MOM (MINUTES OF MEETING)

Date: 27TH January 2026 – 11:00 AM to 11:30 AM

Attendees

Haripriya Varanasi - Scrum Master
Siva Mani Subrahmanya Hari Vamsi Pullipudi - Product Owner
Rishika Baddam - Development
Vaibhav Hasu- Development
Aryan Patel Kolagani - Development
Sujith Sriram Nangunoori – Development

Issac Gaang – Professor

William Martinez – Client
Gilberto Guevara Velasquez - Client

Notes:

Capstone deliverable

- An **analytics-driven decision-support MVP** for **flood early warning** (FEWS).
- End-to-end, **repeatable pipeline**: **data acquisition → cleaning → processing → analytics/insights → outputs** (UI is "last mile").
- The system should be **open-source and reusable** in other regions/countries (not Puerto Rico-only).

We need to focus on (priority)

1. **Data pipeline + data "janitor" work**
   a. Pull data from sources, clean it, standardize it, and make it machine ready.
   b. Document everything in **Jupyter notebooks** for sharing + peer review.
2. **Analytics and insights (not just maps)**
   a. Don't build a "map with dots" app.
   b. Build **metrics, patterns, trends, predictors**, and explainable signals that support decisions.
   c. Look for "unknown unknowns" (insights hidden in the data).
3. **Readiness/alerting layer**
   a. Build indicators/predictors that help and warn *before* it's too late.
   b. Emphasized sensors:
      i. **Water level** (height)
      ii. **Water flow/discharge** (speed/volume)

        iii. **Soil saturation / water table depth** ("bed of water") - this may be a key early predictor.

4. **Response layer (optional but valuable)**
    a. After a flood starts: support logistics decisions (e.g., **routes**, **drop points**, **who needs help first**).
    b. Use **demographics** (Census APIs) to support prioritization (have/have-nots, medical needs, etc.).

Data sources to use

- Public/government sources like **USGS**, **NOAA**, and Army **Corps of Engineers**.
- **US Census** demographics via API.

Constraints and working style

- **Open-source** work, likely **Apache 2.0** license.
- **Peer review culture**: others should be able to run and verify your results.
- Use **GitHub** (they want an official GMU repo so they can share their notebooks/APIs with you).
- Keep UI simple; **web-based is fine** (mobile-friendly via browser).

Organizing the team

- One big team split into subteams, e.g.:
    - **Subteam A:** data pipeline + validation + data catalog
    - **Subteam B:** dashboard/UI + "decision cues" + **playback views** (replay events so others can reproduce/verify)

Bottom line

They want a **serious, reproducible analytics pipeline** that produces **decision-ready outputs** for floods - **not** a pretty map app.