to the traditional or current products, why our product is much better, depends on the users/customers, we need to focus on their pain points)

BrainStorming

Ideas:
- I recommend Agentic RAG (reasoning) AI
- Multimodal Models : Multimodal Transformers, GeoTransformer / ST-Transformer, Perceiver IO.
- **Intermediate / Cross-Attention Fusion:** Each modality informs the other via attention layers—common in transformers like Perceiver IO.
- tools:
    - weather forecast
    - crop season in different parts of the world
        - soil moisture, vegetation health, best crop growth depending on weather conditions
    - Global Food index (differentiating nations on the basis of crop production, their availability and needs)
    - Index on the basis of shared crops
    - platform to request and register the available/required crops
    - make it modular depending on choice of language
- ***we missed three points here:***
- Supply Chain Disruption Prediction Module
- Retail Inventory & Demand Optimization Tool
- Climate Transparency & Traceability Layer

Single Platform
- dummy company with X food sources:
    -
    - Certain events how they affect food volumes
- Check if food sources are in danger ->
- Forecasting -> Volumes

Users:

1. Large Food Retailers & Distributors
- Problem: Supply disruptions and price volatility due to climate extremes.
- Need: Early warnings (prices to high because of drought), alternative sourcing suggestions, risk scoring.
- Value: Secure supply, reduce costs, avoid empty shelves

APIs:
    - meteomatics
- Nasa dataset for food sources

Database:

Table: Suppliers

- SupplierId (One example founded, fenaco)
- Name (fenaco)
- Available Crop Types **[Wheet, Potatos, Rice] (Done)**
- Volumes (fenaco)
- Location (fenaco)
- price
- expiry date

Table: Company

- CompanyId (Landi)
- Location (Landi)
- Name (Landi)
- budget (transportation)
- Needs:
  - 40kg rice
  - 20kg wheet

Table: CompanySupplierMapping

- CompanyId
- SupplierId
- Volume
- CropType [Rice]
- Truck, Train

fastapi
dash / plotly (map -> leaflet)

- frontend

- Setup frontend backend structure basic communication
- Write sid a summary

- look for data sources
- github
- pitch / presentation

- web app:
  - User Management (1 account for 1 company)
  - user should be able to select suppliers and the crop type and volume
  - For each selected Supplier => Monitor Risk (every 2 days)
      - Regression Model => Risk index
    - If risk was found (threshold) => show in the dashboard and do something

    - Recommendation on what to do
      - Available suppliers with enough volume

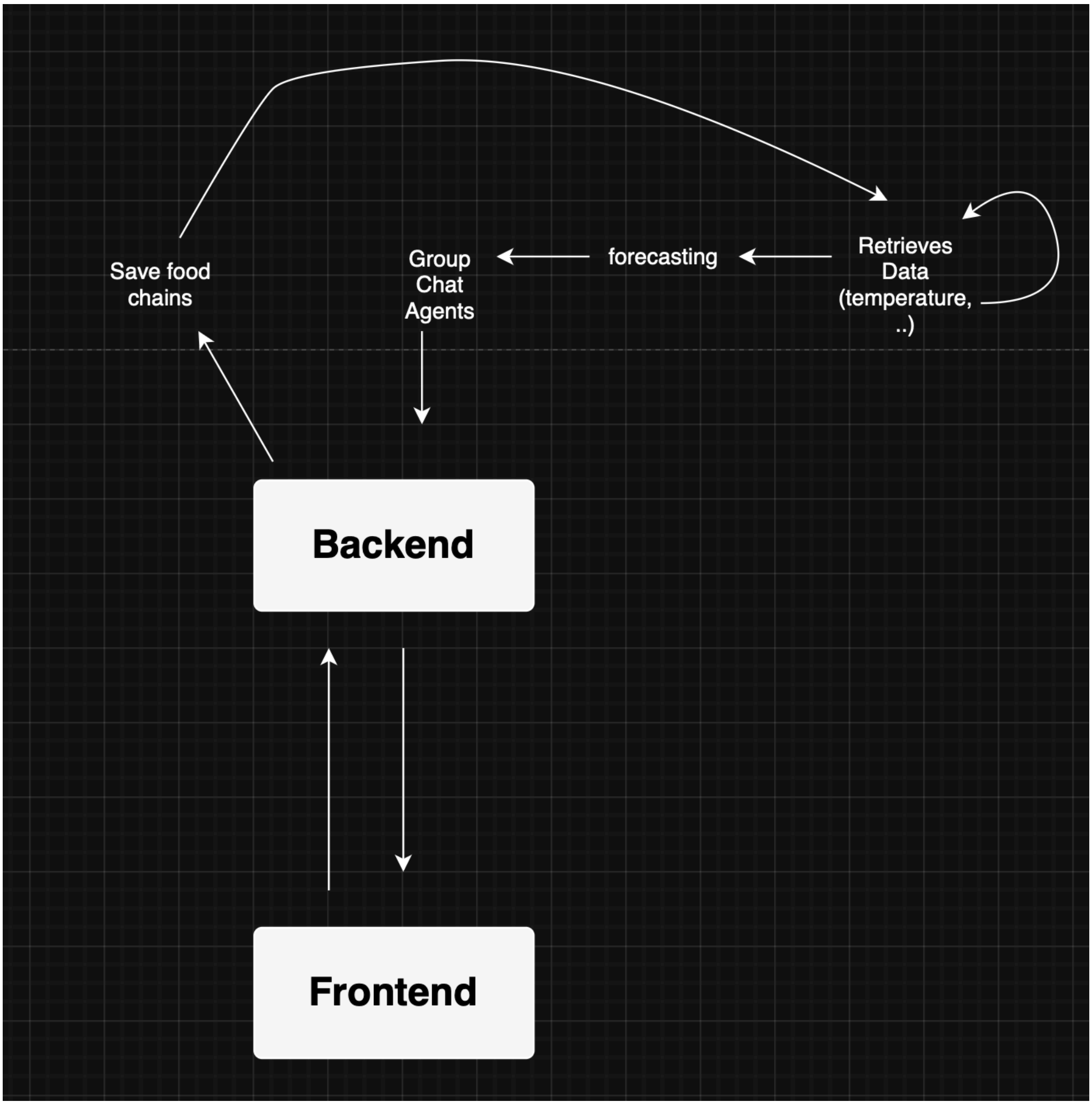
        - Product is not available
        - langgraph

  - Food Waste (rule based / Decision Tree) => explain why
    - store when food was harvest => food goes bad after nth days
    - if too much food left
      - lower price
      - show flag in frontend
      - include those suppliers more often in recommendations

https://food.opendata.ch/#fooddata

https://nsidc.org/data/nsidc-0712/versions/1

https://nsidc.org/data/data-access-tool/SPL1BTB_NRT/versions/105



Idea: The approach is to forecast multiple critical factors for each supplier location (temperature, rainfall, soil moisture, vegetation health, etc.) and combine them into a single risk score for each product. And output an explanation why the crops are at risk.

Execute flow every x-th hours?

- How long is the forecast? a few days? weeks? months?（ can we use users to choose the time period based on their needs, they can select days/weeks/months by their needs ）

Agent Layers & Responsibilities

Layer 1: Environmental Forecast

- **Temperature Agent** → predicts temperature at supplier location.
- **Soil Moisture Agent** → forecasts soil moisture.
- **Rainfall Agent** → forecasts precipitation.
- **Vegetation Agent** → tracks NDVI/EVI trends.

  Output: forecasts + initial factor-level stress signal.

- Maybe not agents but **GNN / ConvLSTM / TSFMs**? Other algorithms?
- Maybe to many different models for the short time?
  - Concentrate on temp and moisture?
- Numerical methods?
- Forecasting APIs?

Layer 2: Expert Agents

- One expert agent per crop (Potatoes, Tomatoes, Coffee, etc.).
- Or one agent for each forecast type (temperature, soil, …)
- Receives forecasts.
- Evaluates **crop-specific stress**:
  - Optimal conditions vs forecast.
  - Outputs **risk signal** per crop per supplier.
  - *“Potatoes in Spain in June should be around 20°C; forecast shows 26°C → stress risk high.”*

Layer 3: Risk Aggregator Agent

- Collects all crop risk signals at a supplier location.
- Computes **final risk score** (Low / Medium / High).
- Generates **explanation**:
  “Potatoes at Supplier X: High risk due to 26°C vs optimal 20°C and soil moisture 30% below normal.”
- structured output: {‘risk_score’: 0.6, ‘explanation’: ‘...’}

Layer 4: Alternatives Suggestion Agent

- Activated **only if final risk > threshold**.
- Inputs:
  1. Supplier location(s) at risk
  2. Crop type
  3. Current portfolio and volumes
  4. Nearby or global alternative suppliers (database)
- Function:
  1. Searches for alternative suppliers or regions where forecasts show **low risk**.
  2. Evaluates **capacity, distance, cost, and environmental impact**.
  3. Returns **ranked alternatives**.

- Output example:
  "Potatoes at Supplier X are high risk. Alternatives:
    1. Supplier Y (Thurgau, Switzerland) – Low risk, 80% volume coverage
    2. Supplier Z (Tyrol, Austria) – Medium risk, 60% coverage"

Dashboard:

- Global Supply Chain Map (Core Layer):
  - **Nodes**: main suppliers for company (e.g., coffee → Brazil, wheat → Ukraine, tomatoes → Spain).
  - **Edges**: trade routes → thickness = volume.
  - **Overlay options in map:**
    - Temperature
    - Precipitation
    - Soil moisture (SMAP)
    - Vegetation health (NDVI/EVI)
    - Climate risk zones (drought, flood, heatwave)
  - Tooltip on click: "Spain, Tomatoes – NDVI -12% below average → Medium risk."

- Food Portfolio Breakdown (Treemap or Donut Chart)
  Share of different food categories in Company supply:
  - 30% Wheat, 20% Fruits, 15% Vegetables, 10% Coffee, etc.
  - Drilldown: click "Fruits" → shows origin countries + risk indicators.

- Food Resilience Index
  - Heatmap or bar chart: ranks food items by climate risk (e.g., Coffee = high, Potatoes = low).
  - Traffic light system: green = stable, yellow = watch, red = high risk.
  - Based on climate anomalies, crop yields, and forecasts.

- Risk Timeline (Forecast Chart)
  - Forward-looking chart (next 6–12 months).
  - Shows how risks for key products evolve over time.
  - Example: "Wheat – risk increasing in summer due to drought forecast in Eastern Europe."

- Alert Cards
  - Small cards with the current top risks:
  - "Coffee beans: drought in Brazil – +40% price volatility possible."
  - "Spanish tomatoes: 2 weeks below-average soil moisture."

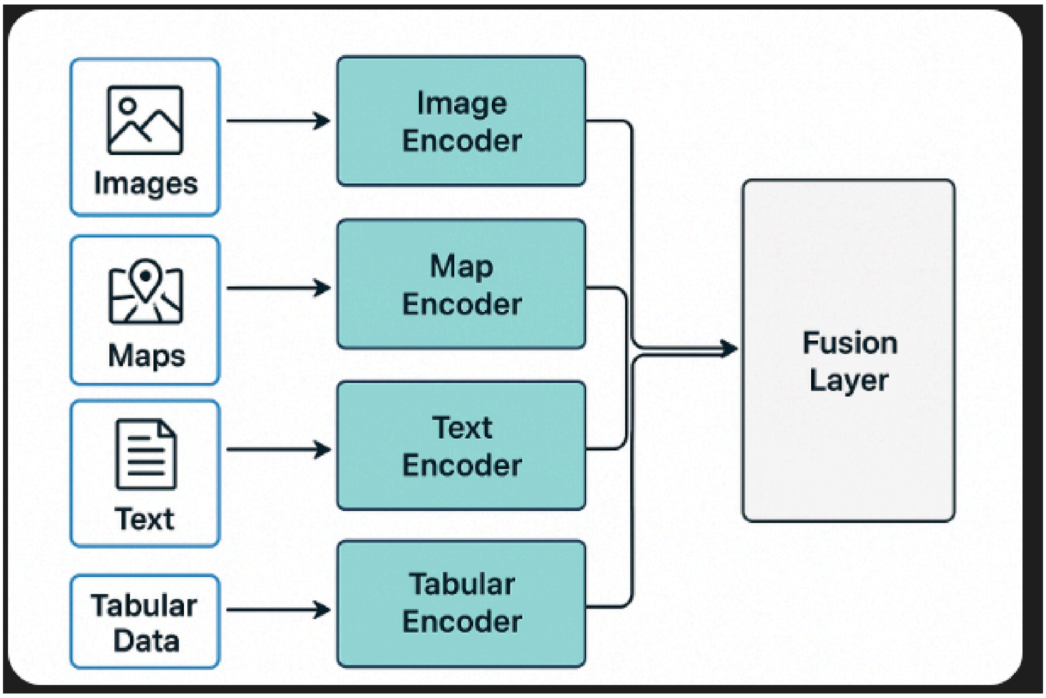
- Click → zooms into the affected area on the map.

Interactive Form Builder

- User clicks "+ Add Supply Chain".
- Selects product → picks country/region
- Inputs approximate volume/% share.
- Save → adds to the map + risk analysis.
-

DataFlow

- User query ⇒ Langgraph ⇒data sources ⇒ agents ⇒ tools ⇒ prompts ⇒ dashboards ⇒ responses

- fusion flow for different data types:
-


-

Data Sources

Vegetation health:
https://modis.gsfc.nasa.gov/data/dataprod/mod13.php

Soil Moisture:
https://smap.jpl.nasa.gov/

precipitation
https://gpm.nasa.gov/data/imerg

Historical crop suitability maps:
https://gaez.fao.org/

Soil:
https://soilgrids.org/

UI Ideas

```
+------------------------------------------------------------------------------+
| 🌾 Risk Dashboard – Zurich Company                                    |
+------------------------------------------------------------------------------+
| Top Bar: [Company Logo]  |  [Select Product: Coffee ▼]  [Time Range: 6m ▼]    |
|                          |  [Alerts 🔔 3]   [⬆️ Import Supply Data] [User ▼]  |
+------------------------------------------------------------------------------+

| LEFT PANEL (Navigation)              | MAIN PANEL (Map & Insights)         |
|--------------------------------------|-------------------------------------|
| 📍 Supply Chain Map                  | 🗺️ Interactive Global Map            |
| 📊 Product Portfolio Breakdown        |  - Supplier nodes (Brazil, Spain…)  |
| 📈 Risk Forecasts                    |  - Trade flow lines to Zurich        |
| 🚨 Alerts & Notifications             |  - Overlay toggles:                  |
| 🧭 Alternatives & CO2 Impact          |     [☑ Rainfall] [☑ Soil Moisture]  |
| 🍽️ Food Waste Monitor                |     [☐ Temperature] [☑ Vegetation]  |
|--------------------------------------|-------------------------------------|
| ➕ Add Supply Chain                   | 🔵 “Click Map to Add Supplier” Mode  |
+--------------------------------------|-------------------------------------+

| BOTTOM PANEL (Data Visualizations)                                   |
|----------------------------------------------------------------------------|
| 📊 Donut Chart: Supply Share per Product                              |
|    - Wheat 30% | Fruits 20% | Vegetables 15% | Coffee 10% …                 |
|                                                        |
| 📈 Food Resilience Index (Bar Chart)                                  |
|    - Coffee: HIGH Risk 🔴  | Wheat: MED Risk 🟡  | Potatoes: LOW Risk 🟢        |
|                                                        |
| 🕒 Risk Timeline (Forecast Chart)                                     |
|    - Coffee: ↑ risk in next 2 months due to Brazil drought                 |
|    - Tomatoes: stable → seasonal recovery                                  |
+------------------------------------------------------------------------------+

| ALERT CARDS (Pinned to Right Side)                                     |
|----------------------------------------------------------------------------|
| 🚨 Coffee beans (Brazil): Drought detected – harvest down 15% forecast        |
| ⚠️ Tomatoes (Spain): 2 weeks below-average soil moisture                     |
| ✅ Potatoes (Switzerland): Stable supply expected                           |
+------------------------------------------------------------------------------+
```

FinalIdea

DataSet

1. **Crop & Vegetation Health: MODIS NDVI & EVI Vegetation Index [10tb]**
   a. Not 10TB already reduced to switzerland
   b. https://search.earthdata.nasa.gov/search/granules?p=C1748066515-LPCLOUD&pg%5B0%5D%5Bv%5D=f&pg%5B0%5D%5Bgsk%5D=-start_date&g=G3786218192-LPCLOUD&polygon%5B0%5D=6.1136%2C47.07578%2C5.71233%2C46.34442%2C6.01919%2C45.8136%2C7.40003%2C45.57768%2C9.81946%2C45.70744%2C11.22391%2C46.74549%2C10.27974%2C47.65378%2C8.40321%2C48.07844%2C6.70371%2C47.6066%2C6.1136%2C47.07578&tl=1355176119.259!5!!&lat=45.039393498799996&long=8.121159668989547&zoom=5.6950348392290016
   c. Alternative Google Earth Engine [add link repo/data access- chu] https://developers.google.com/earth-engine/datasets?hl=zh-cn
2. Meteo link
   a. Open Meteo [link - chu] https://open-meteo.com/
   b. MeteoSwiss Climate Data → meteoswiss.admin.ch
3. Govt data/UN data [survey/data] -> chu https://www.fao.org/faostat/en/#home

TL;DR Summary

Use space / satellite / Earth observation data (and allied analytics) to help make agricultural supply chains more climate-resilient, efficient, and food-secure — e.g. by improving forecasting, reducing waste, optimizing logistics

Our solution will support the actors in the chain (farmers, distributors and/or food processing companies, and end consumers) to make better decisions under climate uncertainty, optimize yield, reduce losses, and make the chain more “smart” and sustainable.

Challenge #5

Key Stakeholders

Stakeholder	Needs / Pain Points	Potential Value from a Solution
Farmers	Climate variability (drought, flood, temperature stress), pest/disease, yield uncertainty, inefficient	Early warning, field-level forecasting, recommendations (when to plant, irrigate, inputs), yield prediction, risk assessment

	resource use (water, fertilizer), crop failure risk	
Distributors / Processors / Companies	Supply variability, spoilage / loss, unpredictable yields, logistical inefficiencies, inventory planning, demand forecasting, procurement risk	Better upstream visibility, demand-driven procurement, optimization of transport and storage, reduced waste, traceability, carbon footprint tracking
End Consumers / Society	Food affordability, food security, environmental impact (GHG, water use), supply stability	Lower risk of food shortages, more sustainably produced food, stable prices, transparency

In addition: regulatory bodies, NGOs, finance / insurers, extension services may also be users / collaborators.

One crop + one pilot region (e.g., wheat in a chosen district).

Deliver a web app that:

- *shows field-level crop/vigor & risk maps from NASA/USGS data,*
- *predicts short-term supply risk (low/med/high) for buyers,*
- *suggests actions for farmers (irrigation/harvest windows),*
- *exposes a simple API so distributors can pull the risk signal.*

Core datasets (ready, open, fast to use)

- *GPM IMERG precipitation (near real-time, 30-min/daily) for rainfall deficits/surplus.*
- *SMAP soil moisture & freeze/thaw (daily composites; root-zone via L4) for water stress.*
- *MODIS NDVI/EVI vegetation indices (16-day, 250m) for canopy greenness trend.*
- *(Optional higher-res tiles for demo visuals) Landsat 8/9 Surface Reflectance.*

These are widely used for agri decisions (see NASA Applied Sciences + NASA Harvest context).

Key Functional requirements:

Farmer view

- *Map + time-series for NDVI, rainfall, soil moisture per field.*
- *Risk tile (Today / Next 7 days): Water stress, flood risk, vigor anomaly.*
- *Action cards (plain language): "Delay irrigation 2–3 days," "Harvest window Thu-Sat."*

Distributor/Company view

- *Supply risk by zone (low/med/high) for next 2–3 weeks.*
- *Procurement insights: where to source more/less; simple CSV/API export.*

System

- *1-click region setup (draw polygon or upload GeoJSON).*
- *API to fetch: /risk?polygon_id=...&date=...*
- *Attribution + dataset info panel (which satellite layers & timestamps).*

Tech design

Stack (fast to prototype)

- *Backend: Python + FastAPI; cron/worker for tile pulls & feature aggregation.*
- *Geospatial: rasterio, geopandas, xarray (or use Google Earth Engine for speed and pre-hosted datasets; optional).*
- *Modeling: lightgbm or catboost for tabular risk scoring; prophet or statsmodels for simple rainfall trend anomaly.*
- *Frontend: React + MapLibre (or Leaflet) with vector tiles & a minimal dashboard.*
- *Storage: PostGIS (polygons/metadata), object store for tiles (e.g., Cloud bucket).*

Micro-services

- *ingestor: pulls GPM/SMAP/MODIS for AOI & time window, writes GeoTIFF/NetCDF.*
- *feature-maker: computes daily features per grid/field; caches parquet.*
- *risk-api: scores & serves results; simple JWT auth.*
- *ui: web app.*

4) Data flow (EO → features → risk → UI)

```
+----------------+     +----------------+     +------------------+     +--------------------
+
| NASA/USGS APIs | --> | Ingestor (ETL) | --> | Feature Store    | --> | Risk Scorer (ML)  |
|  GPM, SMAP,    |     |  (pull AOI     |     |  per field/day:  |     |  (tabular model + |
|  MODIS, Landsat|     |   & timeframe) |     |  NDVI, rain, SM) |     |   rules/anomalies)|

+----------------+     +----------------+     +------------------+     +----------+---------
+
                                                                                  |
                                                                                  v
                                                                         +------------------+
                                                                         | Risk API (JSON) |
                                                                         +--------+--------+
                                                                                  |
                                  +----------------------+                      v
                                  | Frontend (Map + UI) | <-----------  GeoJSON/tiles
                                  +----------------------+       (zones, fields, scores)
```

Feature examples (daily or 8/16-day, per field/zone)

- *NDVI latest, ΔNDVI vs 3-period median (stress).*
- *Cumulative rainfall vs climatology (SPI-like simple z-score).*
- *SMAP surface/root-zone moisture percentile.*
- *Growing degree days (basic).*

(IMERG = rainfall, SMAP = moisture, MODIS = vegetation—ideal minimal trio).

5) Request flow diagram (user → inference → data)

```
User (Farmer/Buyer)
        |
        | 1. Select AOI/field/date
        v
Frontend ------------- 2. /risk?polygon_id&date ----------> Risk API
        |                                                  (LightGBM score + rules)
        |                             3. fetch features ----^
        |                             (from Feature Store)
        |                                                   4. lazy ingest if cache miss
        |                                          (Ingestor pulls GPM/SMAP/MODIS)
        v
Map + Cards  <--------- 5. GeoJSON + tiles + recommendations (LLM)
(visualization)
```

GenAI integrations

1. *Explainability + Actions*
   An LLM turns raw indicators into plain-language recommendations ("Your field's NDVI dropped 9% while soil moisture is at p20: likely heat stress; consider early-morning irrigation; avoid fertilizer until rainfall resumes.").
   - *Prompt = {last 4 timesteps of features + crop + calendar stage}.*
   - *Guardrails: don't over-claim; include confidence/"data freshness."*
2.
3. *Natural-language query*
   "Where should I prioritize procurement next week?" → converts to API queries + map filters; returns a short summary + links.
4. *What-if generator (lightweight)*
   "If rainfall stays below 10 mm for 7 days, how does risk change?" → recompute with altered rain inputs (no retraining).

We anchor all GenAI outputs on the cited satellite features (NASA/USGS datasets).

ML model options

Goal: 2-class or 3-class Supply Risk (Low / Medium / High) at zone/field granularity.

Option A (recommended for 48h): LightGBM/CatBoost on tabular features

- *Inputs: last N (e.g., 4) timesteps of NDVI, ΔNDVI, rainfall anomaly, moisture percentile, GDD, month.*
- *Labels: weak labels (heuristics) for demo (e.g., "High risk if NDVI↓ & rainfall anomaly<−1 & SMAP<30p"), or tiny curated examples.*
- *Pros: trains in minutes; strong baselines; easy feature importance.*
- *Cons: demo labels aren't ground truth (note as limitation).*

Option B (no training): Rule-based scoring + calibration

- *Compute a weighted score from normalized indicators; map to L/M/H.*
- *Pros: zero training time; transparent.*
- *Cons: cruder; but great for hackathon MVP.*

Option C (if you already have labels): Temporal models

- *1D CNN/LSTM on short multivariate sequences (NDVI/rain/SM).*
- *Pros: captures dynamics. Cons: longer to wrangle; risky in 48h.*

Side models (nice-to-have)

- *Prophet/ARIMA for short rainfall/NDVI trend extrapolation.*
- *Isotonic regression on top of A/B for better probability calibration.*
-

48 hour execution plan

Day 1 (Build the spine)

- *Define AOI polygons; seed with demo fields.*
- *Implement Ingestor for IMERG daily, SMAP L3/L4, MODIS NDVI (last 90 days) for AOI.*
- *Build feature-maker & parquet cache; compute simple anomalies/percentiles.*
- *Implement Option B rule-based risk (then, if time, swap to LightGBM).*
- *Stand up Risk API and Leaflet/MapLibre UI with GeoJSON layer.*

Day 2 (Polish + GenAI + demo)

- *Add LLM explainer + action cards (RAG prompt with feature JSON).*
- *Add download/export (CSV + simple procurement summary).*
- *Add tiles and toggles for NDVI/rain/moisture layers; legends & timestamps.*
- *(Stretch) Train small LightGBM with weak labels; compare to rules.*
- *Write README + dataset attribution citing IMERG/SMAP/MODIS/Landsat & NASA programs.*

Demo

1. *Draw AOI → app pulls last 60–90 days of IMERG/SMAP/MODIS.*
2. *Toggle layers (NDVI trend, rainfall anomaly, soil moisture pct).*
3. *Click a field: see risk = HIGH + LLM recommendation.*
4. *Switch to buyer view: zone heatmap + CSV export for sourcing.*
5. *"What-if": reduce next-week rainfall → risk increases → procurement shifts.*

--- *for more details you can refer:----*

Phase 1: Data & Pipeline Foundation

1. *Data sourcing & access*
   - *Ingest satellite imagery (e.g. NDVI, surface reflectance, spectral bands)*
   - *Acquire historical weather data, climate projections, soil maps, topography, land use, etc.*
   - *Partner with local farms / agencies to get yield / sensor / on-ground data*
2. *Data preprocessing & harmonization*
   - *Radiometric / atmospheric corrections, alignment, mosaicking*
   - *Resampling / projection harmonization*
   - *Gap filling / cloud masking*
   - *Feature engineering (vegetation indices, soil moisture proxies, stress indices)*
3. *Data storage & pipeline engineering*
   - *Choose data storage (data lake, geospatial database)*
   - *Build ETL pipelines (ingest → clean → feature store)*
   - *Versioning, provenance tracking*
4. *Exploratory data analysis (EDA)*
   - *Understand correlations, trends, anomalies*
   - *Visualization – maps, time series, spatial plots*
5. *Ground truth alignment & labeling*

- *Align yield or field-level outcome data to image/time windows*
- *Partition datasets for training / validation / test*

Phase 2: Modeling & Forecasting

1. *Select modeling approaches*
    a. *E.g. machine learning (random forest, gradient boosting, neural nets), time series forecasting (LSTM, ARIMA), geospatial models*
    b. *Potential hybrid models (satellite + weather + soil + management practices)*
2. *Train yield / stress forecasting models*
    a. *Predict per-field yields, stress events (drought, heat), disease / pest risk, water stress*
    b. *Forecast horizons (e.g. seasonal, weekly)*
3. *Supply chain decision models*
    a. *Demand forecasting (consumer demand, seasonal patterns)*
    b. *Optimization models (how much to procure from which region, logistics routing, buffer stocks)*
    c. *Risk models (scenario analysis under climate extremes)*
    d.
4. *Validation & evaluation*
    a. *Use test sets, cross-validation*
    b. *Spatial / temporal generalization tests*
    c. *Compare against baselines*
    d. *Sensitivity / ablation analyses*
5. *Iterate & tune*
    a. *Feature selection, hyperparameter tuning*
    b. *Incorporate domain knowledge / constraints*
    c. *Debug edge cases / failure modes*

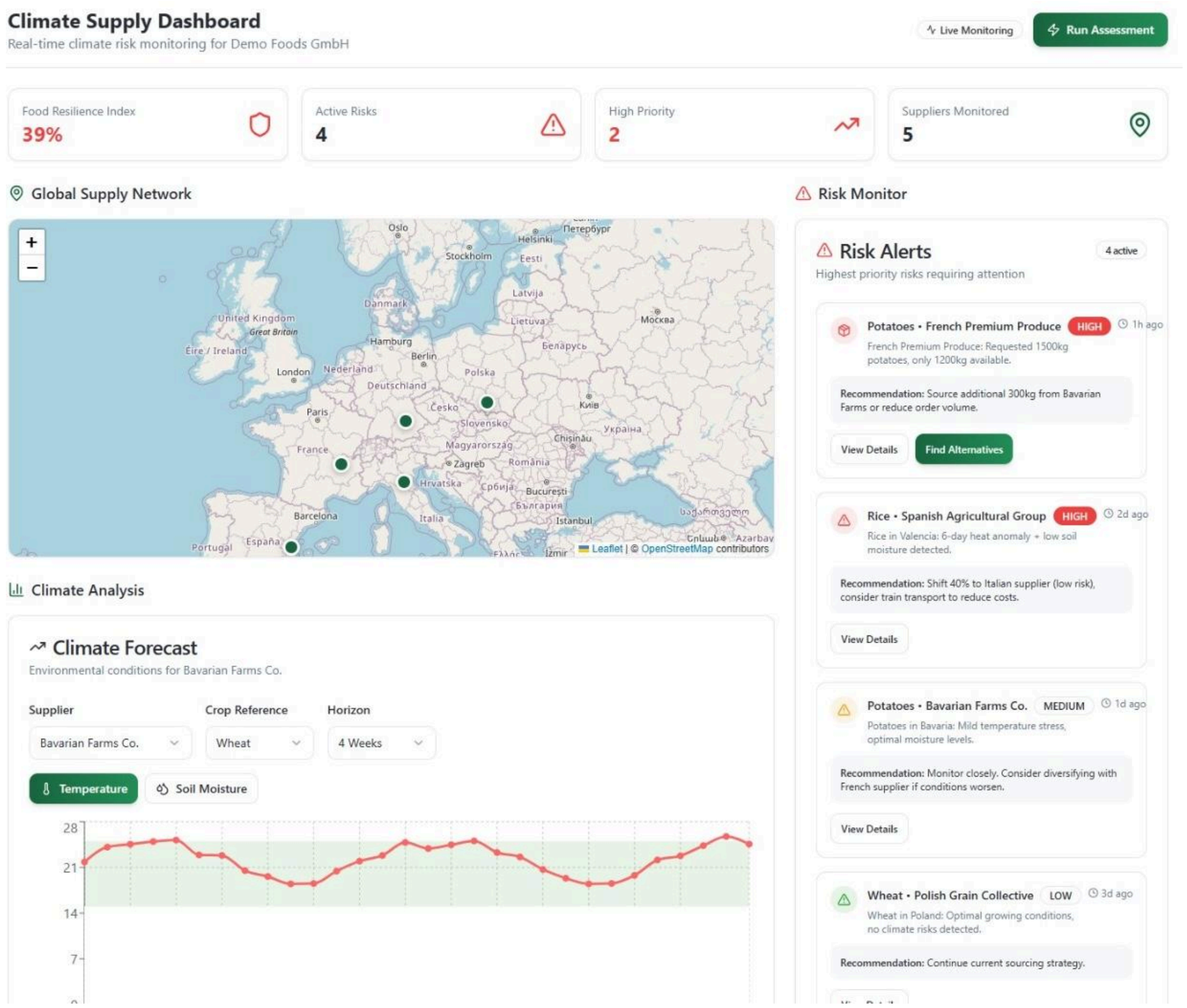
Phase 3: Modeling & Forecasting + Agentic AI

1. *Backend / API development*
    a. *Serve model predictions (e.g. via REST or GraphQL APIs)*
    b. *Data query endpoints (spatial queries, time window queries)*

    c. *Manage user data, authentication, roles (farmer vs distributor)*
    d.
2. *Frontend / dashboard / visualization*
    a. *Map-based UI for farmers (view predicted stress maps, recommended actions)*
    b. *Dashboard for distributors (supply projections by region, risk zones)*
    c. *Scenario / "what-if" module (simulate effects of climate anomalies, shifts)*
3. *Alerts / notification modules*
    a. *Trigger warnings (e.g. anticipated drought)*
    b. *Recommendations (e.g. delay planting, increase irrigation)*
    c. *Interface to send alerts (SMS, email, mobile push)*
4. *Feedback / user input loop*
    a. *Allow users to input what they actually did / results*
    b. *Use that feedback for continuous improvement*
    c.
5. *System testing & QA*
    a. *Integration testing, performance testing, user acceptance*

== STOP READING HERE==

Example Dashboard (screenshot generated by AI)



Dashboard Concept (with Food Waste Priority)

Company Setup

- Each company creates a **profile** with its requirements:
    - Crop types (e.g., wheat, rice, potatoes)
    - Required volumes
    - Preferred transport modes (truck, train, etc.)
    - Budget limits

Supplier Data

- Suppliers are already registered in the system with their stock data:
    - Crop type, price, location
    - Initial & remaining volumes
    - Harvest date and expiry date
    - Storage conditions

(Since real-world data may not be available, **dummy stock data** will probably be used for simulation.)

Matching & Selection

- Companies select which suppliers should cover their defined needs.
-

Monitoring

Once mappings are created, the system begins **continuous monitoring**:

1. **Climate Risk (Regression Model):**
    - Supplier regions checked periodically (temperature, soil moisture, crop sensitivity).
    - If riskthresholds are exceeded → an **alert** is triggered.
        i. Retrieving climate forecast for the suppliers region via API
        ii. Regression Model for risk score
2. **Food Waste Risk (Supportive):**
    - System tracks suppliers with large remaining volumes close to expiry.
    - This information is **used in recommendations**: suppliers at higher food-waste risk are prioritized when alternatives are needed.

Multi-Agent System (LangGraph)

When a supplier is at **climate risk** -> Agents evaluate alternatives:

- **Finance Agent** → Budget check
- **Supply Stability Agent** → Reliability check
- **Logistics Agent** → Transport & distance check
- **Waste Agent** → Promotes suppliers with large stock close to expiry => cheaper and reduce food waste.

Agents discuss with each other to come up with e.g. top 3 alternatives with explanations (advantages / disadvantages for each alternative).

Food Waste Integration:
If Supplier A has stock nearing expiry, and Supplier B is flagged as risky, the Waste Agent argues to prioritize Supplier A in the recommendation list.

Dashboard Features

- **Map View:** All suppliers shown with traffic-light indicators (green/yellow/red for risk level).
- **Risk Alerts:** Only triggered for climate/volume risks.
- **Recommendations Section:**
  - Shows alternative suppliers if one becomes risky.
  - Suppliers with food waste risk are ranked higher (prioritized).
  - Each suggestion includes reasoning from the agents.
- **Forecast Panel:** Climate (and stock) trend graphs for selected suppliers.

[missing flows]

Frontend (Dash/Plotly, Leaflet):

- Company sets up needs and selects suppliers.
- Dashboard displays risk alerts, map view, and ranked recommendations.

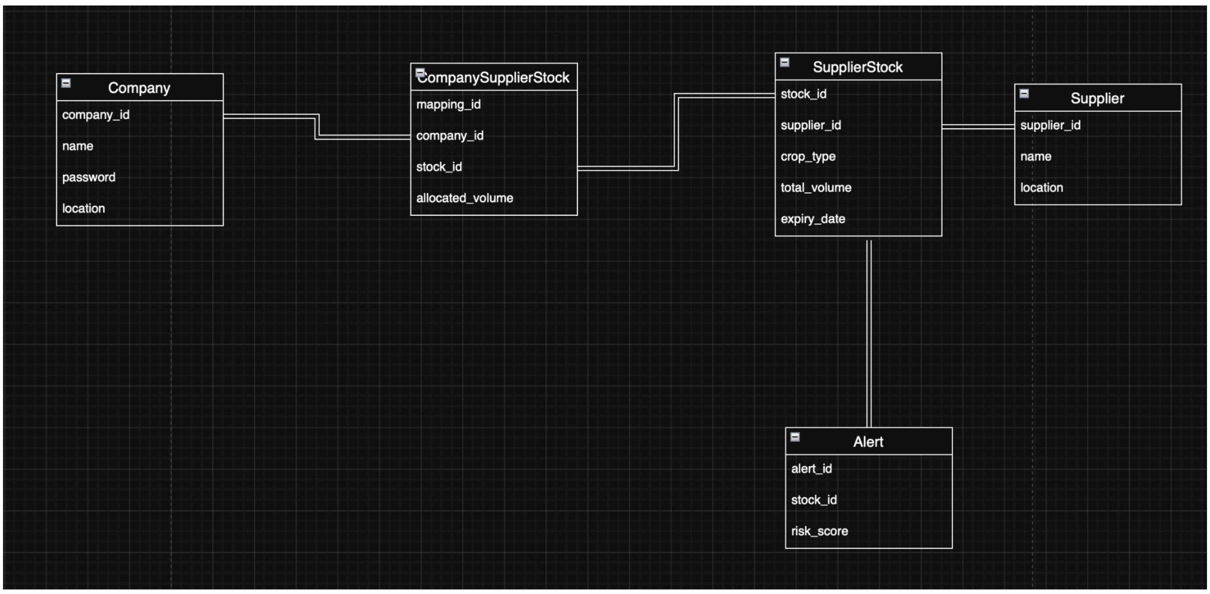
Backend (FastAPI):

- Runs regression models for climate risk.
- Identifies food waste risk (remaining volume + expiry date).
- Triggers multi-agent reasoning in LangGraph.

Database Layer (SQLite)

We'll use **SQLite** for simplicity (lightweight, file-based, portable). This allows rapid prototyping and keeps the schema explicit and relational.

Schema:



Query Patterns:

-
- Check supplier region climate risks (alerts stored in **Alerts**)
- Identify suppliers with stock nearing expiry (alerts + prioritization for recommendations)
- Retrieve most recent alerts and recommendations per company/supplier

Notes:

- SQLite can handle this well for a **prototype or small-scale deployment**.
- If scaling to larger, multi-user production → migrate schema to RDS (MySQL/Postgres).

UpdatedPlan

Prio

- SQLite database with dummy data
    - Pre-store companies, suppliers, and stock for demo.
    - Tables for Alerts & Recommendations

- Cloud Access
    - Container Apps
    - LLM Endpoint
    - DB

- Climate risk detection
    - Calculates risk scores for supplier region (get supplier location, crop types, … from DB).
    - Get Data from Forecasting API, …
    - Train Model
    - Save Model (weights)
    - Check how to get the data in the future

- Backend
    - Serve endpoints to get data from DB (Suppliers, Alerts, …)
    - Periodically iterate over each supplier location to calculate risk score
    - If risk score above threshold => trigger Agent
    - Save risk score always into db. With recommendations if above thresold

- Multi-agent/Single Agent rule logic
    - Finance, Logistics, Waste agents rank alternatives + reasoning
    - Structured Output

- Dashboard
    - Map view -> Shows supplier locations (location from DB)
    - Dashboard: Alerts panel Lists active climate & waste alerts (based on risk score in different colors / traffic light) -> get alerts from DB
    - Recommendations panel -> Displays alternative suppliers with reasoning. Get Recommendation from DB

- Scenario simulation:
    - Simulate supplier outage impacts.

Project Summary

Project Summary:

TerraTrace: Climate-Resilient Agricultural Supply Chain Intelligence

Objective:
Leverage satellite and Earth observation data, allied analytics, and AI-driven recommendations to make agricultural supply chains **climate-resilient, efficient, and food-secure**. The solution empowers farmers, distributors, food processors, and consumers to make better decisions under climate uncertainty, optimize yield, reduce losses, and enhance sustainability.

Scope & Pilot:

- **Pilot Crop & Region:** Focus on one crop (e.g., wheat) in a chosen district.
- **Prototype Deployment:** Web application covering field-level monitoring, risk assessment, and supply chain recommendations.
- **Stakeholders:**
  - **Farmers:** Receive early warnings, field-level forecasts, and actionable recommendations (planting, irrigation, harvesting).
  - **Distributors/Processors:** Access supply risk scoring, demand-driven procurement insights, and optimized logistics guidance.
  - **End Consumers & Society:** Gain from more sustainable, stable, and transparent food supply chains.

Core Capabilities:

1. **Satellite-Based Analytics:**
   - NASA/USGS datasets: GPM IMERG (precipitation), SMAP (soil moisture), MODIS NDVI/EVI (vegetation health).
   - Optional Landsat 8/9 imagery for high-resolution demo visuals.

2. **Risk Scoring & Recommendations:**
   - Field-level climate and crop stress assessment (water stress, flood risk, vigor anomaly).
   - Short-term supply risk (low/medium/high) for buyers.

    - Actionable guidance via natural-language recommendations powered by GenAI (LLM) anchored on satellite data.

3. **Web Application & API:**
    - Interactive map dashboards for farmers and distributors.
    - Action cards and alerts for decision-making.
    - Exportable insights (CSV/API) for procurement planning.

4. **Machine Learning & Rules Engine:**
    - Rule-based scoring for rapid prototyping.
    - Optionally, lightweight LightGBM/CatBoost models for predictive risk scoring using NDVI, rainfall anomaly, soil moisture percentile, and growing degree days.
    - What-if scenario simulations for climate-driven supply disruptions.

5. **Tech Stack:**
    - Backend: Python + FastAPI; PostGIS for spatial data; cron/worker pipelines for feature extraction.
    - Frontend: React + MapLibre/Leaflet; dashboards and field/zone maps.
    - Data Management: Raster and vector storage (GeoTIFF, Parquet), feature caching, and API endpoints.

6. **Multi-Agent System (LangGraph):**
    - Agents evaluate climate, logistics, finance, and food waste risk to provide ranked supplier recommendations.
    - Integrates supplier stock, expiry dates, and environmental risk into actionable decisions.

Expected Outcomes:

- Reduced crop loss and improved supply stability.
- Optimized procurement and reduced food waste through intelligent recommendation prioritization.
- Enhanced climate resilience and sustainability across the supply chain.

- Prototype framework ready for scaling across multiple crops, regions, and stakeholders.

Deliverables:

- Web application dashboard with map layers, risk alerts, and actionable insights.
- REST API for fetching per-field/zone supply risk and recommendations.
- Demo-ready satellite-based monitoring and predictive analytics pipeline.
- Documentation of datasets, ML models, and LLM explainability features.

Innovation & Impact:

- Combines **remote sensing, ML, and LLM-powered actionable insights** for supply chain decision-making.
- Promotes **sustainable agriculture** by aligning procurement, crop management, and consumer supply with climate-informed intelligence.
- Enables **smart, data-driven, and explainable decision-making** across all actors in the agricultural value chain.

9. During the hackathon, our team faced several key challenges that tested both our technical and collaborative skills. One major hurdle was integrating diverse data sources satellite imagery, climate metrics, and logistics data into a single, seamless platform. Ensuring data accuracy and consistency across regions required creative preprocessing and validation methods. We also encountered difficulties optimizing AI models to deliver timely, interpretable insights rather than overwhelming users with complex outputs.

Collaboration under time pressure taught us the importance of clear communication, rapid prototyping, and iterative testing. We learned to balance innovation with practicality focusing on creating a solution that is not only technically sound but also user-friendly and scalable. From a broader perspective, the hackathon reinforced how cross-disciplinary teamwork combining AI, climate science, and design can lead to meaningful solutions that address global challenges like food security and climate resilience.

8. Our solution uniquely integrates real-time satellite data with AI-driven predictive analytics to monitor rainfall, soil moisture, and vegetation health at the field level. It employs GenAI to generate clear, actionable decision cards, enabling users to respond proactively rather than reactively. By combining climate intelligence with supply chain risk modeling, the platform

delivers a unified solution that connects farmers, distributors, and policymakers. Complex datasets are transformed into intuitive visual insights through interactive dashboards and map layers. This system reduces food waste by detecting disruptions early, enhances sustainability, and strengthens adaptability across food systems. Ultimately, it bridges artificial intelligence, remote sensing, and human decision-making to create a powerful tool for building climate resilience in agriculture.

Farmers benefit from early alerts that help protect crops and manage resources effectively, while cooperatives can coordinate food distribution more efficiently. Governments gain a powerful tool for planning interventions and resource allocation, and NGOs can target aid to areas most affected by climate risks. Logistics providers use predictive insights to anticipate delays and reroute shipments, while insurers and financial institutions assess climate risk exposure and investment resilience. Overall, the platform supports global food security, sustainable trade, and rapid disaster response. Its adaptable framework can even extend to space-based monitoring, offering future applications for off-world or extra-terrestrial food systems.