POI Management System — Demo Checklist (Printable)

Prepared for recording the final demo
Date: {{today}}

Before you hit record

- Open VS Code and run `main.py` in a terminal.
- Ensure `big_demo.json` is in the project root.
- Menu shows numeric order (1..20, 0 last).
- All dates use **dd/mm/yyyy**; ratings are **integers 1..10**.

---

1) Intro (10–20s)

- Small OOP project: **Types → POIs → Visitors → Visits**.
- Distances on **centers**; invariants: **no POI id reuse**, **delete type only if unused**.
- Deterministic tie-breaks; **distinct counts by default**.

2) Load data (optional per run)

- **20** → Load config from JSON → `big_demo.json` → “Config loaded.”
- **4** → Counts per type (note: `zoo` has **0** POIs).

3) Attributes & PQ1

- **6** → List POIs of type `museum` → shows attribute dicts per POI.
  Note: add/delete/rename attributes **migrates values** across existing POIs.

4) Spatial queries (centers + ε boundary)

- **9** → Within radius (PQ4): center **(0,0)**, radius **5** → expect `(0,0)` park and `(3,4)` cafe.
- **10** → Exactly on boundary: same center/radius → **only** `(3,4)`.
- **5** → Nearest-k (PQ5): point **(10,10)**, **k=3** → tie-break **distance → id → name**.

5) Visitor & POI lists

- **7** → Visitor history (VQ1): visitor **1** → all visits sorted by **date**.
- **8** → Visitors for a POI (VQ2): POI **1** → show **distinct = y** then **n** to compare.

6) Rankings & counts (distinct by default)

- **11** → Top-k POIs by **distinct visitors** (VQ5), **k=3**.
- **12** → Top-k visitors by **distinct POIs** (VQ4), **k=3**.
- **13** → Counts: **distinct visitors per POI** (VQ2) — includes **zero-visit** POIs.
- **14** → Counts: **distinct POIs per visitor** (VQ3) — includes **zero-visit** visitors.

7) Coverage fairness (VQ7)

- **15** → **m = 3**, **t = 2** → visitors with ≥ **3 DISTINCT POIs** across ≥ **2 DISTINCT types**.
  In `big_demo.json`: **Sam** and **Bo** qualify (add Library visit to make **Ana** qualify).

8) Live inputs (kept out of JSON to demonstrate interaction)

- **1** → Add type: `library` with attr `books`.
- **2** → Add POI: **id 900** `City Library` (type `library`) at **(15,15)**.
- **3** → Record visit: **visitor 2** → **poi 900** → **15/10/2025** → **rating 9**.
  (QoL: existing visitors don't prompt name/nationality again.)
- Re-run **5 / 11 / 12 / 15** to show immediate effects.

9) Rename + migration + deletion rules

- **17** → Rename museum attribute: `tickets` → `ticket_price`; verify with **6** (PQ1).
- **18** → Rename POI type: `gallery` → `art_gallery`; verify with **4** (Counts per type).
- **19** → Delete POI type: try `park` (should **fail** — in use), then delete `zoo` (unused) → **Deleted**.
- **16** → Delete POI **id 4**; verify with **13** (Counts per POI).
  Visitor history (VQ1) still shows past visits (we **preserve history**).

10) Validations (show one example)

- **3** → Record visit with **rating 11** → error "Rating must be 1..10".
- Or bad date `2025-10-10` → error "Date must be `dd/mm/yyyy`".
- Then record a **valid** visit to show success.

Wrap-up (15–30s) — talking points

- Encapsulation with **@property**; **typing** hints; `from __future__ import annotations`.
- Two-file core (`models.py`, `registry.py`) + optional loader (`config.py`).
- Distances use **centers**; **epsilon-aware** boundary comparisons.
- PQ2 closest pair **O(n^2)** by choice (documented in report).
- Date format **dd/mm/yyyy**; **venv** used; `requirements.txt` tracked.

Live typing cheat-sheet

- PQ4/10: **(0,0)**, **r=5**
- PQ5: **(10,10)**, **k=3**
- New type: **library** / **books**
- New POI: **id 900**, `City Library` @ **(15,15)**
- Visit: **visitor 2** → **poi 900** → **15/10/2025** → **rating 9**
- VQ7 thresholds: **m=3**, **t=2**

Print tip: In your browser, choose **Print → Save as PDF** for a clean export of this page.