Purchase Management — User Flow & API Guide

Nov 24, 2026 · @Leela

Purpose and audience

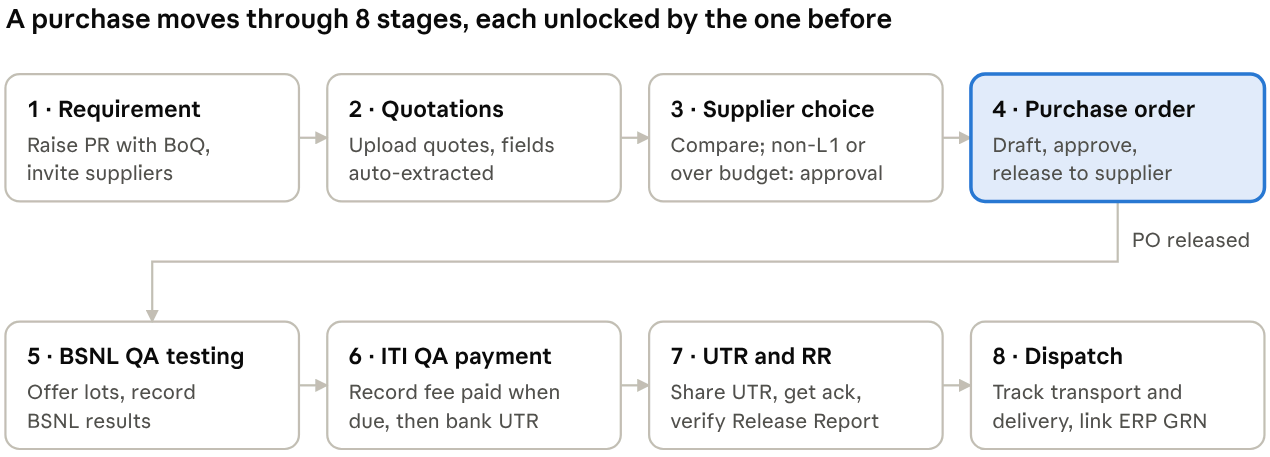
The Purchase module tracks one purchase from requirement to delivery in 8 screens. The frontend is built and runs on sample data stored in the browser. This guide shows what each screen does, who uses it, and which APIs replace the sample store.

- **Users** (Purchase team, managers, admins, finance): read sections 2 to 4 for the flow, roles and statuses.
- **Backend developers**: read sections 4 to 9 for the data model, endpoints, rules and the frontend functions each endpoint replaces.

Ledgr records what happens outside it. It does not send payments to banks. It does not create GRNs or stock entries; those stay in the company's ERP.

End-to-end user flow

A purchase runs from requirement (stage 1) to delivery (stage 8). Each stage opens only when the one before it is complete. The Purchase Order is the hinge: stages 5 to 8 all work on a released PO.



purchase flow · 8 stages

Approvals happen at stage 3 (only for a non-L1 or over-budget choice), stage 4 (every PO) and stage 7 (RR verification).

Stage	Screen and menu	Who acts	What the user does	Stage is done when
1. Requirement	Purchase Requirements (Screen 1)	Purchase staff	Create a PR with BoQ lines, budget, closing date; invite suppliers. Can save as draft	Suppliers are invited
2. Quotations	Requirement Details (Screen 2)	Purchase staff	Upload each supplier's quotation; fields are read from the file and checked by the user. A re-quote becomes a new revision	At least 3 suppliers have quoted
3. Supplier choice	Quotation Comparison (Screen 3)	Purchase staff; admin or manager approves	Compare totals and terms, choose a supplier. A non-L1 choice needs a justification. Non-L1 or over budget goes for approval	Supplier finalized
4. Purchase order	Purchase Orders, PO details (Screen 4)	Purchase staff; finance confirms tax; admin or manager approves	Fill bill-to and ship-to, confirm GST (IGST or CGST + SGST), set QA fee terms, submit, approve, release	PO released (Rev 0)
5. BSNL QA testing	PO, Track QA (Screen 5)	Purchase staff	Record each lot the supplier offers to BSNL, then BSNL's result	All quantities passed

Stage	Screen and menu	Who acts	What the user does	Stage is done when
6. ITI QA payment	PO, Payment Status (Screen 6); also Payments and Reimbursements, QA Payments	Purchase or finance staff	When the PO milestone is due, record the payment made outside Ledgr; add the bank UTR now or later	QA fee fully paid
7. UTR and Release Report	UTR and Release Reports (Screen 7)	Purchase staff; admin or manager verifies RR	Record that the UTR was sent, the supplier's acknowledgement, and the Release Report with 4 checks	RR verified
8. Material dispatch	Material Dispatch: directory, then dispatch details (Screen 8)	Purchase staff	Record each dispatch (transporter, LR, e-way bill), update transit and delivery, link the ERP GRN number	All delivered and GRNs linked

The three tracks in stage 7 are independent: sending the UTR does not mark it acknowledged, and neither verifies the RR.

Roles and access

Admins, managers and Purchase-department staff can open the module. Only admins and managers can approve. The backend must enforce these checks itself, not just rely on the UI.

Role	How it is identified	Can do
Admin	role = admin	Everything, including all approvals and RR verification sign-off
Manager	role = manager	Everything, including all approvals and RR verification sign-off

Role	How it is identified	Can do
Purchase staff	department matches "purchas" or "procure" (case-insensitive)	Create and edit PRs, quotations, selections, PO drafts, QA, payments, UTR, RRs, dispatches; submit for approval. Keeps their own expense screens
Other employees	anything else	No access; the Procurement menu is hidden

Actions that need an admin or manager:

- Approve or reject a supplier selection that needs approval (non-L1 supplier or over budget)
- Approve or return a PO that is Pending Approval
- Complete RR verification after all four checks pass

Finance verification of PO tax (the `taxVerified` flag) is a checkbox today. Who may tick it is an open question (section 10).

Status models

Most statuses are **derived** from the records rather than set by hand. The backend should compute them on read, or recompute them whenever a child record changes. Only the statuses marked **stored** are written directly.

Purchase requirement (Screens 1 to 3)

Status	Label	Rule
draft	Draft	Stored. Saved without inviting suppliers
awaiting	Awaiting Quotations	No quotations yet
partial	Partially Received	Fewer quotations than invited suppliers, and fewer than 3
received	Quotations Received	All invited suppliers quoted, but fewer than 3
ready	Ready for Comparison	At least 3 suppliers quoted (MIN_QUOTES = 3)

Status	Label	Rule
evaluation	Under Evaluation	Selection is a draft or was rejected
approval	Pending Approval	Selection submitted
approved	Selection Approved	Selection approved
closed	Closed / Awarded	Stored, or the supplier is finalized

A quotation count uses only each supplier's latest revision.

Supplier selection (stored in `finalization.state`)

draft → submitted → approved or rejected → finalized. A selection that needs no approval (the L1 supplier, within budget) goes straight from draft to finalized.

Purchase order (stored in `po.status`)

Status	Label	Set by
draft	Draft	Created from the finalized quotation. Also set when an approver returns it (`returned = true`)
pending_approval	Pending Approval	Preparer submits
approved	Approved	Admin or manager approves
released	Released to Supplier	Release action. Creates revision 0

BSNL QA lot (stored per offer)

offered (Offered to BSNL), in_progress (QA In Progress), clarification (Clarification Required), retest (Retest Required), passed (QA Passed), failed (Failed / Rejected).

The PO-level QA status is derived: Pending PO Release, Awaiting QA Offer, QA In Progress, Clarification / Retest, QA Passed (all quantities certified), QA Failed.

ITI QA payment (derived per PO)

Status	Label	Rule
no_terms	No QA fee in PO	No QA terms or fee is 0
paid	Payment Recorded	Outstanding = 0
not_due	Not Yet Due	Trigger date not reached
partial	Partly Paid	Something paid, balance left
overdue	Overdue	Due date (trigger date + due days) has passed, nothing paid
due	Payment Due	Trigger reached, nothing paid, not overdue

The **Record Payment** action is allowed only in due, overdue or partial.

UTR, acknowledgement and Release Report: three independent tracks

Track	Values	Derived from
UTR submitted	No UTR yet, UTR Pending Submission, Partly Submitted, UTR Submitted	Payments that have a UTR vs UTR submission records
Supplier acknowledged	Not applicable yet, Ack Pending, Partly Acknowledged, Acknowledged	UTR submissions vs acknowledgement records
RR verified	RR Awaited, RR Received, Under Verification, Discrepancy, RR Verified	Release Report records

One track never completes another. Each Release Report also has its own stored status: received, under_review, verified, discrepancy.

Dispatch

Each dispatch has a stored status: dispatched, in_transit, delivered. The PO dispatch status is derived:

Status	Label	Rule
locked	Awaiting RR Verification	No verified RR and no dispatches
awaiting	Awaiting Dispatch	Verified RR, no dispatches
partial	Partially Dispatched	Dispatched qty < ordered qty
dispatched	Fully Dispatched	All dispatched, some still in transit
grn_pending	Delivered · GRN Pending	All delivered, some without an external GRN reference
completed	Delivered (GRN Linked)	All delivered and linked to a GRN

Data model

The frontend nests everything under one requirement (`PurchaseRequirement.po.qaPayments[]` and so on). The database should use one table per entity, keyed by `company_id`. Each list below becomes a child table. Field names match `src/purchases/types.ts`. Dates are `yyyy-mm-dd`; timestamps are ISO 8601.

Entity	Parent	Key fields
Supplier (master, read-only)	company	id, name, gstin, city, email, empaneled
Material (master, read-only)	company	code, name, category, unit, spec
PurchaseRequirement	company	id, prNo, project, requestDate, department, requestedBy, deliveryDate, costCenter, purpose, priority, budget, deliveryPeriod, paymentTerms, warranty, compliance, closingDate, closingTime, attachments[], status (draft or closed only)
RequirementItem (BoQ line)	requirement	id, materialCode, name, spec, qty, unit, location

Entity	Parent	Key fields
Invited supplier	requirement	supplierId
Quotation	requirement	id, supplierId, quoteNo, revision, quoteDate, validUntil, gstRate, discount, freight, other, hsn, deliveryPeriod, paymentTerms, warranty, compliance (full or partial), fileName, createdAt, createdBy
QuotationLine	quotation	itemId, rate
Finalization (selection)	requirement, 1:1	supplierId, reason, approvalRef, justification, document, state, savedAt, savedBy, decidedBy, decidedAt, decisionNote
PurchaseOrder	requirement, 1:1	poNo, issueDate, quotationId, supplierId, billTo, shipTo, deliveryDate, instructions, taxVerified, status, preparedBy, createdAt, submittedAt, decidedBy, decidedAt, decisionNote, returned, releasedAt, releasedBy
PoParty (bill-to, ship-to)	PO	name, address, stateCode, gstin, contact, phone
PoRevision	PO	rev, at, by, summary, total
QaPaymentTerms	PO, 1:1	fee, payer, payee, trigger (call_letter, qa_start or qa_passed), dueDays, clause
QaOffer (QA lot)	PO	id, lotNo, itemId, qty, callLetterNo, callDate, bsnlRef, testingLocation, supplierContact, notes, documents[], status, officer, expectedCompletion, remarks, certificate, updatedAt
QaPaymentRecord	PO	id, ref, amount, valueDate, mode, bank, voucherRef, utr (nullable), remarks, advice, recordedAt, recordedBy, utrAddedAt

Entity	Parent	Key fields
UtrSubmission	PO, linked to a payment	id, paymentId, utr, sharedAt, sharedBy, channel, recipientName, recipientEmail, reference, note, evidence, recordedAt
SupplierAck	PO, linked to a submission	id, submissionId, ackDate, contact, response, attachment, recordedBy, recordedAt
ReleaseReport	PO	id, rrNo, issuer, issueDate, receivedDate, offerIds[], qty, file, status, reviewer, note, uploadedBy, uploadedAt, verifiedBy, verifiedAt
RrCheck	release report	key (document, quantity, material or consignee), state (pending, passed or failed), note
Dispatch	PO	id, ref, lotNo, dispatchDate, transporter, vehicleNo, lrNo, ewayBill, ewayValidUntil, eta, origin, contact, documents[], status, deliveredDate, createdBy, createdAt
DispatchLine	dispatch	itemId, qty
GrnLink	dispatch, 1:1	ref, date, receivedQty, discrepancy, attachment, recordedBy, recordedAt
Event (audit log)	requirement or PO	at, title, detail, by, plus a `stream` column: requirement, po, payment or dispatch

UTR submissions and acknowledgements are separate tables from payments. A payment row with a UTR does not imply a submission row exists.

API endpoints per stage

These are suggested REST endpoints under `/api/purchases`. Every write returns the updated parent record (requirement or PO) with its derived statuses, so a screen can refresh from one response. `:rid` = requirement id, `:po` = PO number.

Masters and dashboard

Method	Path	Purpose
GET	/suppliers?q=	Supplier master (empaneled flag, GSTIN)
GET	/materials?q=	Material master for BoQ lines
GET	/dashboard	Counts, pending actions, recent activity

Stage 1 and 2: requirements and quotations

Method	Path	Purpose
GET	/requirements?status=&priority=&project=&q=&page=	Directory with derived status
GET	/requirements/:rid	Full requirement: items, suppliers, quotations, selection, PO summary, history
POST	/requirements	Create. Body includes `draft: true` to save without inviting. Server assigns prNo
PUT	/requirements/:rid	Edit a draft or add invited suppliers
POST	/requirements/:rid/quotations	Add a quotation. Server assigns the revision number
POST	/quotations/extract	Multipart file upload. Returns prefilled fields (supplier match, quote no, dates, rates, GST). Same engine as /expenses/extract

Stage 3: comparison and supplier selection

Method	Path	Purpose
GET	/requirements/:rid/comparison	Ranked quotations (L1, L2…) with totals, and the selection check for each supplier
PUT	/requirements/:rid/selection	Save selection as a draft
POST	/requirements/:rid/selection/submit	Submit for approval
POST	/requirements/:rid/selection/decision	Admin or manager. Body `{ approve, note }`
POST	/requirements/:rid/selection/finalize	Finalize the supplier and close the quotation round

Stage 4: purchase order

Method	Path	Purpose
GET	/purchase-orders?status=&q=&page=	PO directory with QA, payment, release and dispatch statuses
POST	/requirements/:rid/purchase-order	Create the draft PO from the finalized quotation. Idempotent: returns the existing PO
GET	/purchase-orders/:po	PO with computed lines, tax split, totals, amount in words
PUT	/purchase-orders/:po	Save draft: billTo, shipTo, deliveryDate, instructions, taxVerified, qaPayment terms
POST	/purchase-orders/:po/submit	Draft to Pending Approval
POST	/purchase-orders/:po/decision	Admin or manager. Approve, or return to Draft
POST	/purchase-orders/:po/release	Approved to Released. Creates revision 0
GET	/purchase-orders/:po/pdf	Printable PO (optional)

Stage 5: BSNL QA testing

Method	Path	Purpose
GET	/purchase-orders/:po/qa	Lots, per-item offered, passed and pending quantities
POST	/purchase-orders/:po/qa-offers	Record a QA offer (call letter). Server assigns lotNo
PATCH	/qa-offers/:id	Update result: status, officer, expected completion, remarks, certificate

Stage 6: ITI QA payment

Method	Path	Purpose
GET	/qa-payments?status=	QA Payments directory (Payments and Reimbursements screen)
GET	/purchase-orders/:po/qa-payment	Terms, trigger date, due date, outstanding, payments
POST	/purchase-orders/:po/qa-payments	Record a payment. Server assigns ref PAY-QA-YYYY-NNN. UTR may be null
PATCH	/qa-payments/:id/utr	Add the bank UTR later. Does not share it with the supplier

Stage 7: UTR submission and Release Report

Method	Path	Purpose
GET	/release-tracking?utr=&ack=&rr=	UTR and Release Reports directory
GET	/purchase-orders/:po/release	The three tracks, submissions, acks, RRs
POST	/purchase-orders/:po/utr-submissions	Record that a payment's UTR was sent to the supplier
POST	/utr-submissions/:id/acknowledgement	Record the supplier's acknowledgement

Method	Path	Purpose
POST	/purchase-orders/:po/release-reports	Record a received RR (multipart file)
PATCH	/release-reports/:id/checks/:key	Set one check to passed or failed, with a note
POST	/release-reports/:id/verify	Admin or manager sign-off. All 4 checks must be passed
POST	/release-reports/:id/discrepancy	Flag a discrepancy with a note
PUT	/release-reports/:id/file	Upload a corrected RR. Resets all checks

Stage 8: material dispatch

Method	Path	Purpose
GET	/dispatches?status=&q=	Material Dispatch directory, one row per PO
GET	/purchase-orders/:po/dispatch	Per-item ordered, released, dispatched, dispatchable; dispatch list
POST	/purchase-orders/:po/dispatches	Record a dispatch. Server assigns lotNo and ref DSP-YYYY-NNNN-NN
PATCH	/dispatches/:id/status	dispatched, in_transit or delivered, with date and note
PUT	/dispatches/:id/grn	Link the external ERP GRN reference

Shared

Method	Path	Purpose
POST	/attachments	Upload a file. Returns `{ id, name, url }` for use in other bodies
GET	/requirements/:rid/history?stream=	Audit log, newest first

Business rules and validations

The frontend checks these rules today. The backend must check them again and reject the request with field errors (section 8).

Quotations and selection

- A supplier's new quotation is revision + 1. Earlier revisions are kept, never overwritten. Comparison uses the latest revision only.
- Quotation total = Σ(qty × rate) − discount, plus GST on that amount, plus freight and other charges (charges are not taxed).
- Suppliers are ranked by total: L1 is the lowest.
- Choosing a non-L1 supplier needs a reason and a written justification (Policy Clause 4.2B).
- A non-L1 choice, or a total over the budget, needs admin or manager approval before it can be finalized.
- Comparison opens only when at least 3 suppliers have quoted (MIN_QUOTES = 3).

Purchase order

- Only one PO per requirement, created only after the supplier is finalized.
- Rates, discount, GST rate, freight, charges and commercial terms come from the finalized quotation. They are not editable on the PO.
- Discount is spread over lines by gross value, and GST is worked per line. Rounding paise go on the last line, so the PO total equals the quotation total exactly. Any variance is shown as an error.
- Tax mode: supplier state (first 2 digits of GSTIN) vs ship-to state. Same state = CGST + SGST at half the rate each. Different state = IGST. Unknown if either is missing.
- A PO can be submitted only when tax mode is known and finance has ticked `taxVerified`.
- Changing the ship-to state clears `taxVerified`, so finance must confirm again.
- Released POs are not edited. Amendments appear only as revision history (Rev 0 = original release).

QA testing

- QA offers are allowed only on a released PO.

- Quantity offered must not exceed what is left to offer on that BoQ line.
- Call letter date must not be before the PO release date. Call letter attachment is required.
- A passed or failed result needs the BSNL officer and the signed certificate. Clarification, retest and failed need remarks.

QA payment

- Trigger date comes from the PO terms: the first call letter date, the first lot that started testing, or the date the last lot passed (all quantities passed).
- Due date = trigger date + due days.
- A payment can be recorded only when the status is due, overdue or partial. Amount must not exceed the outstanding balance.
- UTR format: 16 to 22 letters or digits. The UTR can be added later; adding it does not mean it was shared with the supplier.

UTR, acknowledgement and Release Report

- A UTR submission needs a payment that has a UTR. Sent date must be between the payment value date and today. A recipient name or a valid email is required.
- An acknowledgement needs an existing UTR submission.
- Recording an RR sets it to received with 4 pending checks. Setting any check moves it to under review.
- Verify is allowed only for an admin or manager, and only when all 4 checks are passed.
- A corrected RR file resets all checks and clears the verification.
- None of these actions changes the status of another track.

Dispatch

- Dispatch is allowed only against quantities released by verified RRs, minus what is already dispatched, per BoQ line.
- Dispatch date: not in the future and not before the latest RR verification. ETA not before the dispatch date.
- Transporter, LR number and at least one document (delivery challan) are required. An e-way bill number, if given, has 12 digits.
- Delivered date must be between the dispatch date and today.

- GRN link: GRN number and date required; date not before delivery. Received quantity must not exceed the dispatched quantity. A shortfall needs a discrepancy note.
- Ledgr never creates the GRN; it only stores the ERP reference.

Cross-cutting conventions

The purchase APIs follow the same conventions as the rest of Ledgr, so the existing client in `src/api/client.ts` works without changes.

Auth and scope

- Bearer token in `Authorization`. A 401 logs the user out.
- Every record is scoped to the user's company. The role checks in section 3 are enforced on the server.

Response and errors

```
{ "success": true, "data": { }, "message": "optional" }
```

- A failed request returns a non-2xx status, or `success: false`.
- Field errors go in `errors`, keyed by the body field name, for example `{ "success": false, "message": "Please fix the highlighted fields.", "errors": { "qty": ["Only 1,000 Km are left to offer on this line."] } }`. The client shows each message under its field.
- Use 403 for a role failure and 409 for a wrong state (for example releasing a PO that is not approved).

Numbering (assigned by the server)

Record	Format	Example
Requirement	PR-YYYY-NNN	PR-2026-007
Purchase order	PO-YYYY-NNNN	PO-2026-0841
QA payment	PAY-QA-YYYY-NNN	PAY-QA-2026-001
Dispatch	DSP-YYYY-NNNN-NN (PO number + lot)	DSP-2026-0840-01
Quotation revision	1, 2, 3 per supplier	R2

Record	Format	Example
QA lot, dispatch lot	1, 2, 3 per PO	Lot 1

Numbers must be unique per company and generated inside a transaction, so two users never get the same number.

Files

- Uploads: quotation files, requirement attachments, selection document, call letters, BSNL certificates, payment advice, UTR evidence, acknowledgement attachments, RR files, dispatch documents, GRN attachment.
- The frontend stores only file names today. The backend should return `{ id, name, url }` and accept the id in later requests.

Audit log

- Every write adds an event `{ at, title, detail, by }`. The server sets `at` and `by` from the logged-in user, never from the request body.
- Events are kept in four streams: requirement history, PO history, payment and release log, and dispatch log. Each screen's History tab reads one stream, newest first.
- Records are never deleted. Corrections are new records (a new quotation revision, a replaced RR file) with their own event.

Frontend integration map

Each store function in `src/purchases/` becomes one API call; the screens do not change. Today these functions write to browser storage (`ledgr.purchases.<companyId>`). After integration they call the endpoint and replace the cached record with the response. The derived-status functions (`reqStatus`, `qaPaymentState`, `releaseState`, `dispatchState`) can stay on the client as long as the API returns the child records.

Frontend function (file)	Endpoint
useRequirements (requirements.ts)	GET /requirements, GET /requirements/:rid
createRequirement (requirements.ts)	POST /requirements
addQuotation (requirements.ts)	POST /requirements/:rid/quotations

Frontend function (file)	Endpoint
extractQuotation (quotationExtract.ts)	POST /quotations/extract
saveSelection, submit = false (requirements.ts)	PUT /requirements/:rid/selection
saveSelection, submit = true (requirements.ts)	POST /requirements/:rid/selection/submit
decideSelection (requirements.ts)	POST /requirements/:rid/selection/decision
finalizeSelection (requirements.ts)	POST /requirements/:rid/selection/finalize
ensurePurchaseOrder (purchaseOrders.ts)	POST /requirements/:rid/purchase-order
savePoDraft (purchaseOrders.ts)	PUT /purchase-orders/:po
submitPo (purchaseOrders.ts)	POST /purchase-orders/:po/submit
decidePo (purchaseOrders.ts)	POST /purchase-orders/:po/decision
releasePo (purchaseOrders.ts)	POST /purchase-orders/:po/release
recordQaOffer (qa.ts)	POST /purchase-orders/:po/qa-offers
updateQaResult (qa.ts)	PATCH /qa-offers/:id
recordQaPayment (qaPayment.ts)	POST /purchase-orders/:po/qa-payments
addPaymentUtr (qaPayment.ts)	PATCH /qa-payments/:id/utr
recordUtrSubmission (release.ts)	POST /purchase-orders/:po/utr-submissions
recordSupplierAck (release.ts)	POST /utr-submissions/:id/acknowledgement
recordReleaseReport (release.ts)	POST /purchase-orders/:po/release-reports
setRrCheck (release.ts)	PATCH /release-reports/:id/checks/:key
completeRrVerification (release.ts)	POST /release-reports/:id/verify
flagRrDiscrepancy (release.ts)	POST /release-reports/:id/discrepancy
replaceRrFile (release.ts)	PUT /release-reports/:id/file
recordDispatch (dispatch.ts)	POST /purchase-orders/:po/dispatches

Frontend function (file)	Endpoint
updateDispatchStatus (dispatch.ts)	PATCH /dispatches/:id/status
recordGrnReference (dispatch.ts)	PUT /dispatches/:id/grn
SUPPLIERS, MATERIALS (requirementsData.ts)	GET /suppliers, GET /materials
SAMPLE_RECORDS, SAMPLE_ACTIONS (mock.ts)	GET /dashboard

The sample data in `requirementsData.ts` (PR-2026-001 to PR-2026-007) covers every stage. It can be used as seed data for backend testing.

Open questions

These decisions are not settled in the frontend and should be agreed before the APIs are built.

- [ ] Who may tick "Finance verified" for PO tax: any purchase user, or only a finance role?
- [ ] Should the supplier and material masters come from the external ERP, or be kept in Ledgr?
- [ ] Is the quotation extraction service the same as expense receipt extraction, or a separate model for quotations?
- [ ] Should the GRN reference be looked up from the ERP automatically, or always entered by hand?
- [ ] Should the server email the PO and the UTR to the supplier, or only record that the team sent them?
- [ ] Is a full PO amendment workflow needed later, or is revision history enough?
- [ ] Does selection approval need levels by amount (for example, above a threshold goes to admin only)?
- [ ] Is there one QA fee per PO, or can a PO carry several QA milestones?