

training - Bug #3284

UI Enhancement for some pages

10/09/2026 06:18 AM - Shreya Agarwal

Status:	New	Start date:	10/09/2026
Priority:	Normal	Due date:	
Assignee:	Shreya Agarwal	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:		Spent time:	36.00 hours
Description			
On the Partner Payout Process page the "Summary by Partner" table displays several columns that are not required UI:			

History

#1 - 10/09/2026 09:17 AM - Shreya Agarwal

Today, worked on the Summary by Partner table on the Partner Payout Process page was enhanced as part of the ongoing UI/UX improvement initiative.

The primary focus of this update was to streamline the table layout by removing unnecessary or redundant data columns that were cluttering the view and not adding significant value to the user's payout summary workflow. This was done to improve readability, reduce visual noise, and make the most important information more prominent for the user.

The changes were implemented in the file PartnerPayoutProcessPage.tsx, where both the table header and the corresponding data rows were updated to reflect the removal of the redundant columns. Additionally, the column span used for the table's loading state and empty state was adjusted from 13 to 11 to ensure proper alignment and consistent rendering across all table states.

It is important to note that despite the reduction in visible columns, all remaining data continues to be rendered dynamically.

also Today, I worked on refining and finalizing the UI/UX design for the Charging Sessions section of our main dashboard. I gathered and evaluated multiple design suggestions to ensure complete visual consistency with our new brand theme and color palette. Key highlights include optimizing the Top 5 Summary KPI cards and the Session Breakdown section to match the modern card layout, while keeping the Session Distribution Donut Chart logic completely intact as required. Additionally, I aligned the frontend visual requirements directly with our API endpoint capabilities (/reporting/tenant/summary), clarifying how metric counts, percentage deltas, and progress indicators should be structured for dynamic integration without unnecessary backend dependencies or extra UI elements.

#2 - 10/12/2026 09:17 AM - Shreya Agarwal

Today enhancements were made to the "Add partner..." dropdown within the Partner Receivables section of the Edit Station Selling Price modal The width of the dropdown was increased from 240px to 320px to provide a wider, more comfortable layout. Each partner's businessName was dynamically mapped from the existing GET /api/v1/reporting/partners API response and bound into the dropdown options alongside their partnerName, without any hardcoding. The options were formatted to cleanly present the partner name with their business name as secondary detail, enabling users to search and filter options seamlessly by either name. Additionally, a TypeScript compilation issue (TS2741) was resolved by properly typing the fallback object during new partner additions, and the dropdown menu's placement was anchored to open downward (bottom-start) with flip detection disabled and a constrained maximum height, preventing the menu from overlapping the modal header.

Also , updated the Partner Received page- to remove the three columns from the main data table: #Received, Stations, and Settled. The corresponding table header cells (TableCell), row data cells (row.advanceCount, row.stationCount, and row.totalReceived), and the table colSpan attributes for the empty state and expandable rows were cleanly removed and adjusted from 10 to 7 columns.

Also , Today's task focused on standardizing the numerical and currency representations within the Station Details Panel (/charging-station-management) to follow the Indian Numbering System (en-IN), referencing the formatting methodology already implemented on the Dashboard , the currency helper functions formatCurrency and formatCurrencyWithThreeDecimals were updated to utilize Intl.NumberFormat('en-IN'), ensuring dynamic currency code handling and authentic Indian comma separation (lakhs and crores) with 2 and 3 decimal precisions respectively. Additionally, the energy calculation helpers truncateToTwoDecimals and roundToThreeDecimals were updated to format both the top-level Energy Delivered (kWh) and per-charger energy delivered values using toLocaleString('en-IN') with appropriate fraction limits, ensuring numbers equal to or exceeding 1,000 dynamically display Indian grouping commas.

#3 - 10/13/2026 09:18 AM - Shreya Agarwal

Today, updated the Partner Received table on the /partnerships/partner-advance page by adding a new column titled Due immediately following the Balance column. The value in this column is dynamically mapped from each partner item's unallocatedBalance provided by the API response, ensuring zero hardcoding. Conditional formatting was implemented such that when the unallocated balance is less than zero (< 0), the amount is displayed in red font, and when it is greater than zero (> 0), it appears in green font, with zero balances remaining in neutral color. To satisfy the formatting criteria, the currency is rendered as an absolute value (Math.abs), ensuring no positive (+) or negative (-) signs are shown alongside the amount. In addition, the table's empty state and expanded row colSpan attributes were updated from 7 to 8 to maintain proper alignment and table responsiveness. Also more changes which is needed in some page are also done .

#4 - 10/14/2026 09:10 AM - Shreya Agarwal

In the Document Verification module (/document-verification) dynamic alphabetical sorting was implemented across all filter dropdowns in both the Pending Verification and All Documents tabs. In

DocumentVerification.tsx

, the charging stations fetched dynamically from the backend API were sorted alphabetically (A–Z) using natural case-insensitive comparison (`localeCompare`), ensuring proper ordering without altering any API bindings. Subsequently, the Document Type and Status filter options were updated to sort their options alphabetically in real time based on their localized display labels while keeping the default top options ("All Types" and "All Statuses") intact. All underlying enum keys and API parameter bindings were strictly preserved with zero hardcoded values, and the entire implementation was verified to compile cleanly with no TypeScript errors.

Also some bug are fixed in ui .