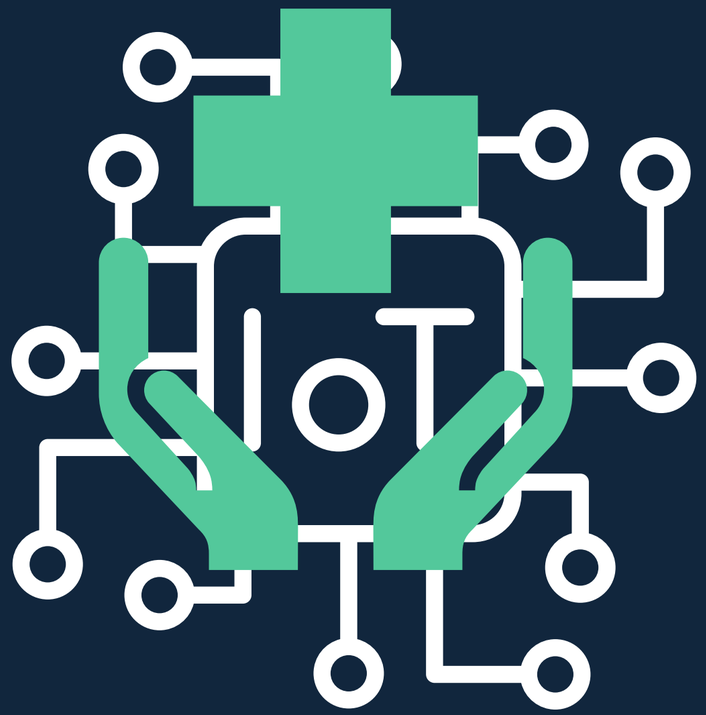


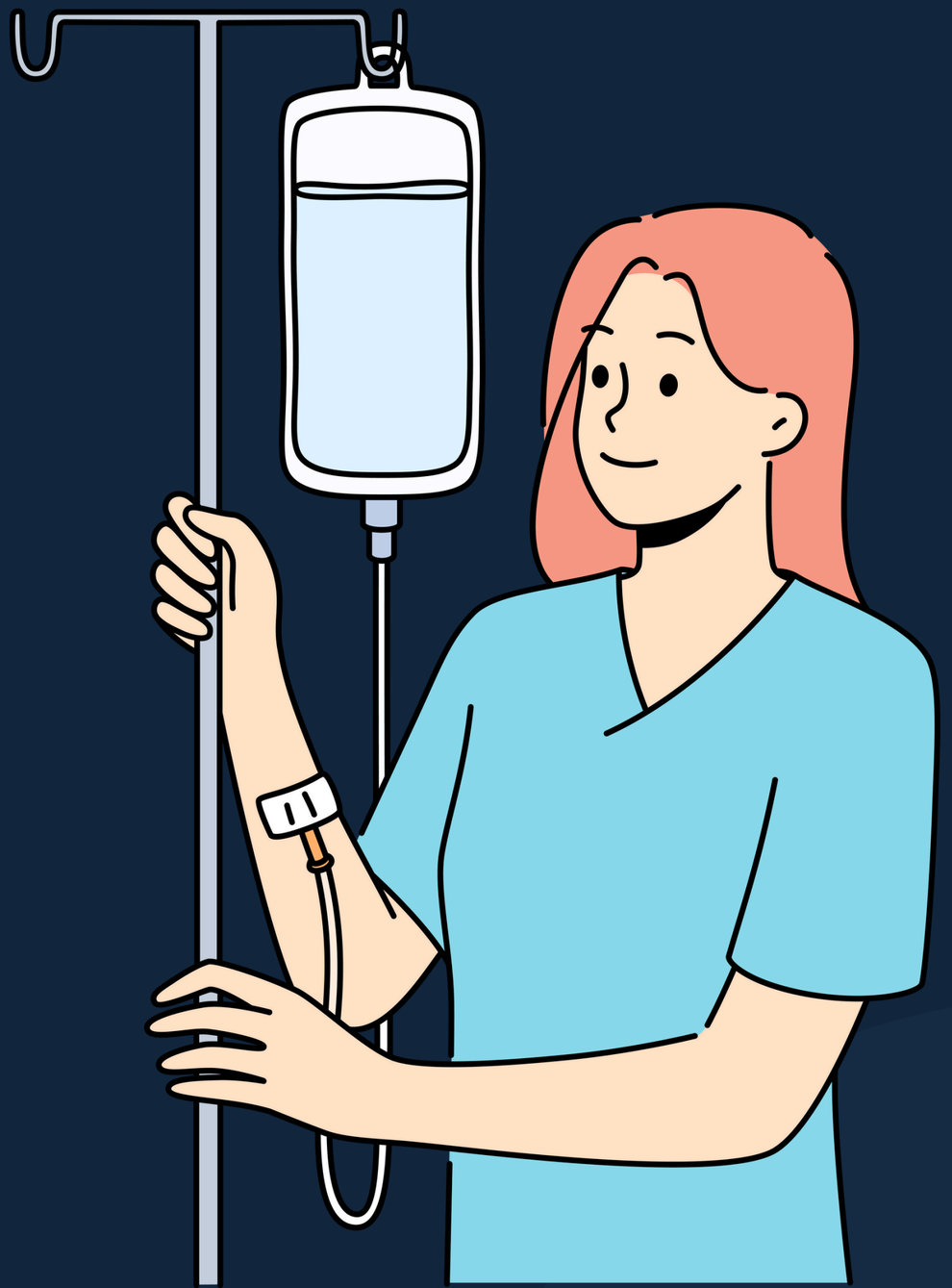
Internet of Things (IoT)-Based Saline Level Monitoring and Alert System

| HARSHIT

Problem Statement

In many government hospitals, nurses are responsible for monitoring multiple patients simultaneously. Due to high workloads, they may not notice when a saline bottle becomes empty, leading to complications such as blood flowing back into the IV line due to vacuum formation.

✳️Manual monitoring of IV fluid levels is prone to human error, which can compromise patient safety✳️

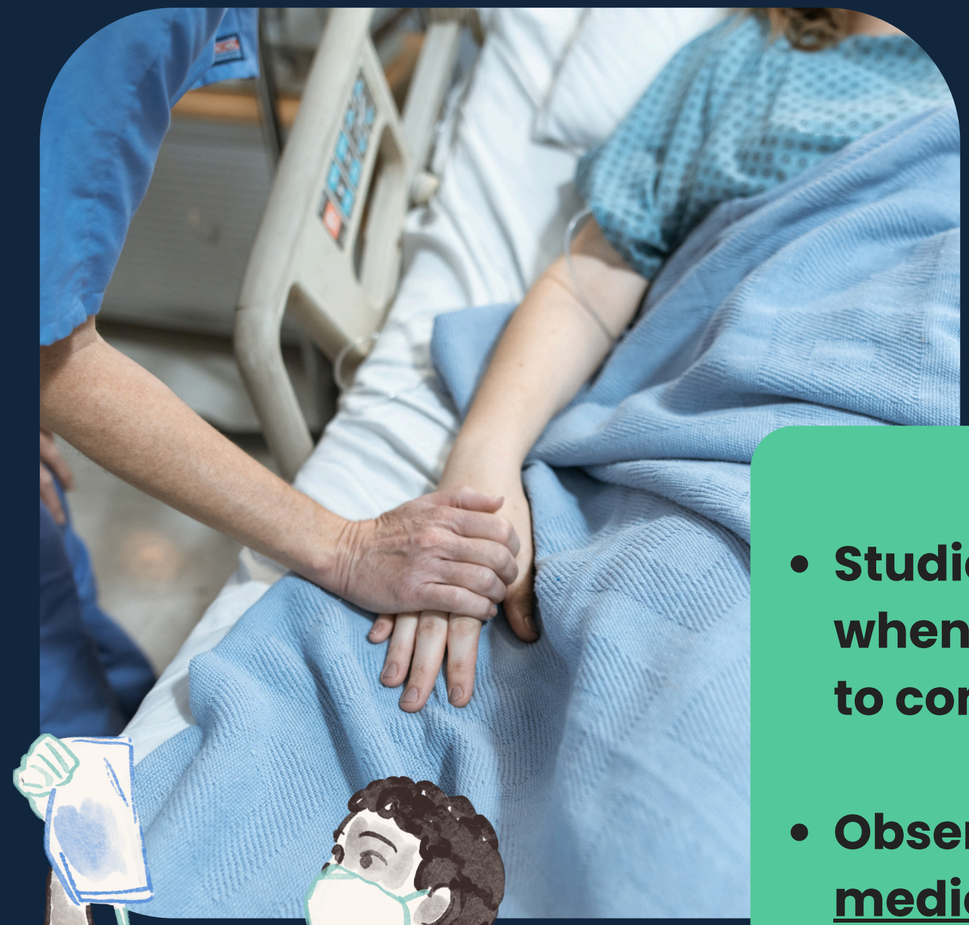


Need for the Project

- Patient Safety: Ensuring timely detection of empty saline bottles to prevent backflow and associated health risks.

- Workload Management: Reducing the burden on nursing staff by automating the monitoring process.

- Efficiency: Implementing a cost-effective solution that can be easily integrated into existing hospital setups.



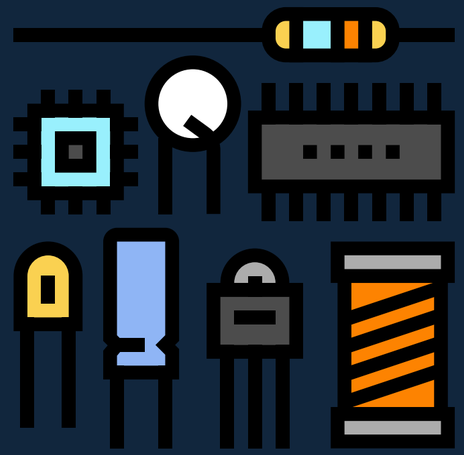
Nursing Challenges:

Error rates and automation benefits

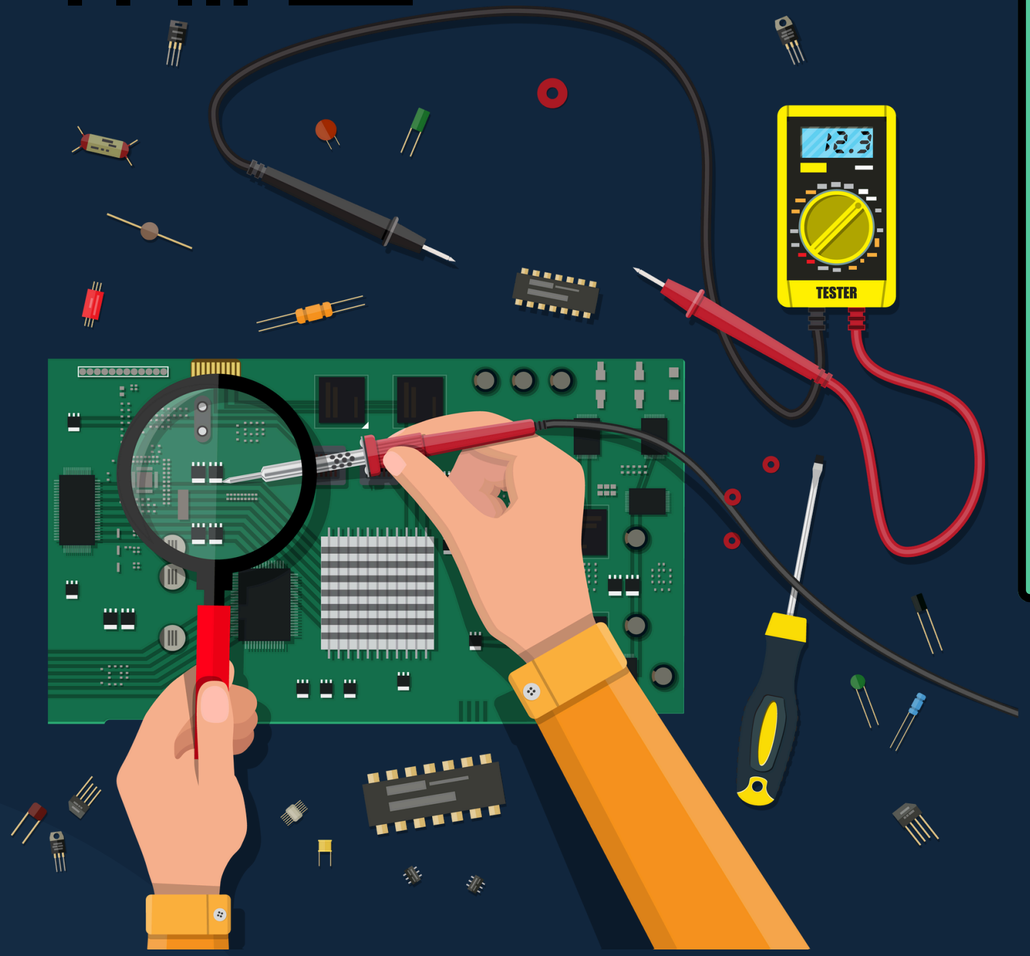
- Studies indicate that nurses experience high stress levels when administering and monitoring IV infusions, often due to complex processes and system factors.
- Observational studies in hospitals have revealed IV medication administration error rates as high as 49%, highlighting the need for improved monitoring systems.
- Implementing automated IV fluid monitoring can alleviate nursing workload and enhance patient safety.

Objective

- **Develop** an IoT-based system to monitor saline levels in real-time.
- **Alert** medical staff promptly when the saline bottle is nearly empty.
- **Ensure** patient safety by preventing complications arising from empty IV lines.

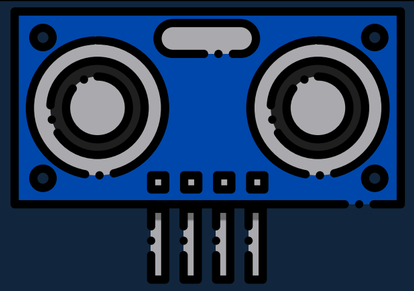


Components Used



- Hardware:

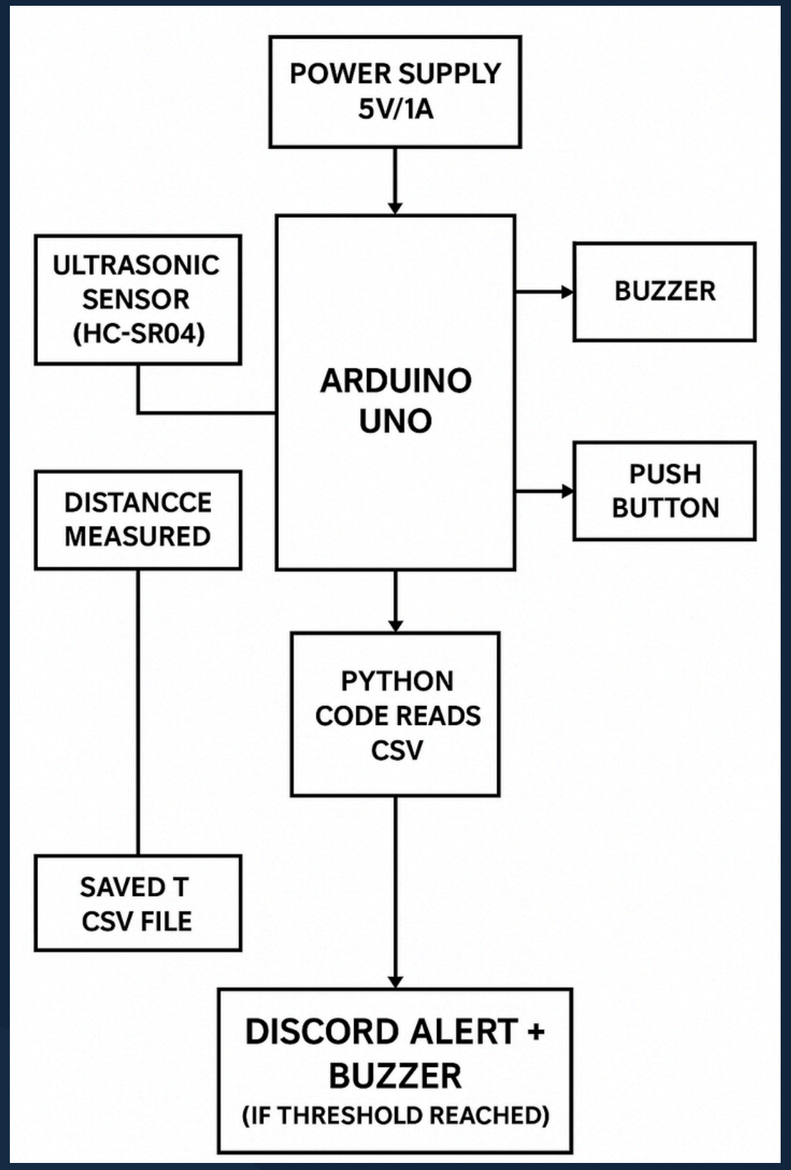
- Arduino Uno
- HC-SR04
- Ultrasonic Sensor
- Active Buzzer
- Jumper Wires
- Breadboard
- Push Button



- Software:

- Arduino IDE
- Python
- CSV File Handling
- Discord Webhook for Notifications

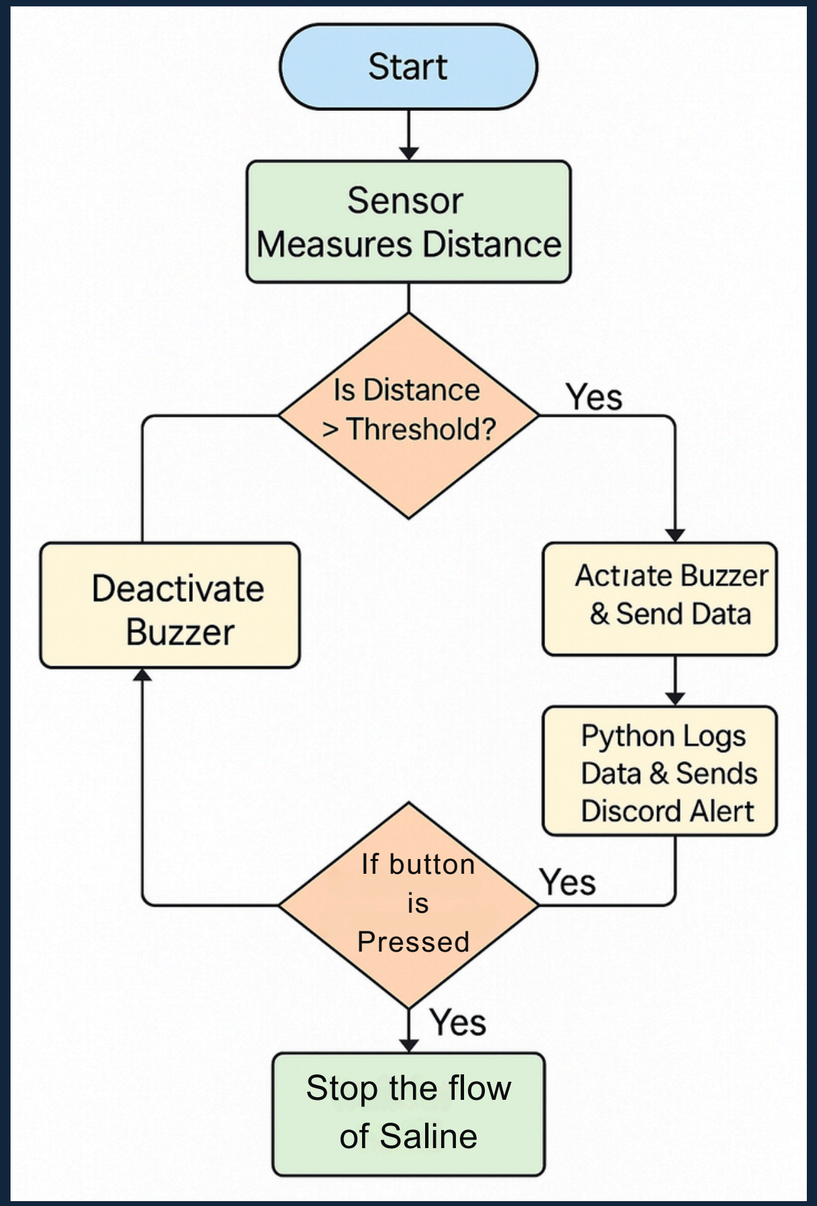
System Architecture

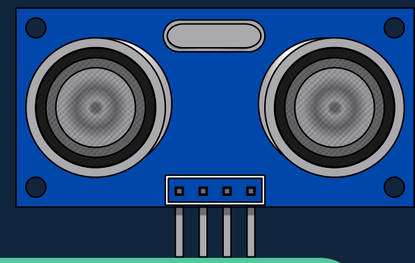


Block Diagram:

- System Workflow Overview:
- The Ultrasonic Sensor (HC-SR04) continuously measures the distance to monitor the saline fluid level in real time.
- The Arduino Uno processes the distance data and checks whether it exceeds a predefined threshold (indicating low saline level).
- When the threshold is crossed:
- Arduino sends the distance data via serial communication to a connected computer.
- A Python script receives the data, logs it into a CSV file for record-keeping, and immediately sends an alert notification via Discord to medical staff.
- At the same time, Arduino triggers a buzzer to provide a local auditory alert for nearby staff.

Working Principle & Flow chart





- **Normal Operation:**

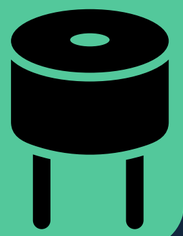
- Ultrasonic Sensor continuously measures the distance to the saline surface.

- When the saline level is adequate, the system remains idle.

- **Alert Condition:**

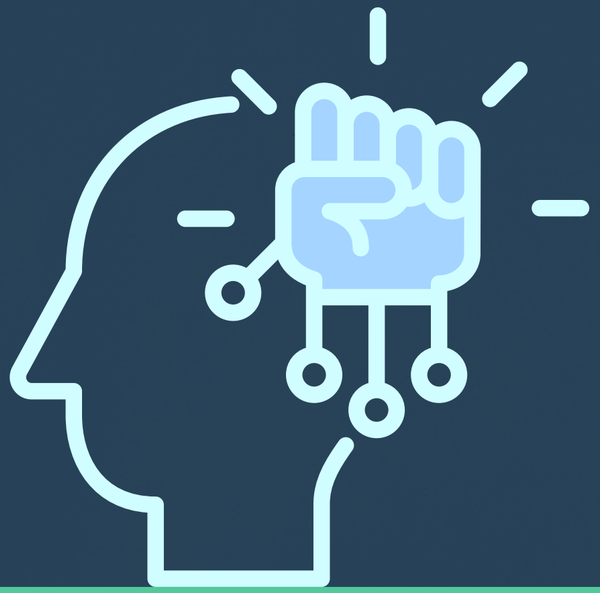
- As the saline level drops and the distance increases beyond the set threshold, Arduino triggers the buzzer.

- Data is sent to the Python script, which logs the event and sends a Discord notification.



Advantages

- **Cost-Effective:** Utilizes affordable components.
- **Scalable:** Can be implemented across multiple beds.
- **Real-Time Monitoring:** Ensures timely alerts.
- **Ease of Integration:** Compatible with existing hospital infrastructure
- **Discord Bot Advantage:**
  - Works reliably.
  - No hosting or server maintenance required.
  - Sends instant mobile alerts to staff or nurse station



Limitations of our current model

- **Dependency on Serial Communication:** Requires a computer for Python script execution.
- **Limited Range:** Ultrasonic sensor has a specific operational range.
- **Manual Reset:** System relies on manual refill detection.
- Requires a dedicated PC or Raspberry Pi to run the Python script.
- No mobile app integration (yet).
- **Not a plug-and-play device** — initial setup needed.
- **Doesn’t shut off saline flow** — only monitoring and alerting.
- Load cell calibration must be precise, else false alarms may occur.

Scalability & Cost-Effectiveness

💡 Low-Cost, Easily Deployable Solution

Hardware Cost per Unit (Prototype Level):

Arduino Uno – ₹400–₹600
Load Cell Sensor – ₹150–₹200 (future upgrade from Ultrasonic Sensor)
Buzzer – ₹10–₹20
Jumper Wires, Resistors – ₹20–₹30
Total Cost: ₹600–₹850 per bed

- No Internet Dependency:
- We opted not to use a Wi-Fi module to:
- Cut down on cost
- Avoid dependency on hospital Wi-Fi (often unstable or unavailable)
- Keep the system reliable and offline-first
- Alert System via Discord Bot:
- Free to use
- Eliminates need for hosting or SMS APIs
- Scalable – can add unlimited beds as channels/messages

🏥 Real-World Application Plan

Hospitals with 10+ Beds:
Each bed gets a sensor module with a unique ID
Load Cell accurately measures weight of the IV bottle
Arduino checks for critical low levels
When weight drops below threshold → Buzzer rings + Discord alert sent
Can be monitored remotely by nurses/staff

📈 Future Scalability

- Phase 1: Replace ultrasonic sensor with load sensor for better accuracy
- Phase 2: Build a custom mobile/web app to replace Discord
- Phase 3: Integrate dashboard for hospital-wide monitoring
- Phase 4: Incorporate battery backup for power outages

⚙️ Why It's Scalable

- Plug-and-play modular design
- Compatible with various IV systems
- No need for complex maintenance
- Affordable for rural/low-resource government hospitals

Future Plan

- **Wireless Communication:** Use **Bluetooth/Wi-Fi mesh** for offline multi-bed networks.
- **Mobile Application:** Replace Discord with our own ***mobile app*** for alerts.
- Real-time *dashboard for nurses* showing all beds and saline statuses.
- Add ***battery backup*** for operation during power cuts.
- **Integration with Hospital Systems:** Seamless integration with existing hospital management systems.

Real-World Implementation Plan

Practical Setup:

- Each bed will be equipped with a load cell placed under the saline bottle to measure its weight continuously.
- Each Arduino will be assigned a Bed ID and paired with the load cell.
- When weight drops below threshold, it sends a signal → buzzer activates → data sent via USB to Python → Discord alert with Bed ID.

Example: 10-bed Hospital

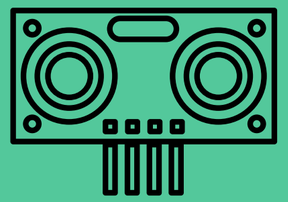
- 10 Load Cells (~₹250 each)
- 10 Arduino Uno (~₹450 each)
- Total hardware cost: **Approx. ₹7,000 – ₹8,000 only.**
- Extremely affordable compared to commercial IV monitors

Questions

Q.Why Load Cell Instead of Ultrasonic Sensor?

Load Cell Advantages:

- More *accurate and reliable* than ultrasonic sensors.
- Unaffected by environmental interference (e.g., bottle shape, sound bounce).
- Gives *precise weight readings*, which are ideal for saline volume estimation.



Q.Why We Used Ultrasonic Temporarily:

- Load cells were *unavailable to us during prototyping*.
- Used ultrasonic sensors as a proof-of-concept due to time/resource constraints

Q. Why Not Use Wi-Fi or Cloud Hosting?

Cost Saving:

- Wi-Fi modules like NodeMCU or ESP8266 increase costs (~₹300 per unit).
- Hosting a web server or app adds maintenance cost.

Connectivity Issues in Real Hospitals:

- Government hospitals often face ***unreliable internet connectivity***.
- USB-to-PC with local processing and Discord alert works **offline-friendly**

Initial Design & Learnings

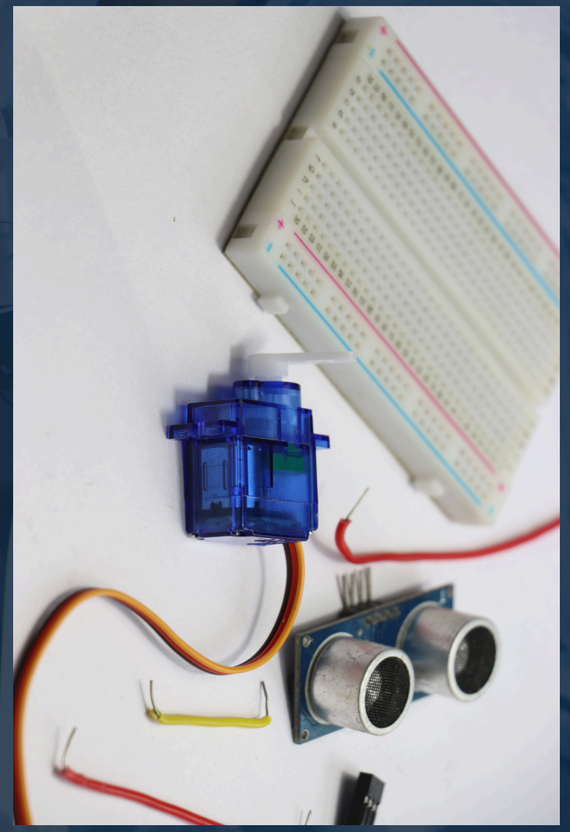
Initial Idea: Build a ***servo motor mechanism*** to stop the saline flow automatically.

- - We tested a system that pinched the IV line when the level dropped.

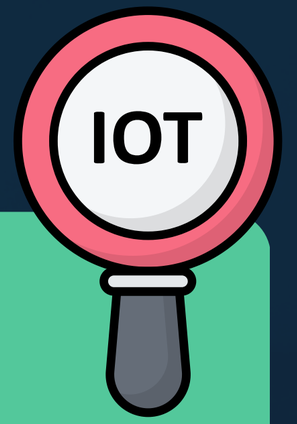
Almost worked! - But after talking to ***infirmary nurses***, we realized:

- IV lines are *disposable* and must be discarded after use.
- Adding servo parts increases cost and ***hampers reusability***.
- Real-world use prefers a ***monitoring system over mechanical intervention***.

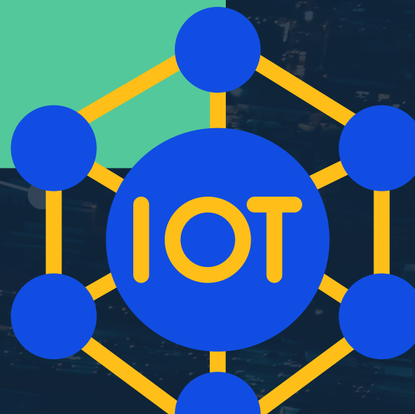
Takeaway: Importance of ***user feedback*** and ***practical thinking*** in tech design.



Conclusion



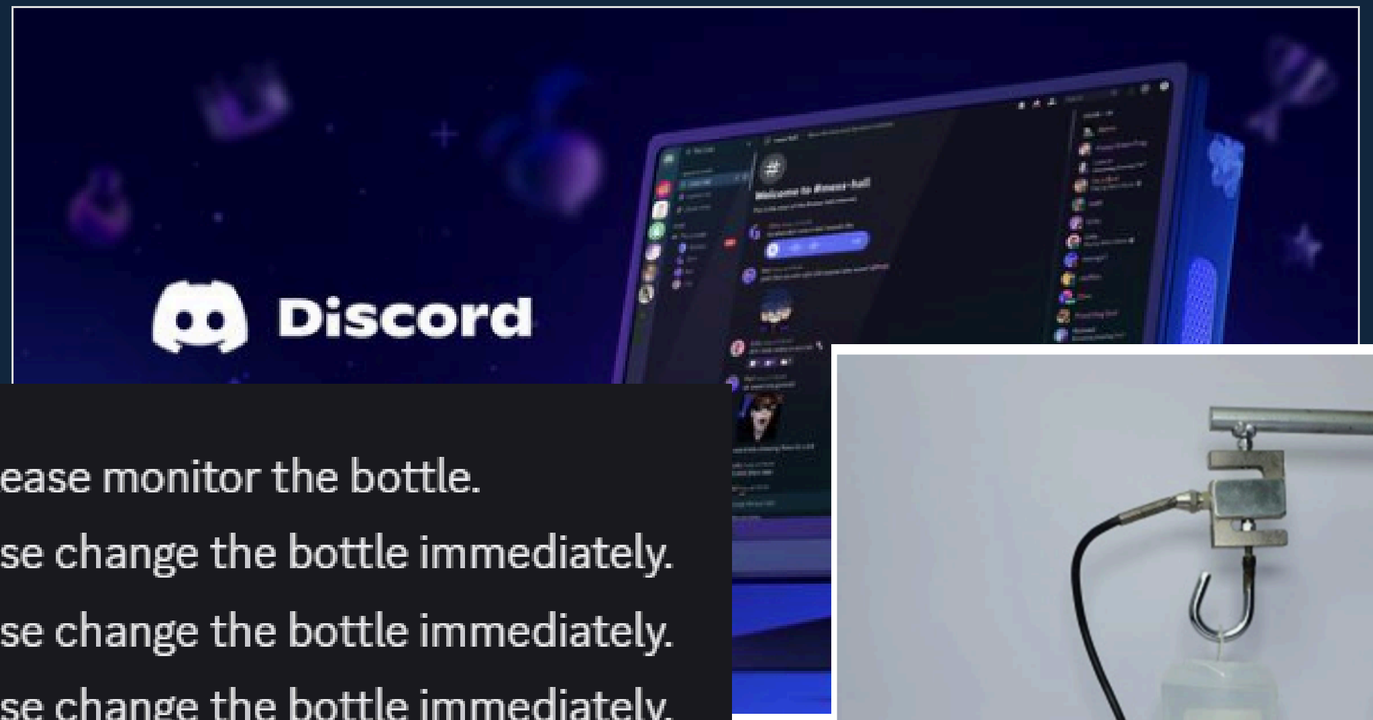
- Successfully designed a cost-effective and scalable IV fluid monitoring system using simple yet powerful components.
- Enhanced patient safety by ensuring real-time alerts through both buzzer and Discord notifications.
- Reduced reliance on manual intervention, making hospital workflows more efficient.
- Demonstrated how basic technology combined with real-world problem-solving can lead to meaningful healthcare innovations.



References

- "Automated Infusion Monitoring Device Using Arduino"
- Downloaded PDF — file:///C:/Users/harsh/Downloads/Automated_Infusion_Monitoring_Device_Using_Arduino.pdf
- "Automated Infusion Monitoring Device Using Arduino-Based IoT (Internet of Things)" - https://www.researchgate.net/publication/372927869_Automated_Infusion_Monitoring_Device_Using_Arduino-Based_IoT_Internet_of_Things
- ResearchGate Link
- "Low-Cost IV Drip Monitoring and Alerting System" (JPNR, 2022)
- Downloaded PDF — file:///C:/Users/harsh/Downloads/jpnr-2022-S03-158.pdf

Join discord link



notifcation APP 3:29 PM

⚠️ **Water level is low**. Please monitor the bottle.

🚨 **Bottle is empty!** Please change the bottle immediately.

🚨 **Bottle is empty!** Please change the bottle immediately.

🚨 **Bottle is empty!** Please change the bottle immediately.

🔎 **Please check the water bottle.** Level is getting low.

🚨 **Bottle is empty!** Please change the bottle immediately.

🚨 **Bottle is empty!** Please change the bottle immediately.

🚨 **Bottle is empty!** Please change the bottle immediately.

🚨 **Bottle is empty!** Please change the bottle immediately.