# Biomedical Assignment

## Task 1: Dataset Discovery & Justification

Candidates must find 2 publicly available datasets related to any two of the following:

- CT
- X-ray
- Ultrasound
- ECG
- Echocardiography

**Deliverables:**

- Dataset name
- Source link
- Modality
- Type of condition covered (e.g., pneumonia, fractures, arrhythmia)
- Why this dataset is useful for diagnostics

**Evaluation Signals**: Ability to search credible sources, understand modality relevance, clarity of reasoning

## Task 2: Authenticity & Clinical Reliability Assessment

For each dataset, assess:

- Source credibility (institution, journal, platform)
- Dataset size and diversity
- Annotation method (radiologist-labelled, automated, unclear)
- Ethical considerations (consent, anonymization)
- Potential biases or gaps

**Deliverables:** A short structured table or bullet analysis per dataset

**Evaluation Signals:** Critical thinking, awareness of clinical risks, ethical maturity

### Task 3: Labeling Framework Design (30 mins)

Choose one dataset and design a labeling schema.

**Example:**

- Modality-specific labels (e.g., “Ground-glass opacity” for CT)
- Binary / multi-class / severity-based labels
- Metadata fields (age, sex, view, machine type)

**Deliverables:**

- Label list with short explanations
- What should be included vs excluded
- How you would ensure consistency across annotators

**Evaluation Signals:** Medical understanding, structured thinking, practical applicability

### Task 4: Data Filtering & Quality Control Plan

Explain how you would:

- Remove poor-quality or irrelevant images/signals
- Handle missing or ambiguous labels
- Deal with class imbalance
- Decide whether a sample is “usable” or “reject”

**Deliverables:** Step-by-step quality control checklist

**Evaluation Signals:** Operational thinking, real-world data awareness

### Task 5: Insight & Reflection

Answer both:

1. What is the biggest risk of using poorly curated diagnostic data?
2. If given 6 months, how would you improve this dataset for clinical-grade AI?

**Evaluation Signals:** Long-term thinking, patient safety mindset

# HR Assessment

## Task 1:

1. What should be the top 5 priorities of an HR Manager in a healthcare startup like Medoc?

2. How is HR in a healthcare-tech company different from HR in a generic startup?

## Task 2: Hiring Plan Design

Medoc needs to hire the following in the next 6 months:

- 3 Software Engineers per month
- 20 Sales & Implementation Manager per month

Design a simple hiring plan covering:

- Hiring channels (campus, referrals, LinkedIn, etc.)
- Screening stages
- Turnaround time per role
- How you would evaluate “culture fit”

**Deliverable**: Table or structured bullets

**Signal Checked**: Hiring execution clarity, realism

## Task 3: HR Policy Judgment

Handle any two scenarios:

**Scenario A:** A high-performing engineer regularly misses deadlines and creates friction with teammates.

**Scenario B:** A junior employee claims unpaid overtime during a product launch phase.

**Scenario C:** Two team members in the same reporting line are in a personal relationship.

For each chosen scenario:

- Immediate action
- Policy-level response
- Long-term prevention

**Signal Checked:** Ethics, balance, maturity

## Task 4: Performance & Feedback Framework

Design a lightweight performance review system suitable for a 30–50 member startup:

- Review frequency
- Metrics (qualitative + quantitative)
- Feedback method
- Handling underperformance

**Signal Checked:** Practical HR systems thinking

## Task 5: Reflection & Values

Answer both:

1. What does “people-first” mean in a high-pressure startup?
2. Where should HR draw the line between empathy and accountability?

**Signal Checked**: Value alignment, leadership mindset

## *Submission Guidelines*

- Single link only
- Clear structure and headings
- No long essays; clarity over volume
- Assumptions must be stated

### *Evaluation Rubric (100 Points)*

- Role understanding & prioritization: 20
- Hiring plan realism: 25
- Scenario handling quality: 25
- Performance framework clarity: 15
- Reflection & values: 15

# Backend Intern

## Assignment: OPD Token Allocation Engine

Design and implement a token allocation system for hospital OPD that supports elastic capacity management.

### Context

Doctors operate in fixed time slots (e.g., 9–10, 10–11). Each slot has a maximum capacity. Tokens are generated from multiple sources:

- Online booking
- Walk-in (OPD desk)
- Paid priority patients
- Follow-up patients

The system must dynamically handle real-world variability such as delays, cancellations, and emergency insertions.

### Task

- Design an algorithm that:
  - Enforces per-slot hard limits
  - Dynamically reallocates tokens when conditions change
  - Prioritizes between different token sources
  - Handles cancellations, no-shows, and emergency additions
- Build the core logic as an API-based service

### Deliverables

- API design (endpoints + data schema)
- Implementation of the token allocation algorithm
- Documentation explaining:
  - Prioritization logic
  - Edge cases

    - Failure handling
- A simulation of one OPD day with at least 3 doctors

## Evaluation Criteria

- Quality of algorithm design
- Handling of real-world edge cases
- Code structure and clarity
- Practical reasoning and trade-offs

# Flutter Intern

## Assignment: Insurance Claim Management System

Build a functional Flutter application for managing hospital insurance claims.

## Task

The application must allow:

- Creation of a patient claim
- Management of:
  - Bills
  - Advances
  - Settlements
  - Pending amounts
- Claim status workflow:
  - Draft
  - Submitted
  - Approved
  - Rejected
  - Partially Settled

Features:

- Add/edit bills
- Automatic total calculations
- Status transitions
- Dashboard view of all claims

## Requirements

- Built using Flutter
- Publicly deployed (Web is acceptable)
- Must be accessible via a live link

## Deliverables

- Live application link
- GitHub repository
- 2–3 minute video walkthrough

## Evaluation Criteria

- Usability and UX
- Correctness of business logic
- Code quality
- Completeness of flows

# AI/ML Intern

## Assignment: Face Authentication Attendance System

Build a working face authentication system for attendance.

### Task

The system must:

- Register a user's face
- Identify the face
- Mark:
  - Punch-in
  - Punch-out

It should:

- Work with real camera input
- Handle varying lighting conditions
- Include a basic attempt at spoof prevention

### Deliverables

- Working demo (local or hosted)
- Complete codebase
- Documentation explaining:
  - Model and approach used
  - Training process
  - Accuracy expectations
  - Known failure cases

### Evaluation Criteria

- Functional accuracy
- System reliability
- Understanding of ML limitations

- Practical implementation quality

# APM (Associate Product Manager)

You are an Associate Product Manager at a B2B healthcare SaaS company serving small and mid-sized hospitals in Tier-2 and Tier-3 Indian cities.

Your product suite includes:

- HMS (core hospital management system)
- OPD & Billing Module
- HR & Payroll Module
- Inventory & Pharmacy Module

The company operates on thin margins. Sales cycles are relationship-driven. A single hospital can make or break a quarter.

You manage sprint planning for a product pod of **10 developers**:

| **Role** | **Count** | **Availability** |
|---|---|---|
| Backend Developers | 4 | 2 full-time, 1 at 50% (exams), 1 at 70% |
| Frontend Developers | 3 | 2 full-time, 1 at 50% (exams) |
| Mobile Developers | 2 | 1 full-time, 1 on partial leave (60%) |
| QA Engineer | 1 | Full-time |

Effective capacity this month is **~7.2 full-time equivalents**, not 10.

Each sprint is 2 weeks. You are planning for **one month (2 sprints)**.

**Active Business Situations**

**Hospital A – Existing Client (3 years)**

- Location: Tier-3 city in UP
- Paying: ₹18,000/month
- Uses: OPD + Billing
- Problem:
  - Their government audit flagged OPD printouts as non-compliant.
  - They need **layout changes in OPD print format** within **10 days**.
  - Threat: “Agar next week tak fix nahi hua, hum dusra software dekh lenge.”

Impact:

- Low revenue
- High churn risk
- Public reputation risk in the region

Estimated Effort:

- 1 Backend Dev: 3 days
- 1 Frontend Dev: 4 days
- QA: 2 days

**Hospital B – New Enterprise Client (In Closing Stage)**

- Location: Tier-2 Maharashtra
- Deal Size: ₹4.2L/year
- Modules promised in sales pitch:
  - HMS Core
  - HR & Payroll (critical)
  - OPD

- Their go-live date is in **3 weeks**.
- CEO has told Sales: “HR module pehle chahiye, warna onboarding rukega.”

Impact:

- Largest deal of the quarter
- If delayed, high chance of deal collapse
- Sales team already overpromised features

Estimated Effort (HR Module MVP):

- Backend: 18 dev-days
- Frontend: 12 dev-days
- Mobile: 6 dev-days
- QA: 6 dev-days

**Hospital C – About to Close**

- Location: Bihar
- Deal Size: ₹1.5L/year
- Current blocker:
  - They run a pharmacy-heavy workflow
  - Need **Inventory + Expiry Tracking + Low-stock alerts**
- Sales says: “Without inventory, they won’t sign.”

Impact:

- Medium revenue
- 80% probability to close if feature is demo-ready
- Competitor already pitching

Estimated Effort:

- Backend: 10 dev-days
- Frontend: 6 dev-days

- QA: 4 dev-days

**Internal Constraints**

- Two developers have university exams this month.
- One mobile dev is on partial leave.
- QA is already overloaded due to regression backlog.
- CTO insists:
    - "No more than 20% tech debt carryover."
    - "At least one sprint must include stability work."

**Your Task**

Design a **one-month sprint plan (2 sprints)** that:

1. Prioritizes work across:
    - Retention risk
    - Revenue impact
    - Sales dependency
    - Engineering constraints
2. Allocates developers realistically using the **actual capacity**, not headcount fantasy.
3. Clearly defines:
    - What will *not* be done this month
    - Which stakeholders will be disappointed
    - Why those trade-offs are rational
4. Includes a communication strategy:
    - What do you tell Sales?
    - What do you tell Customer Support?
    - What do you tell the CEO?
    - How do you frame delays without burning trust?

## Deliverables

1. **Sprint Roadmap (2 Sprints)**
   - Task-wise breakdown
   - Developer allocation
   - Time estimates
2. **Visual Plan**
   - Kanban board, flowchart, or timeline
   - Must show overlaps, bottlenecks, and risk zones
3. **Written Justification**
   - Why Hospital A is or isn’t prioritized
   - Why HR for Hospital B is or isn’t first
   - Why Inventory for Hospital C is or isn’t delayed
   - How capacity constraints shaped every decision

## Evaluation Criteria

- Quality of prioritization logic
- Awareness of real-world constraints
- Honesty of trade-offs
- Practicality of execution
- Maturity of communication strategy

# Hospital Growth Manager

## Assignment: High-Impact Cold Outreach

Write a single cold email to:

hr@medochealth + assignment

Example:

- hr+assignment@medochealth.com

You may:

- Sell any product
- At any price
- Using any pitch style

## Rules

- No templates
- No generic language
- The email must be compelling enough to demand a response

## Deliverables

- One complete email

## Evaluation Criteria

- Persuasion quality
- Originality
- Business clarity
- Ability to trigger engagement

# Founder’s Office Intern

## Assignment: Founder-Level Outreach

Write a single email addressed to:

hr@medochealth + assignment

Example:

- hr+assignment@medochealth.com

This email must be written from the perspective of someone working in the Founder’s Office.

You may:

- Propose any product, idea, or opportunity
- At any scale or price point
- In any style

However, this is not a feature pitch or a routine sales mail.

The email should:

- Reflect strategic thinking
- Show understanding of how businesses grow
- Frame the idea in terms of leverage, direction, or long-term value
- Feel like something a founder would actually pause to read

## Deliverables

- One complete email

## Evaluation Criteria

- Strategic depth
- Positioning and tone

- Originality
- Business understanding

# Embedded System Engineer

We are building a wearable AI assistant.

It clips to a shirt, listens for a command, and talks to an LLM (Large Language Model) in the cloud.

**The Challenge:**
It must run on a tiny battery for **12 hours**.

**Part 1: Hardware & Architecture**

**Goal:** Prove you understand low-power design and audio processing.

**Question 1: The "Always-Listening" Problem**

To catch a wake word (like *"Hey Computer"*), the microphone must always be on.
This drains the battery.

**Proposed Design:** Using a dual-core SoC (System on Chip)

- **Core A:**
    - High performance
    - Wi-Fi capable
    - Power hungry
- **Core B:**
    - Low power
    - Slow speed
    - Limited RAM

**Your Task:**
Explain how you would split the work between Core A and Core B to maximize battery life.

- Which core handles wake-word detection?
- Which core handles cloud upload?

### Question 2: Memory Calculation (Audio Buffer)

We need to record user audio before sending it to the cloud.

- **Format:**
  - 16 kHz sample rate
  - 16-bit depth (2 bytes per sample)
  - Mono
- **Duration:** 10 seconds
- **Constraint:** The chip has **256 KB of SRAM** available for this buffer.

**Task:**
Calculate the total size of the 10-second audio clip.

- Will it fit in RAM?
- Show your math.

### Question 3: Privacy Hardware

Users are worried about being recorded secretly.

- We want a **"Hard Mute"** switch that physically cuts microphone power.

**Schematic Check:**
If the software crashes, can it accidentally turn the mic back on?

- How would you wire the switch to ensure *absolute* privacy?

## Part 2: Firmware Logic (C Code)

**Goal:** Write a robust state machine for the device.

### The Scenario

Write the C code that manages the device's main states.
You are writing the *"Brain"* of the pin.

### Mock Functions (Assume these exist)

### Mock Functions (Assume these exist)

- bool is_button_pressed(void)
  Returns true if the user holds the touch bar.
- int get_battery_level(void)
  Returns a value from 0 to 100.
- void set_led_color(int color)
  Colors: OFF, GREEN (Listening), RED (Error).
- void start_audio_stream(void)
  Starts sending audio to Wi-Fi.
- void stop_audio_stream(void)
  Stops sending audio.

### The Logic Rules

1. **IDLE State**
   - Default state
   - LED is OFF
   - Audio is OFF
2. **LISTENING State**
   - Enter ONLY if:

- Button is pressed **AND**
- Battery > 10%

- If battery < 10%, go to **ERROR** immediately
- Turn LED GREEN
- Start Audio

3. **Hold-to-Talk**
   - Device stays in LISTENING only while the button is held
4. **Finishing**
   - When the button is released:
     - Stop Audio
     - Turn LED OFF
     - Return to IDLE
5. **Safety**
   - If the button is stuck (held for > 30 seconds):
     - Force a reset to IDLE
     - Prevent overheating

### Constraints

- Use a switch statement for the state machine
- Do **not** use floating point numbers
- Write clean, readable code