Phase 6 — User Interface Development (VendorConnect)

Title Page

- **Project:** VendorConnect — Vendor & Purchase Order Management
- **Phase:** 6 — User Interface Development (Reports & Dashboards)
- **Author:** [Your Name]
- **Date:** [Submission Date]

Objective

Build a modern, usable UI for VendorConnect focused on **reporting and dashboards** (no LWC). Deliverables include:

- Vendor performance report
- Active Purchase Orders report
- Expired Purchase Orders report
- VendorConnect Dashboard combining the three reports
- Scheduled dashboard refresh and access control

Scope

This document covers end-to-end steps to create Salesforce-native UI components using **Reports** and **Dashboards** (no code). It includes:

- Prerequisites & sample data
- Detailed step-by-step process flow (with decision points)
- Exact clicks and filters to build each report
- Dashboard assembly and scheduling
- Testing / UAT checklist and acceptance criteria
- Suggested screenshots to include in your submission

Prerequisites

1. You must have **System Administrator** or Report Builder permissions.

2. Custom objects exist: Vendor__c and Purchase_Order__c (API names may vary).
3. Purchase_Order__c must have **Allow Reports = true** (Object Manager → Purchase Order → Edit → Allow Reports).
4. Sample data should be present (see Sample Data section). If empty, create records using the UI or Execute Anonymous.

Sample Data (use this to populate UI before screenshots)

Vendors

Vendor Name	Rating__c	Compliance_Status__c	Delivery_SLA__c
Vendor A	A	Compliant	7
Vendor B	B	Pending	14
Vendor C	C	Non-Compliant	21

Purchase Orders

PO Name	Vendor Name	Amount__c	Status__c	Start_Date__c	End_Date__c	Expected_Delivery__c
PO-001	Vendor A	1000	Completed	2025-09-01	2025-09-05	2025-09-05
PO-002	Vendor A	2000	Completed	2025-09-06	2025-09-15	2025-09-15
PO-003	Vendor B	1500	Submitted	2025-09-02	2025-09-10	2025-09-10
PO-004	Vendor B	2500	Completed	2025-09-03	2025-09-12	2025-09-12
PO-005	Vendor C	3000	Draft	2025-09-07	2025-09-17	2025-09-17

Process flow (high-level)

1. **Prepare**: confirm objects exist, Allow Reports enabled, sample data present.
2. **Create Custom Report Type (CRT)** if needed (Vendor with POs, Purchase Orders).
3. **Build Reports** (Vendor Performance, Active POs, Expired POs).
4. **Assemble Dashboard** using the saved reports as components.
5. **Set Dashboard refresh** schedule and share folder access.
6. **Test & Validate** (functional + data checks).
7. **Sign-off** and include screenshots in final documentation.

The detailed step-by-step process with decision points is below.

Detailed step-by-step (process flow with decisions)

Step 1 — Prepare the org

1. Login as Admin.
2. Go to **Setup → Object Manager → Purchase Order**.
    - Confirm **Allow Reports** is checked. If not: **Edit** → enable **Allow Reports** → **Save**.
3. Ensure Vendor__c and Purchase_Order__c have the fields used in reports: Amount__c, Status__c, Expected_Delivery__c, etc.
4. Load sample data (see Sample Data table) if the reports would otherwise be empty.

Decision A: If Allow Reports was disabled, enable it and then continue. If the Purchase Order object does not exist, create it first (not covered here).

Step 2 — Create Custom Report Types (CRT)

Rationale: CRTs allow easy reporting across custom objects (Vendor ↔ Purchase Order).

1. Setup → Quick Find → **Report Types** → **New Custom Report Type**.
2. **Primary Object**: Vendor__c.
    - **Report Type Label**: Vendors with Purchase Orders
    - **Report Type Name (API)**: Vendors_with_Purchase_Orders (auto-generated)
    - **Store in Category**: choose or create Vendor Reports
    - **Deployment Status**: Deployed → Next.
3. Define related object: **Purchase_Order__c** (select relationship *Each "A" record must have at least one related "B" record*). → Save.
4. (Optional) Create another CRT with Primary Object **Purchase_Order__c** and label Purchase Orders if you prefer a purchase-order centric report type.

Decision B: If you prefer tabular PO-only reports (no vendor grouping), you can skip CRT and create a Report directly on Purchase Orders (Purchase_Order__c must be reportable).

Step 3 — Report 1: Vendor Performance Report

Purpose: Show vendors with count of completed POs and total completed amount.

1. App Launcher → Reports → **New Report**.
2. Select **Vendors with Purchase Orders** (CRT) → Start Report.
3. Format: **Summary** (so you can group by vendor).
4. Columns to include (drag from left fields):
    - Vendor Name (Vendor__c.Name)
    - Rating__c (Vendor field)
    - COUNT of Purchase Orders (use the report *Rows* and summarize it as a count)

- SUM of Amount__c (on Purchase_Order__c → click column → Summarize → SUM)

5. Filters:
   - **Status__c = Completed**
   - (Optional) Date range filter on Start_Date__c or End_Date__c as needed.
6. Run → verify numbers match expected sample totals (Vendor A total 3000, Vendor B total 2500).
7. Save → **Report Name**: Vendor Performance Report → Folder: Vendor Reports.

Step 4 — Report 2: Active Purchase Orders

Purpose: List POs that are currently active (Submitted or Approved).

1. Reports → New Report → select **Purchase Orders** report type (or CRT Purchase Orders).
2. Keep **Tabular** or **Summary**; for a list choose **Tabular**.
3. Columns:
   - PO Name (Name)
   - Vendor Name (Vendor__r.Name)
   - Status__c
   - Amount__c
   - Expected_Delivery__c
4. Filters: remove defaults. Add:
   - **Status__c = Submitted OR Status__c = Approved**
5. Run to verify results → Save as **Active Purchase Orders Report** in Vendor Reports.

Step 5 — Report 3: Expired Purchase Orders

Purpose: Find POs that should have been delivered but are not complete.

1. Reports → New Report → **Purchase Orders**.
2. Columns: PO Name, Vendor Name, Status__c, Expected_Delivery__c.
3. Filters:
   - **Expected_Delivery__c < TODAY**
   - **Status__c != Completed**
4. Run → verify rows are those late POs → Save as **Expired Purchase Orders Report**.

Step 6 — Create Dashboard

1. App Launcher → Dashboards → **New Dashboard**.
2. **Name:** VendorConnect Dashboard.

3. Folder: Vendor Reports → Create.
4. Add components (click **+ Component**):
    - Component 1: **Vendor Performance Report** → Component Type: *Table* or *Bar Chart* (if summarizing spend). Title: Vendor Performance.
    - Component 2: **Active Purchase Orders Report** → Component Type: *Table*. Title: Active Purchase Orders.
    - Component 3: **Expired Purchase Orders Report** → Component Type: *Table* or *Metric* for count. Title: Expired Purchase Orders.
5. Arrange components on the dashboard canvas → Save.

Decision C: If dashboard needs visual metrics for executive view, change components to charts (bar/pie/metric) instead of tables.

Step 7 — Schedule Dashboard Refresh & Share

1. Open the dashboard → **Subscribe** or **Schedule Refresh** (depending on your Salesforce edition and permissions).
2. Set schedule: Daily at 7:00 AM for Procurement team (example).
3. Share dashboard folder Vendor Reports with user groups or roles (Setup → Folders or from Dashboard access settings).

Step 8 — Test & Validate (UAT)

Run this checklist with sample user accounts (Procurement Officer, Manager):

Functional tests

- Vendor Performance Report shows correct counts and sums for Status = Completed.
- Active PO report contains only Submitted/Approved POs.
- Expired PO report only lists POs with Expected_Delivery__c in the past and not Completed.
- Dashboard shows all components and refreshes with scheduled updates.

Security & Access

- Profiles assigned to VendorConnect app can open reports and dashboard.
- Folder sharing allows target users to view dashboard but not necessarily edit.

Data validation

- Totals match manual calculations from sample data.

Step 9 — Sign-off

1. Get stakeholder sign-off (Procurement Manager).
2. Document screenshots and finalize Phase 6 deliverables.

Screenshots to collect (file name suggestions)

1. 01_Object_AllowReports.png — Purchase Order object details showing Allow Reports checked.
2. 02_Create_CRT.png — New Custom Report Type screen (Vendors with Purchase Orders).
3. 03_VendorPerf_ReportBuilder.png — Vendor Performance report in builder (columns + summarize).
4. 04_ActivePO_Report.png — Active Purchase Orders report run view.
5. 05_ExpiredPO_Report.png — Expired Purchase Orders report run view.
6. 06_Dashboard_EditView.png — Dashboard edit mode with components placed.
7. 07_Dashboard_View.png — Final dashboard (run view).
8. 08_Dashboard_Schedule.png — Schedule/Subscribe settings.
9. 09_Access_Sharing.png — Folder sharing settings for Vendor Reports.

Deliverables (what to submit)

- Phase 6 Documentation (this document)
- Saved reports: Vendor Performance Report, Active Purchase Orders Report, Expired Purchase Orders Report
- Dashboard: VendorConnect Dashboard (saved in Vendor Reports folder)
- Screenshots (9 images listed above)
- UAT sign-off (email or document)

Appendix — Useful SOQL checks (Developer Console)

- Verify completed PO totals for Vendor A:

```
SELECT Vendor__c, COUNT(Id), SUM(Amount__c)
FROM Purchase_Order__c
WHERE Status__c = 'Completed' AND Vendor__r.Name = 'Vendor A'
GROUP BY Vendor__c
```

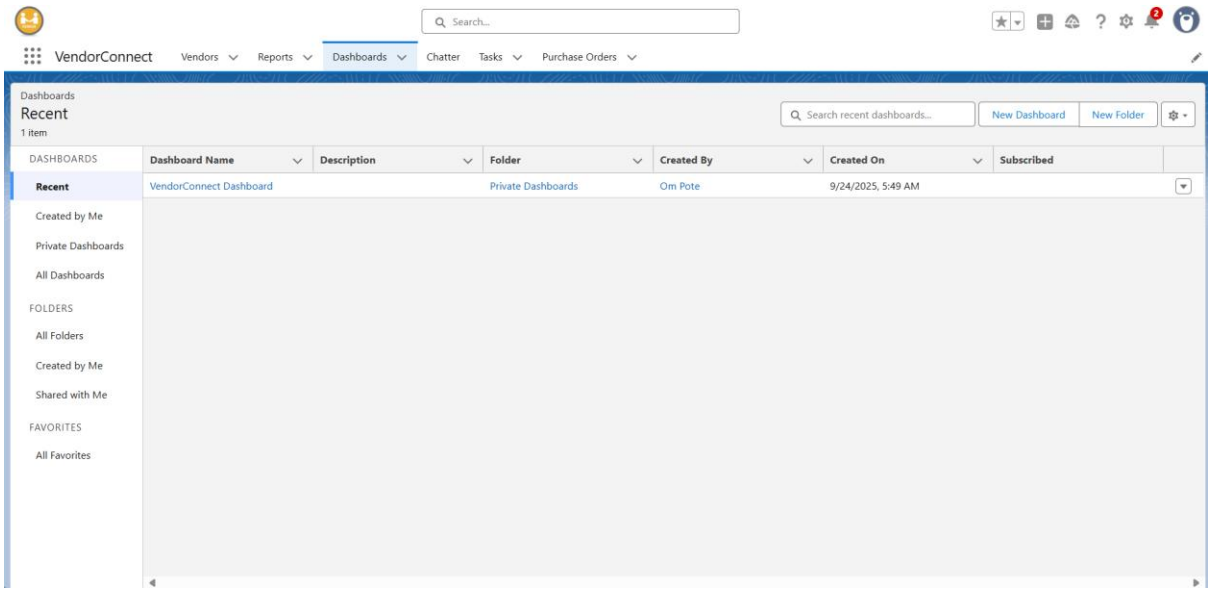
- Verify active POs:

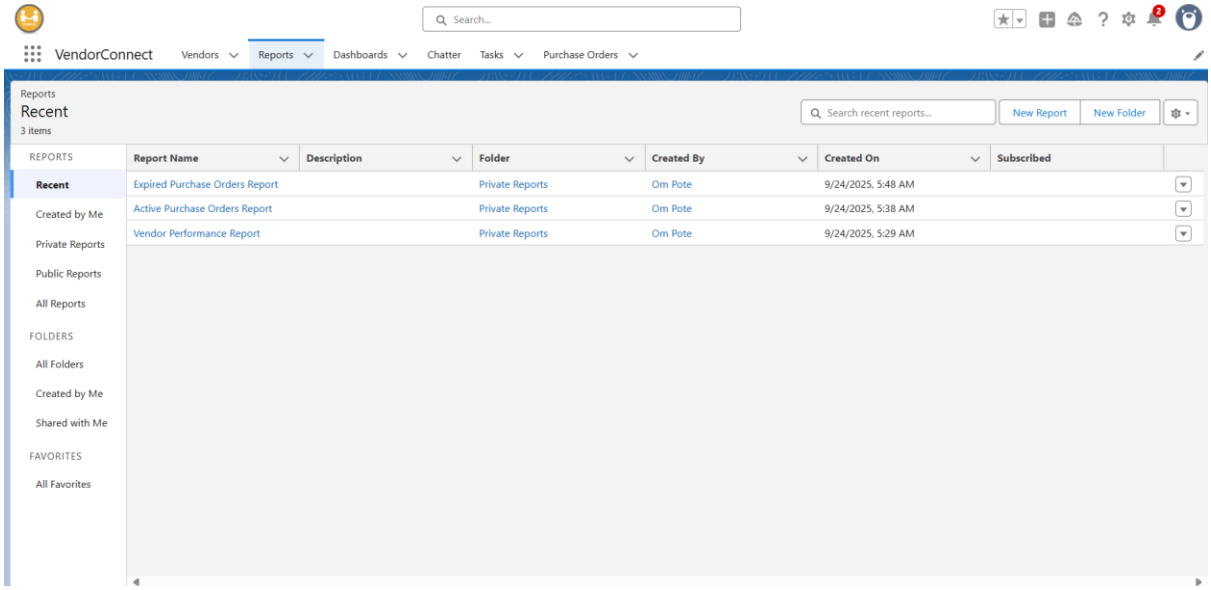
```
SELECT Id, Name, Vendor__r.Name, Status__c, Amount__c, Expected_Delivery__c
FROM Purchase_Order__c
WHERE Status__c IN ('Submitted','Approved')
```

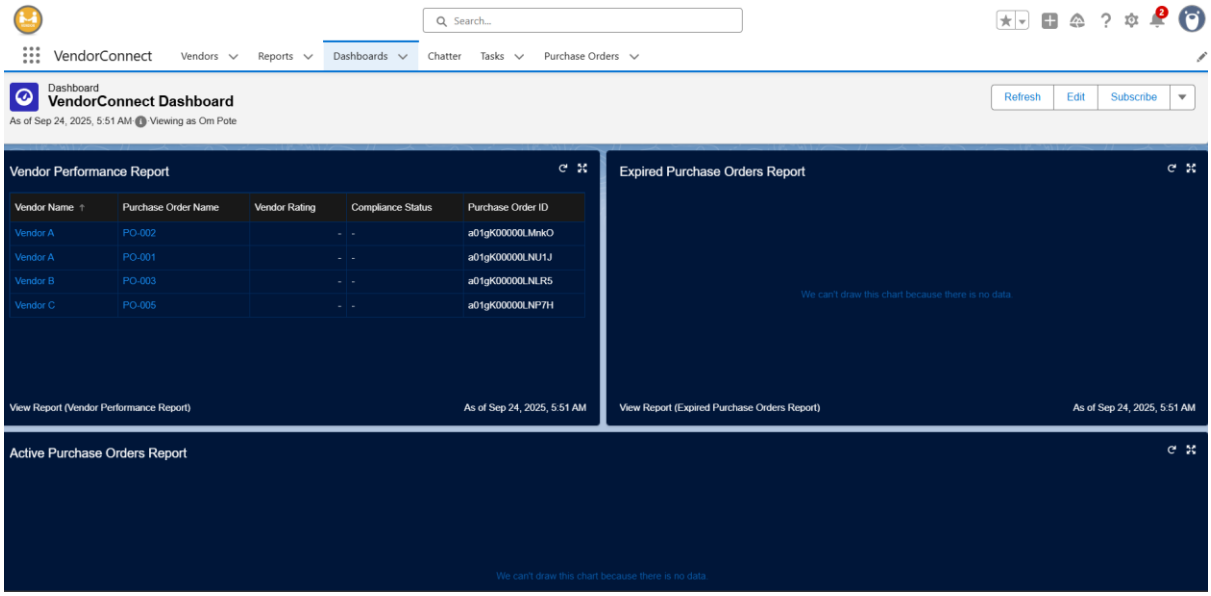
Notes & Next Steps

- Phase 6 is complete once the Dashboard and Reports are signed off.
- **Phase 7 (Testing & Validation)**: prepare test cases, run tests, and fix any data or security gaps.

Screenshots:-







End of Phase 6 documentation — VendorConnect