**Smart Bin-Connect – Detailed Team Work Plan**

**🔹 Member 1 – Hardware Integration (Ultrasonic + GPS + ESP32)**

**Main Goal:** Make the bin “sense and report” its status.

**Responsibilities:**

* Connect **Ultrasonic sensor** to ESP32 to measure waste level.
* Connect **GPS module** to ESP32 to fetch real-time coordinates.
* Ensure ESP32 can read **distance (cm)** and **latitude/longitude** accurately.
* Send **data via Wi-Fi** to backend (Member 2) in a readable format.
* Test sensors for **accuracy, noise reduction, and reliability**.

**Deliverables:**

* Working ESP32 prototype sending **bin level + location** to backend.
* Pin diagram, connection notes, and sensor test logs.

**Learning Outcome:**

* Hands-on experience with **IoT hardware setup**.
* Understanding of **sensor calibration and data collection**.
* Working knowledge of **ESP32 programming and Wi-Fi communication**.

**🔹 Member 2 – Backend & Automation Logic**

**Main Goal:** Receive data from hardware and automatically generate complaints.

**Responsibilities:**

* Set up **backend framework** (Node.js / Express / Flask).
* Create **APIs** to receive sensor + GPS data from ESP32.
* Store incoming data in a **database** (MongoDB / Firebase / SQL).
* Implement **auto-complaint generation logic**:
  + If bin level > threshold → generate complaint.
  + Attach **location** and **timestamp** to the complaint.
* Optionally integrate **basic AI APIs** (OpenAI GPT for complaint text) to make complaints more natural.

**Deliverables:**

* Functional backend system that **receives data** and **logs complaints automatically**.
* Database with bin-level logs + complaints.
* API endpoints ready for integration with **Member 3’s modules** and **Member 4’s dashboard**.

**Learning Outcome:**

* Full-stack backend development (APIs, database management).
* Real-time data handling from IoT devices.
* Exposure to basic AI/NLP integration for automated text generation.

**🔹 Member 3 – User Interaction Modules (NLP + ML + Manual Input) ✅**

**Main Goal:** Enable smart, intelligent user reporting via multiple modes.

**Responsibilities:**

1. **Voice Input (NLP):**
   * Implement **speech-to-text** using Google Speech API / OpenAI Whisper.
   * Extract **user location + complaint details** from speech.
2. **Image Input (ML):**
   * Build or use a **pretrained ML model** to detect bin status from photos.
   * Output whether the bin is full, half, or empty.
3. **Manual Input:**
   * Simple form submission (location + status + optional comment).
4. **Data Formatting:**
   * Ensure all three input modes output **consistent structured data**:
   * {
   * "user\_id": "123",
   * "location": "latitude, longitude",
   * "status": "Full / Half / Empty",
   * "mode": "Voice / Image / Manual"
   * }
5. **Integration Preparation:**
   * Test output with **Member 2 backend APIs** to make sure data flows correctly.

**Deliverables:**

* Fully functional **voice + image + manual input modules**.
* Sample inputs tested and logged.
* Documentation of models/APIs used and sample results.

**Learning Outcome:**

* Practical experience with **NLP (voice-to-text)** and **ML (image classification)**.
* API handling and backend integration knowledge.
* Skills in **data formatting, preprocessing, and consistency for multi-mode input**.

**🔹 Member 4 – Dashboard & Integration**

**Main Goal:** Connect everything and provide a **visual, real-time interface** for authorities.

**Responsibilities:**

* Build **dashboard** using React, HTML/CSS, or any frontend framework.
* Integrate **real-time bin status** from backend and user modules:
  + Display **color-coded bin levels** on a map (Google Maps / Leaflet).
  + Show **complaints log** (auto-generated + user-reported).
  + Include filters (by location, status, or mode).
* Integrate **Member 3’s output** (voice/image/manual) and **Member 2’s auto-complaints** seamlessly.
* Provide optional **admin controls** for marking complaints as resolved.

**Deliverables:**

* Fully interactive dashboard with **live updates and visual representation**.
* Integrated map showing **bin locations + statuses**.
* Complaint log with timestamps, mode, and resolution status.

**Learning Outcome:**

* Frontend development + integration skills.
* Real-time dashboard handling multiple data sources.
* Practical exposure to **IoT + AI + web integration**.

**💡 Summary – Why This Division Works**

* **Member 1:** Pure hardware focus → reliable input.
* **Member 2:** Backend + automation → integrates hardware data and auto-generates complaints.
* **Member 3 (You):** High learning value → NLP + ML + data formatting, independent but feeds into backend.
* **Member 4:** Integration + dashboard → ensures all data is visible and usable.

✅ Balanced workload, clear ownership, and **all members get learning opportunities**.