Take-Home (2–3 hours): “Mini Support Queue”

Build a small React app (TypeScript optional) that lets an agent manage a ticket queue. You can use either a tiny mock API (recommended: MSW) or simple in-memory functions.

Data model
A ticket has:
id, title, description, priority (VIP | Regular), status (Open | Assigned | Resolved), createdAt, assignee

Requirements
1) Create Ticket (Form)
Create a form with:
- Title (required, 5–80 chars)
- Description (required, 20–500 chars)
- Priority (VIP or Regular)

On submit:
- Add the ticket to the queue (new tickets start as Open)
- Show inline validation errors

2) Queue List (Agent View)
Display a list/table of tickets with:
- Title, Priority, Status, Created time
- Filter by Status (Open/Assigned/Resolved)
- Queue ordering: VIP tickets first, then Regular; FIFO within each group

3) Assignment Actions
Implement two actions:
- Assign Next: assigns the next eligible ticket (next Open ticket respecting ordering) to “me” (hardcode agent = agent-1)
- Assign Specific: each ticket row has an “Assign to me” button (only if status is Open)

4) Resolve
Each assigned ticket can be marked Resolved.

5) “Backend-ish” behavior (light)
Use one of:
- MSW mock API with GET /tickets, POST /tickets, POST /tickets/:id/assign, POST /tickets/:id/resolve
- OR a simple in-memory “service layer” that returns Promises

Add:
- Simulated latency (300–800ms)
- Simulated failure on assignment (e.g., 10% chance) and show an error banner/toast

Deliverables
GitHub repo + README (setup + assumptions)

Keep it simple, but include:
- Loading/empty/error states
- 1–2 tests (e.g., queue ordering + assign-next logic)

Evaluation focus
Correct ordering + state updates, clean component structure, UX for errors/loading, and reasonable API/service abstraction.