RaftWorks formerly LowCodeFlow

FREE RESOURCE

THE BUILD SHEET SERIES · 03

The HR Tech Build Checklist

HR tools live or die on two things: **whether the permissions model matches how the org actually works, and whether anyone opens the tool in week three.** This checklist covers both — the data model, the integrations, and the adoption mechanics most builds skip.

(01) People data — handle with care

it's all sensitive

- [ ] **Classify every field:** directory info, compensation, performance, health/leave — each needs a different access rule.
- [ ] **Write the retention policy first:** what happens to a candidate's data after rejection, an employee's after exit.
- [ ] **GDPR-proof the basics:** export and erasure flows for candidates and employees, built before launch.
- [ ] **Keep compensation and performance data out of general tables** — separate storage, separate permissions.
- [ ] **Red flag:** salary data reachable by any "admin" role that support staff also hold.

(02) The permissions matrix

draw it before building

- [ ] **Four base roles minimum:** employee, manager, HR, admin — mapped against every screen and action in a simple grid.
- [ ] **Managers see their team, not the org.** Reporting lines drive visibility, and they change — sync them, don't hardcode.
- [ ] **Approval chains as data, not code:** leave, expenses, offers — configurable steps so a re-org doesn't need a rebuild.
- [ ] **Anonymous where promised:** surveys and feedback need real anonymity (thresholds, no raw exports), not a checkbox.
- [ ] **Audit log for HR actions** — offer changes, comp edits, terminations — who did what, when.

(03) Integrations — meet people where they work

adoption lives here

- [ ] **SSO first (Google/Microsoft).** Every login screen you add costs you adoption.
- [ ] **Map the system of record:** which fields your tool owns vs which sync from the HRIS/payroll — one direction per field.
- [ ] **Slack/Teams for the moments that matter:** approvals, reminders, and nudges happen where people already are.
- [ ] **Calendar sync for anything scheduled** — interviews, reviews, onboarding sessions.
- [ ] **Plan the CSV escape hatch:** HR will always need imports and exports; make them first-class, not an afterthought.

(04) Launch & adoption QA

week three is the test

- [ ] **Log in as each role** and try to reach data you shouldn't — especially comp and performance.
- [ ] **Run one full workflow end-to-end:** e.g. job posted → applied → interviewed → offer → onboarding checklist done.
- [ ] **Fire every notification once** — then check the digest settings, because notification fatigue kills HR tools.
- [ ] **Test the re-org:** move a manager, confirm visibility and pending approvals follow correctly.
- [ ] **Onboard a real non-HR user with zero instructions** — if they need a walkthrough, the launch isn't ready.
- [ ] **Measure week-3 active use, not week-1** — and have one owner in HR accountable for it.

Building an HR tool?

We've shipped applicant tracking systems, field-workforce apps, and role-based team platforms — permissions matrix first, integrations included.

raftworks.co/hr-tech

connect@lowcodeflow.co

Not legal advice — confirm employment-data rules for your jurisdictions