**📝 Meeting Description – Presentation Prep**

This meeting is focused on finalizing our group presentation. Each team member must:

* ✅ Prepare your **speech/script** for your assigned section
* ✅ Review the **slides** to ensure they support your talking points
* ✅ Be ready to **connect your section to the overall system and the client’s needs**, showing how your part contributes to a polished and effective solution for farmers

**🔧 Technical Context – For Everyone’s Reference**

Our project is built using a **microservices architecture**, chosen for its scalability, modularity, and maintainability.

**⚙️ Key Technologies:**

* **Backend**: Node.js with **Express.js**
* Each **microservice** acts as an individual API responsible for a specific task (e.g., fetching sensor readings, storing logs, providing analytics)
* Each service connects to its **own database**, ensuring separation of concerns and faster querying

**📱 Frontend Integration:**

* The **KMP mobile app** and **Next.js web app** communicate with these APIs
* Example:  
  → To get **sensor data**, the app calls the sensor API → which fetches from its database  
  → For **historical data**, the app calls a different API → which returns past records

**📶 Offline-First Design:**

* The app is **fully offline-capable** — all core features work without internet
* Data is saved to the device’s **local storage**
* Once internet access is available, data will **sync automatically** to the cloud databases
* This ensures that farmers always have access to their data and that central systems remain up to date

**🎤 Guidance by Role**

(*These are examples of how to tie your speech to the system as a whole and the client’s needs.*)

| **Topic** | **How to Relate It** |
| --- | --- |
| **System Architecture / Grouping** | Explain how grouping services into microservices allows us to build a modular, scalable system that’s easier to maintain and update — which is essential for long-term client use. |
| **Tent Design / Usability** | Emphasize ease of use for farmers in rural areas: a simple interface, large buttons, intuitive layout, and most importantly, offline reliability. Show how this improves their day-to-day experience. |
| **Functional & Non-Functional Requirements** | Link the client’s needs (e.g. offline access, reliability, accurate data) directly to the system design. Show how both technical and user needs are being addressed through the features implemented. |