training - Bug #2814

User Ledger Report with Debit, Credit & Running Balance

07/08/2026 09:07 AM - Prashant Jain

Status:	New	**Start date:**	07/08/2026
Priority:	Normal	**Due date:**	
Assignee:	Shreya Agarwal	**% Done:**	0%
Category:		**Estimated time:**	8.00 hours
Target version:		**Spent time:**	31.00 hours

Description

Description:

Develop a User Ledger Report to provide a complete transaction history for each user, including all debit and credit entries with a running balance.

Requirements
Filters
Date Range (From Date – To Date)
User (Optional)
If no user_id is selected, display ledger data for all users.
If user_id is provided, display the ledger only for the selected user.
Station (Optional)
Transaction Type (Debit/Credit/All)
Voucher Number (Optional)
Status (Optional)
Report Columns
Transaction Date
Voucher No.
User ID
User Name
Station Name
Transaction Type (Debit/Credit)
Debit Amount
Credit Amount
Opening Balance
Running Balance
Closing Balance
Reference No.
Description/Remarks
Created By
Report Logic
Calculate Opening Balance based on the balance before the selected date range.
Display all debit and credit transactions in chronological order.
Maintain a Running Balance for each transaction.
Calculate Closing Balance at the end of the selected period.
Support consolidated reports for all users as well as individual user ledger reports.
API Requirements
Support filtering by:
from_date
to_date
user_id
station_id
transaction_type
voucher_no
status
Pagination support.
Sorting by transaction date.
Export to Excel and CSV.
Expected Outcome

Users should be able to view a complete ledger containing all debit and credit transactions, voucher details, opening balance, running balance, and closing balance. The report should support both consolidated and user-specific views and be suitable for

finance, audit, and reconciliation purposes.

History

#1 - 07/08/2026 09:10 AM - Yashaditya Singh

I implemented and updated the export functionality, reviewed the CSV and Excel export services, and modified the export API to return ledger data in JSON format instead of CSV based on the new requirement. I also verified the service, repository, and controller flow to ensure the export endpoint correctly returns all filtered ledger records while keeping the existing Excel export functionality unchanged. Additionally, I tested the API responses and validated the end-to-end data retrieval flow for the User Ledger Report.

#2 - 07/08/2026 09:26 AM - Shreya Agarwal

Today I implemented the complete User Ledger page for the Expenses module, following the existing UI patterns from Claims, Advances, and Approvals pages. I created the page component with full filtering support including date range, status, transaction type, and employee filters, along with proper pagination and loading states. I added the necessary route configuration in rolebasedroute.tsx and integrated the page into the sidebar navigation under the Expenses section with the reporting.employee.user_ledger.view permission. The summary cards were removed as requested, and the employee name display was fixed to show the username directly instead of using the displayEmployeeName function. I also implemented CSV export functionality with proper error handling and created the required API service and type definitions for the ledger data. All changes are consistent with the existing codebase and maintain the same UI/UX patterns as the other expense pages.
some changes need to be test which is in progress
also I try to understand the fundamentals of GraphQL and explored how it is integrated into React applications using Apollo Client. I tried to understood the concepts of queries
also I researched the getPartnerPayoutProcessPartnershipdetail API to understand where the data comes from and how it is stored, analyzing the table structures and data flow for partner payout .

#3 - 07/09/2026 09:03 AM - Shreya Agarwal

implementation of the User Ledger page for the Expenses module, following the existing UI pattern of Claims, Advances, and Approvals pages.
Key Accomplishments:
Created complete User Ledger page with filters (Date range, Status, Transaction Type, User), summary cards (Opening Balance, Total Expenses, Total Received, Closing Balance), and transaction details table with proper styling.
Integrated API endpoints - Main data fetch (/user-ledger) with proper parameter handling (userid instead of employeeUserId), and export functionality (/export) with JSON to CSV conversion.
Customized UI elements - Renamed columns (Debit → Received, Credit → Expenses), added arrow icons, and set default values (0.00).
Fixed multiple issues - Default user selection (first employee auto-selected), status filter default to "All Statuses", pagination handling, and resolved 400 error by preventing API calls without userId.
Implemented export functionality - Successfully integrated the export endpoint, converting JSON response to CSV format with proper headers and data mapping for all transaction fields.
Added permission constants - Created and integrated view and export permissions in the expense permissions file for proper access control.
and did some extra changes which is needed for user-ledger according to requirements.

#4 - 07/09/2026 09:44 AM - Yashaditya Singh

I worked on the Employee User Ledger report by updating the filtering logic to support vouchers based on both party_user_id and approved employee advances created or posted by the selected user. I modified the repository queries to return debit entries, added station details in the response, removed the employee cash ledger restriction to include all relevant ledger entries, and tested the API with multiple users. I also validated the returned data against database records, investigated missing voucher entries by comparing API results with accounting vouchers and voucher lines, and continued debugging the remaining discrepancies in the ledger report.