

Sprint Summary

#	Item	Area	Layer	Status
1	Base SKU Name Update	SKU Master	DB · UI · Hook	✓ Done
2	Warehouse Qty Fix	Inventory	Edge Fn · UI	✓ Done
3	Packaging Stock from Tally	Inventory	Edge Fn · Cron	✓ Done
4	Production Lead Time Editable	SKU Master	UI	✓ Done
5	Shipping Price Calculation	Orders	Lib · UI	✓ Done
6	Shipping Weight in SKU Table	SKU Master	DB · UI · Queries	✓ Done
7	Remove Channel Dropdown	SKU Master	UI	✓ Done
8	Calendar on Order Card	Orders	UI Component	✓ Done
9	0% Fill Rate Fix	Orders	Business Logic	✓ Done

Detailed Item Breakdown

#1 Base SKU Name Update SKU Master · DB · Hook - New <code>shelf_life</code> table keyed by <code>base_sku_name</code> (default 9 months) introduced alongside the custom hook <code>useShelfLife</code> for clean cross-component data access. - SKU grouping now uses <code>base_sku_name</code> as the parent key — multiple variants share one shelf-life definition, removing per-SKU duplication. - Expanded SKU card surfaces an editable shelf-life field. Drives auto expiry on all batches: <code>expiry = mfg_date + shelfMonths</code>. - Eliminates manual expiry entry errors and ensures every batch record stays consistent with the master shelf-life setting.	#2 Warehouse Qty of SKUs Fix Inventory · Edge Fn · UI - Resolved the long-standing discrepancy between the Warehouse tile count and the Batches modal figure that was causing ops confusion. - Tile now sourced exclusively from <code>skus.current_stock</code>, written by <code>tally-fetch-batch-stock</code> summing Tally batch closing stocks including <code>OPENING-2526</code>. - Fetch scoped strictly to the Finished Goods godown — negative stocks and duplicate godown entries no longer inflate or deflate the total. - Batches modal continues listing dispatched batches from <code>order_items</code>, making both views internally consistent and reconcilable.
#3 Packaging Stock from Tally Inventory · Edge Fn · Cron - New edge function <code>tally-fetch-packaging-stock</code> calls Tally's Stock Summary API, parses the response, and maps each item to <code>packaging_materials.tally_name</code>. - Resolved <code>current_stock</code> values are upserted in Supabase on every run — no manual data entry required. - Wired into the existing cron scheduler to run automatically on the same cadence as the SKU batch sync. - Packaging planners now see live Tally stock without opening Tally or maintaining a parallel spreadsheet.	#4 Production Lead Time Editable SKU Master · UI - Replaced the static read-only lead-time label in <code>src/pages/SKUs.tsx</code> with an inline number input paired with a Save button. - Follows the established <code>shelfLifeEdits</code> pattern — <code>leadTimeEdits</code> state tracks pending changes per SKU before committing. - On save, the handler writes <code>production_lead_time_days</code> directly to the <code>skus</code> row via a Supabase update call. - Ops team can now correct lead times in-app instantly — no developer DB edit or deployment needed.
#5 Shipping Price Calculation Orders · Lib · UI - New module <code>lib/shippingCalc.ts</code> encapsulates the full resolution pipeline: <code>weight × pieces → carrier weight slab → zone (A/B/C from party <code>region_zone</code>) → price</code>. - Order cards surface estimated freight alongside order totals, giving ops a pre-dispatch cost view before confirming. - Dispatch flow uses the same calculation to automatically highlight the cheapest carrier for each order. - Replaces manual carrier rate lookup — significant time saving on high-volume dispatch days.	#6 Shipping Weight in SKU Table SKU Master · DB · Queries - Added <code>weight_grams</code> and <code>pieces_per_unit</code> to the SKU master; both fields are surfaced and inline-editable in <code>SKUs.tsx</code>. - Fields joined into order/SKU queries in <code>useSupabaseData.ts</code> and <code>usePaginatedOrders.ts</code> so shipping math always has correct per-SKU inputs. - Fixed spurious 'missing weight' console warning that fired on async load even when data was present. - Provides the data foundation #5 (Shipping Calc) depends on — both items co-delivered in this sprint.
#7 Remove Channel Dropdown SKU Master · UI - Removed per-SKU <code>channel_tag</code> selector from the SKU master form and card view entirely. - Channel attribution is now driven exclusively at the party/order level, preventing conflicting tags between SKU and order. - Simplifies SKU creation workflow and eliminates the root cause of incorrect channel reporting in dashboards. - No DB migration required — column retained for legacy data; only the UI entry point has been removed.	#8 Calendar on Order Card Orders · UI Component - Replaced the plain-text Delivery by date field on <code>OrderCard.tsx</code> with an interactive shadcn <code>Calendar</code> inside a <code>Popover</code>. - Workflow: click date chip → calendar opens → select date → popover closes → <code>delivery_date</code> patches Supabase immediately. - Eliminates date format errors from free-text entry; dramatically faster on mobile where typing dates is error-prone. - Zero new dependencies — reuses the shadcn/ui <code>Calendar</code> already present in the design system.

9 0% Fill Rate Fix

Orders - Business Logic

ALL 9 ITEMS COMPLETED

P Fill rate now computed per line as `min(qty_ordered, current_stock) / qty_ordered`, then aggregated across the full basket. | Date: 8 May 2026 | Status: Shipped

- Zero-stock orders now correctly resolve to `0%` — previously the division returned `NaN` or rendered blank, masking the issue.
- Swiggy/Blinkit all-or-nothing fulfillment warning now correctly fires for any fill rate below `100%`, including the `0%` edge case.
- Prevents silent order misses on zero-stock baskets that were previously passing validation without triggering any alert.

All 9 items delivered and merged · Next steps tracked in upcoming sprint.