QA take home Challenge

We've designed this take-home QA challenge to help us better understand how you approach testing, think critically, and communicate your findings. The tasks included are intended to reflect the types of challenges you may encounter in this role.

Please read the instructions carefully, and don't hesitate to reach out if you have any clarifying questions. While we aim to keep the challenge reasonable and respectful of your time, we encourage you to treat this as a chance to showcase your skills and thought process.

We look forward to reviewing your submission!

📁 0-README

Time-box: ~4 h of effort, return within 5 days.

Output: single PDF *or* Git repo (Markdown) ± optional 5-min Loom walkthrough.

Assignment Tasks

Be concise – reviewers will not read > 5 pages.

Task A – "Release-0" plan

Create a pragmatic release checklist for enabling hedged reporting in next month's window.

Include: smoke & regression scope, envs, cut-off, owner matrix, comms, rollback/flag strategy.

Task B – Defect-triage matrix

For BUG-101/102/103: guess origin (FE, API, Ops, unclear), next investigative step, evidence needed, SLA class, block / ship decision.

Task C – Finance QA snippet

Design and describe one automated (or semi-automated) test that verifies the hedged TWR calculation end-to-end. You may use any programming language, pseudo-code or structured test design. Keep the solution concise, without full framework boilerplate. Additional points for short description of how you would validate correctness e.g. test data setup, expected value, where the test would fit in QA process.

---

1 – System Snapshot

1.1 High-level architecture

```
┌──────────────────────────┐
│  React Front-End (WebCo) │  ← modern stack (Vite + TS)
└────────────▲─────────────┘
             │HTTPS / REST
```

```
┌──────────────────────────────┐
│ "ProdX" API image            │   ← vendor-supplied Docker image
│ (pricing, orders, rpt)       │
└──────────────▲───────────────┘
               │connection (internal)
┌──────────────┴───────────────┐
│ Postgres (managed)           │
└──────────────────────────────┘

* **Cloud:** private OpenShift *or* public AWS EKS.
* **Ops release cadence:** **one production release per month** (typically
first Tuesday).
* **Business context:** The platform is used by **Independent Asset
Managers (IAMs)** to onboard, monitor and advise their private-bank
clients. All interaction happens through a **browser-based portal**; there
is no native desktop or mobile app.
```

1.2 What the QA / Release Manager coordinates

Step	Responsibility
✉️ **1 – Gather builds**	*Ask* WebCo for latest signed-off FE image; *ask* ProdX vendor for latest API image **and** migration script.
🔍 **2 – Deploy & test**	Work with IT Ops to deploy both images to **DEV → INT → PROD**; run smoke + regression packs at each stage.
📣 **3 – Go/No-Go**	Hold async checklist; verify ticket closure; sign-offs from Product, Ops, Vendor.
🚀 **4 – Prod push**	Coordinate IT Ops during the monthly window; monitor dashboards; run post-deploy checks.
✅ **5 – Follow-up**	Confirm success, file retro items, raise vendor tickets if defects surface.

---

2 – Change Request – "Enable Live Currency-Hedged Reporting"

This feature **leverages an existing vendor API**, plus small Ops config and FE work. No new backend code is required.

2.1 User story

As an IAM portfolio manager accessing the web portal

I want reports to show returns *after* FX hedging in real-time

So that clients see performance net of currency risk.

2.2 Acceptance criteria

ID	Given / When / Then
AC1	**Given** a EUR account with USD assets **When** I open "Performance" **Then** the chart shows hedged time-weighted return (TWR).
AC2	**Given** hedging is **disabled** via Ops config flag **Then** UI behaves exactly as today.
AC3	**Given** hedging is **enabled When** I export a PDF **Then** both chart & table reflect hedged figures to 4 dp.
AC4	UI must gracefully handle API latency > 1 s (loader + max-3 retries).
AC5	Audit events stored in audit.fx_hedge_requests with user-ID, timestamp ms, elapsed ms.

2.3 Tech touch-points

Component	Change
API	Call **existing** endpoint GET /pricing/v2/hedged (already GA).
FE	Add toggle logic, loader, chart values, PDF template tweak.
Ops	Introduce flag X-HEDGE-ENABLED; run vendor migration script 238 to ensure audit table exists.

3 – Known Quirks & Constraints

- **ProdX sandbox quota** – only 10 test portfolios; resets nightly at 02:00 UTC.
- **WebCo cadence** – merges to main Tuesday & Friday 16:00 CET. Notably, **WebCo is an external consultant.**
- **Ops release window** – **one release per month** (first Tuesday, 18:00-20:00 CET).
- **Logging** – FE logs in Loki via X-Request-ID; API logs stdout; vendor has no remote access.
- **Browser mix** – 85 % Chrome, 10 % Edge, 5 % Safari among IAM users (no IE).
- **SLA** – P0 = block release; P1 = ship with workaround; P2 = sprint backlog.

4 – Sample Bug Reports (to triage)

Key	Summary	Details
BUG-101	Wrong IRR on PDF for multi-currency account	Steps: portfolio qa-hedge-003, hedged flag ON, export PDF. Expected 5.43 %, got 5.38 %. Raw JSON shows irr:5.42571.
BUG-102	502 from /pricing/v2/hedged spikes > 30 %	Observed 21-Jul-2025 17:40-18:10 UTC. Logs show upstream reset.
BUG-103	Dashboard caches stale data after blue-green switch	After release-223, some users see yesterday's values until hard refresh. Cloud-flare page-rule suspected.

5 – Finance Reporting Spec (excerpt)

5.1 Formulas

```
Definitions
-----------
V_t  : portfolio value at time t (base ccy)
C_t  : net cash flow at time t
FX_t : hedge ratio at time t (0-1)

Hedged Time-Weighted Return
---------------------------
Step 1  V'_t = V_t × (1 − FX_t × (1 − h_t))
Step 2  TWR  = Π_{i=1}^{n} ( V'_i − C_i ) / V'_{i-1}  − 1
Step 3  Annualise if > 365 d: TWR_ann = (1+TWR)^{365/days} − 1
```

Edge cases: zero-value period, missing FX, FX > 1, rounding (4 dp display, 6 dp internal).