Technical Task: Infection Cluster Detection (Backend-first, tiny UI)

- **Timebox:** ~5 hours (stretch goal optional). Use of AI in your development process is *highly* encouraged.
- **Deadline:** Saturday, 17:00 (Bangkok time).

Objective

Build a Python service that ingests hospital records containing patient movement data and infection test results, then automatically detects clusters of infections. Expose a minimal UI to view detected clusters, their members, and basic metrics.

Stretch goal: For each cluster, generate a concise, clinician-friendly LLM summary describing its size, locations, timeframe, and possible significance.

Background

Hospitals need to identify and investigate infection clusters—groups of related cases that may indicate a common exposure or outbreak—as early as possible. These clusters often emerge from contact tracing data, where patients are linked through shared locations (e.g., wards, theatres) or overlapping stays.

Currently, contact tracing is often performed manually, weeks or months after transmission has occurred. At NEX, our goal is to fully automate this process so infection prevention teams are alerted the moment a transmission could have occurred.

Your task will simulate this process using synthetic hospital data. Any algorithm, data structures, and optimisations are entirely up to you.

Background Reading

- CDC: Overview of Outbreak Investigation (Outbreak Investigation)
- Recent Studies:
    - Network models to detect hidden clusters of infection in hospitals (Wan 2023).
    - The CATHAI tool (CATHAI App Demo, Cuddihy 2022) uses genomic sequencing to identify related pathogens in real-time.
    - Advanced mathematical algorithms for detecting clusters of indirect transmission (Myall 2022).
    - Multisource surveillance integrating antibiotic usage, bacterial specimen inspection rates, and positive culture results (Fan 2019, and Fan 2020).

Data (CSV we'll provide)

transfers.csv

Line rows where each row is a single patient's record of entry and departure from a single hospital ward. A patient may have multiple ward visits during their hospital stay, or across multiple hospital stays.

- transfer_id (string) : primary key, unique transfer identifier.
- patient_id (string) : Foring key, unique patient identifier
- ward_in_time (date) : date of admission to ward
- ward_out_time (date) : date of discharge from ward
- location (string) : the location of the patient

microbiology.csv

Records of all microbiological tests performed in the hospital. Each microbiology test is taken from one patient, it confirms the presence or absence of a microbial pathogen.

- test_id (string) : primary key, unique test
- patient_id (string) : foreign key linking to a unique patient identifier
- collection_date (date) : the date on which the test was taken on
- infection (string) : i.e. the infection type being tested for

- result (string) : 'positive' or 'negative' based on whether the infection was identified

---

What is a cluster?

A cluster is defined as: Two or more patients with a positive test for the same organism who are linked in both space and time by at least one contact event.

- **Contact event:** An occasion where two patients were in the same location (e.g., ward) on the same day.
- The contact must occur within ±14 days of both patients' positive test dates (more contact within this window indicates a more substantial likelihood of transmission).
- **Clusters can extend beyond a single pair of patients:** if Patient A is linked to Patient B, and Patient B is linked to Patient C, then all three belong to the same cluster (even if A and C never directly overlapped).

Requirements

Backend (Python preferred)

- Ingest the two CSV datasets.
- Identify potential infection clusters for each infection based on the definition provided above.

Minimal UI

- Table/List/Network of detected clusters.
- Simple way to view details of a cluster.

Stretch goal — LLM Analyser

- For each cluster, produce a short, plain-language summary suitable for clinicians.
- Can use OpenAI API or any LLM; must fall back to a mock response if no API key is set.
- Keep summaries factual and ≤120 words.

Deliverables

Deadline: Saturday, 17:00 (Bangkok time) & be prepared to share your screen demo'ing the prototype.

- Code + README with:
- Functional Prototype (does not need to be pretty!)

Task Presentation

Prepare to:

1. Demo your solution live (share your screen)
2. Walk us through:
    - Your architecture
    - Your cluster detection approach
    - Discuss design decisions and trade-offs
3. Share ideas for future expansion of the system