**📘 Project Plan Prompt: Main Project – Portable POS System**

**📌 Project Title:**

**Portable POS System**

**🎯 Objective / Purpose**

To build a **fully offline**, modular, and portable Point-of-Sale (POS) system for Windows 11 with integrated features for sales, CRM, inventory, and rewards.  
The system will feature a **futuristic neon-glass UI**, be **electron-based**, and maintain **no internet dependency** — ideal for field or mobile retail use. Splash loading page showing company logo as well.

**🧱 Framework & Platform**

| **Category** | **Spec** |
| --- | --- |
| OS | Windows 11 |
| Tech Stack | Electron (preferred) or Python |
| Database | Local SQLite (no better-sqlite3) |
| UI Frontend | HTML, CSS, JS (AJAX-style loading) |
| Backend Logic | Vanilla JS APIs in /brain folder |
| Resolution | 1920×1080 (Fixed UI) |

**📁 File Structure**

D:\Portable\_POS\_System

├── brain # Backend logic (routes, validations)

├── memory # Local DB and state files

└── public

├── css # CSS per module

├── js # JS per module

└── modules # HTML files per module

**🧩 Development Phases**

| **Phase** | **Description** |
| --- | --- |
| 1 | Boilerplate Setup (structure + sidebar) |
| 2 | Dynamic Module Loading via index.html |
| 3 | Implement Modules: Rewards → Vouchers → Promos → etc. |
| 4 | Full Feature Integration |
| 5 | Testing, Bug Fixing, Packaging |

**🎨 UI/UX Guidelines**

| **Element** | **Spec** |
| --- | --- |
| Main Window | 1920×1080 fixed |
| Sidebar | 300px, vertical, collapsible |
| Main Content | 1620px max width for modules |
| Scroll Behavior | No scroll on forms; scroll allowed on tables only |
| Theme | Glassmorphic, light-neon-blue, sleek and futuristic |

**📂 Sidebar & Navigation**

**Sidebar Structure (Only one submenu open at a time):**

| **Main Menu** | **Submenus** |
| --- | --- |
| 📊 Dashboard | — |
| 🛒 Point of Sale | ➕ New Order♻️ Update Order |
| 🎁 Promotions | 💸 Promos🎉 Rewards🎟️ Vouchers |
| 🧾 Invoicing | 🛠️ Supplies🧹 Maintenance💳 Billing |
| 📦 Orders | — |
| 💰 Expenses | — |
| 📚 Inventory | — |
| 👥 Customers | — |
| 📜 Rewards Ledger | — |
| 🏦 Suppliers/Billers | 🧑‍💼 Suppliers🧾 Billers🏛️ Banks |
| 📅 Calendar | — |
| 📈 Reports | — |
| 🛠️ Settings | — |

**🔢 Auto ID Format System**

| **Type** | **Format Example** |
| --- | --- |
| Customer | C-20250718-0001 |
| Order | O-YYYYMMDD-#### |
| Invoice | I-YYYYMMDD-#### |
| Supplier | S-YYYYMMDD-#### |
| Biller | B-YYYYMMDD-#### |

* Counter resets daily at 00:00:00 Philippine Time
* Format generated only via backend (never client-side)

**🔒 Offline Requirement & Limitations**

| **Rule** | **Enforced?** |
| --- | --- |
| No pip / npm runtime dependency | ✅ |
| No better-sqlite3 usage | ✅ |
| Fully offline, no internet | ✅ |
| Local DB (JSON or sqlite) | ✅ |
| No cloud sync or APIs | ✅ |
| No UI mockups, only working code | ✅ |

**✅ Finalized Development Plan**

**Phase 1: Boilerplate**

* **Initialize project: npm init -y, install Electron & SQLite**
* **Set up folders /brain, /memory, /public/css, /public/js, /public/modules**
* **Create index.html with 1920×1080 fixed layout**
* **Add sidebar UI (no functions yet)**
* **Add splash screen with logo**
* **Confirm the app launches via npm start**

**Phase 2: Sidebar Navigation & Dynamic Module Loader**

* **Sidebar buttons load modules using fetch()**
* **Sidebar submenu toggling logic (1 open at a time)**
* **Load module HTML + its CSS + JS cleanly after inject**

**Phase 3: Create Core Modules**

* **Start with Rewards module (empty shell)**
* **Add basic UI elements per module (no data yet)**
* **Hook front to back later**

**Phase 4: Backend Integration**

* **Use sqlite3 to connect to local .db**
* **Create auto-ID system (e.g., C-YYYYMMDD-0001)**
* **Start wiring real backend logic to each module**

**Phase 5: Final Build**

* **Package app into .exe with Electron Builder**
* **Test on another PC (offline)**
* **Optional: Create .bat shortcut to auto-run**

**🎁 Module Prompt: Rewards**

**📄 Page Title:**

**Rewards**

**🧭 Purpose**

This module manages reward-earning rules for customers. Admins can define how customers earn reward points based on transactions or specific identifiers (e.g., amount, item, item group, customer).  
Rewards convert into peso equivalents using a **global conversion rate**.

**🧱 Layout Structure**

| **Section** | **Height Ratio** | **Scroll** |
| --- | --- | --- |
| Rewards Form Tile | 1/3 | ❌ No scroll |
| Rewards Table Tile | 2/3 | ✅ Table only |
| Max Width | 1620px | Fixed |

**🎨 Theme & Behavior**

* **Theme**: Glassmorphic, light neon-blue, sleek/futuristic
* **No wrapping** on form — entire tile must fit within 1620×360px
* **All form fields conditionally rendered**, based on selected rule
* Table supports **pagination, filtering, and searching**

**🧮 Global Conversion Rate Summary**

Displayed directly below the page title:

Global Conversion Rate: [ X Point/s ] = [ ₱ Y ] [Set Button]

"X Point/s is equal to ₱Y since MM/DD/YYYY."

* "Set" button opens a modal to update conversion rate + effective date
* Conversion affects **display only**, not past records

**🔷 Form Tile – Set Rewards Rule**

**Fields & Logic**

| **Field** | **Details** |
| --- | --- |
| **Rewards ID** | Max 10 characters, alphanumeric (case-sensitive)“Generate” button for auto-ID |
| **Earn Rule** | Dropdown: Amount, Item, Item Group, Customer |
| **Conditional Fields** | Show two extra fields depending on rule type (see below) |
| **Start Date** | Required (cannot be later than End Date) |
| **End Date** | Optional |
| **Note** | Max 100 characters, multi-line (2–3 lines UI) |
| **Save** | Button to validate and save rule |

**Conditional Field Behavior**

| **Earn Rule** | **Field 1** | **Field 2** |
| --- | --- | --- |
| **Amount** | Amount (₱) | Points |
| **Item** | Item ID (6-digit format) | Points |
| **Item Group** | Group ID (2-digit format) | Points |
| **Customer** | Customer ID (C-YYYYMMDD-####) | Points |

**⚠️ Validations**

* Rewards ID must be unique
* Format checks:
  + Item ID: 6-digit padded
  + Group ID: 2-digit padded
  + Customer ID: follows C-YYYYMMDD-####
* Required fields enforced
* Start Date ≤ End Date (if End Date is provided)

**📊 Table Tile – Rewards Rule List**

**Header Controls**

| **Control** | **Function** |
| --- | --- |
| Search Bar | Search by Rewards ID, Item ID, etc. |
| Filter by Status | Dropdown: Active, Suspended, Ended |
| Show List Selector | Choose between 10 / 25 / 50 entries per page |
| 📤 Export Button | Export full Rewards data to CSV/XLSX |

**Table Columns**

| **Column** | **Description** |
| --- | --- |
| Rewards ID | Unique reward identifier |
| Rule Type | Based on selected Earn Rule |
| Earn Rule | Item/Amount/Group/Customer |
| Start Date | Start of validity |
| End Date | Optional |
| Status | One of: Active, Suspended, Ended |
| Note | Additional info |
| Action | Edit button (opens modal) |

**Edit Modal (Action Button)**

Editable Fields:

* Start Date
* End Date
* Points
* Note
* Status

❌ **Non-editable Fields:**

* Rewards ID
* Earn Rule
* Rule Type

**Pagination Controls**

* Prev / Next buttons
* Page Jumper: [ currentPage ] / [ totalPages ]

**📤 Export (CSV/XLSX)**

| **Functionality** | **Description** |
| --- | --- |
| Button Label | "Export" (above table, same row as search) |
| Exported Fields | All data in the table (no hidden columns) |
| Format | CSV and XLSX |
| Filename Format | rewards\_export\_YYYYMMDD-HHMMSS.csv |
| Export Scope | Full export (not filtered unless specified) |

**🧠 Backend API Notes**

| **Endpoint** | **Description** |
| --- | --- |
| GET /api/rewards | Retrieve all reward rules (with filters) |
| POST /api/rewards | Create new reward rule |
| PUT /api/rewards/:id | Update editable fields |
| GET /api/rewards/export | Export rewards data as CSV/XLSX |
| POST /api/conversion-rate | Update conversion rate |
| GET /api/conversion-rate | Fetch latest rate |

**🔗 Module Integrations**

| **Connected Module** | **Description** |
| --- | --- |
| Customers | Used for customer-specific reward rules |
| Orders | Reward points triggered by completed orders |
| Rewards Ledger | Pulls data for earned/used point summaries |

**🎫 Module Prompt: Vouchers**

**📄 Page Title:**

**Vouchers**

**🧭 Purpose**

This module manages single-use or multi-use **refill vouchers** that customers can redeem. Admins can enroll individual or booklet-style vouchers with refill value, active dates, and status. Tracking includes circulation, usage, and expiration.

**🧱 Layout Structure**

| **Section** | **Height Ratio** | **Scroll** |
| --- | --- | --- |
| Voucher Enrolment Tile | 1/3 | ❌ No scroll |
| Voucher Table Tile | 2/3 | ✅ Table only |
| Max Width | 1620px | Fixed |

**🎨 Theme & Behavior**

* **Theme**: Neon-blue, glassmorphic, cyber-retro look
* Layout must fit within 1620px x 1080px (excluding sidebar)
* Scrollable only for the **table**, not for form
* Same controls and pagination logic as Rewards module

**📆 Summary Section**

Displayed below the title:

As of DD/MM/YYYY, [X] in Circulation, [Y] Used, [Z] Expired.

* Auto-updated on enrollments or status changes
* Pulled from backend summary stats

**🔷 Voucher Enrolment Tile**

**Dropdown: Voucher Type**

| **Type** | **Description** |
| --- | --- |
| **Single** | Enroll one voucher with a 10-digit Voucher ID |
| **Multi/Booklet** | Enroll a range of vouchers with shared refill value |

**📥 Fields Based on Type**

| **Field** | **Type** | **Description** |
| --- | --- | --- |
| Voucher ID | Single | 10-digit, zero-padded, unique |
| Start ID | Multi | Starting ID of booklet (10-digit) |
| End ID | Multi | Ending ID (must be ≥ Start ID) |
| Refill Equivalent | Both | Number of refills the voucher provides |
| Start Date | Both | Date voucher becomes valid |
| End Date | Both | Optional end date |
| Save Button | Both | Validates and registers voucher(s) |

**⚠️ Validations**

* All voucher IDs: exactly **10 digits**, zero-padded
* For multi: **Start ID must be ≤ End ID**
* **Refill** must be positive
* **Voucher ID** uniqueness enforced
* Dates: Start Date ≤ End Date (if provided)

**📊 Voucher Table Tile**

**Header Controls (Top of Table)**

| **Control** | **Function** |
| --- | --- |
| 🔍 Search Bar | Search by Voucher ID |
| 🔽 Filter by Status | Circulation, Used, Expired |
| 📤 Export Button | Export all voucher data |
| 📥 Import Button | Upload vouchers in bulk (see format below) |
| Show List Selector | 10 / 25 / 50 entries per page |

**Table Columns**

| **Column** | **Description** |
| --- | --- |
| Voucher ID | 10-digit unique ID |
| Date Added | Auto-filled on enrolment |
| Refill | Assigned refill count |
| Date Start | When the voucher becomes valid |
| End Date | Optional expiration date |
| Status | One of: Circulation, Used, Expired |
| Action | Edit button opens modal to update details |

**Edit Modal (Action Column)**

| **Editable Fields** |
| --- |
| Start Date |
| End Date |
| Refill Count |
| Status |

❌ **Non-editable**: Voucher ID, Date Added

**📤 Export (CSV/XLSX)**

| **Feature** | **Description** |
| --- | --- |
| Button Label | "Export" (top-right of table) |
| Data Scope | Entire voucher list (unfiltered or filtered) |
| File Format | CSV and XLSX |
| Filename | vouchers\_export\_YYYYMMDD-HHMMSS.csv |

**📥 Import (CSV/XLSX)**

| **Feature** | **Description** |
| --- | --- |
| Button Label | "Import" (next to Export) |
| Import Fields Required | Voucher ID, Refill, Start Date, Status |
| Field Rules | Voucher ID: 10 digits, Refill: number, Status valid |
| Auto-fill Fields | Date Added (current), End Date = null if missing |
| File Format | CSV or XLSX |
| Duplicate Check | Skips any voucher ID that already exists |
| Result Modal | Example: "20 added, 3 skipped (duplicates)" |

**🔄 Pagination & Table Features**

* Prev / Next buttons
* Jump to: [ currentPage ] / [ totalPages ]
* Sortable columns (optional future enhancement)

**🧠 Backend API Notes**

| **Endpoint** | **Purpose** |
| --- | --- |
| GET /api/vouchers | Fetch voucher list |
| POST /api/vouchers | Add new single or multiple vouchers |
| PUT /api/vouchers/:id | Edit fields (status, dates, refill) |
| GET /api/vouchers/export | Export current list |
| POST /api/vouchers/import | Bulk add via CSV/XLSX |

**🔗 Module Integrations**

| **Connected Module** | **Description** |
| --- | --- |
| POS / Orders | May allow redeeming vouchers (future) |
| Customers | Optional future link for usage tracking |

**💸 Module Prompt: Promos**

**📄 Page Title:**

**Promos**

**🧭 Purpose**

The Promos module is designed to manage various promotional campaigns by generating **promo codes** that apply different rules—discounts, freebies, or conditional offers like “Buy X Get Y.” Promos include validity dates, optional redemption caps, and usage tracking per customer.

**🧱 Layout Structure**

| **Section** | **Height Ratio** | **Scroll** |
| --- | --- | --- |
| Promo Code Form Tile | 1/3 | ❌ No scroll |
| Promo Table Tile | 2/3 | ✅ Table only |
| Max Width | 1620px | Fixed container |

**🎨 Theme & UX Consistency**

* **Neon-blue**, sleek, glassy, futuristic (same global look)
* Form must fully fit without scrolling
* Scroll only allowed on table
* Responsive layout inside 1620×1080 (excluding sidebar)

**📆 Promo Summary Section**

Displayed directly under the page title:

As of DD/MM/YYYY, [X] Active Promo Codes, [Y] Used, [Z] Expired.

* Updated automatically based on current data
* Pulled from backend promo stats

**🔷 Promo Code Generator Tile**

**Promo Type (Dropdown)**

| **Promo Type** | **Description** |
| --- | --- |
| Fixed Discount | ₱ off |
| Percentage Discount | % off |
| Free Item | Get an item for free |
| Buy X Get Y - Free | Buy A, Get B free (quantity-based) |
| Buy X Get Y - ₱ Off | Buy A, Get B at flat ₱ discount |
| Buy X Get Y - % Off | Buy A, Get B at % discount |

**Common Fields**

| **Field** | **Description** |
| --- | --- |
| Promo Code | 15-char, alphanumeric, case-sensitive; with **Generate** button |
| Rule Fields (dynamic) | See table below (depends on selected promo type) |
| Start Date | Required |
| End Date | Optional |
| Max Redemptions | Optional max number of customer uses |
| Note | Optional, max 100 characters, 2–3 lines |
| Save Promo | Validates and submits rule |

**🎛️ Rule Field Matrix**

| **Promo Type** | **Fields Required** |
| --- | --- |
| Fixed Discount | ₱ Discount Amount |
| Percentage Discount | Discount % (1–100) |
| Free Item | Item ID (6-digit), Optional Description |
| Buy X Get Y - Free | Buy: Item ID/Group + Qty → Get: Item ID/Group + Qty |
| Buy X Get Y - ₱ Off | Same as above + ₱ Discount (flat) |
| Buy X Get Y - % Off | Same as above + Discount % |

🧩 For Buy/Get Promos:

* Toggle between **Item ID (6-digit)** or **Item Group ID (2-digit)**
* Enforce format and zero-padding

**⚠️ Validations**

* Promo Code must be unique and 15 chars max
* Required fields must be completed based on Promo Type
* Start Date ≤ End Date (if provided)
* Discount values must be within valid ranges
* Cannot create multiple promos with same code or conflicting active ranges

**📊 Promo Table Tile**

**Controls (Header Section)**

| **Control** | **Function** |
| --- | --- |
| 🔍 Search Bar | Search by Promo Code, Rule Summary |
| 🔽 Filter by Status | Active, Used, Expired, Suspended |
| 📤 Export Button | Export all promo data |
| 📥 Import Button | Import basic promo data (see format) |
| Show List Selector | 10 / 25 / 50 per page |

**Table Columns**

| **Column** | **Description** |
| --- | --- |
| Promo Code | 15-char unique code |
| Type | Promo Type (e.g. Fixed, Buy X Get Y - ₱ Off) |
| Rule Summary | Short auto-summary of the promo logic (Buy Group 01 → Get 02 @ 10%) |
| Start Date | Valid from |
| End Date | Optional |
| Status | Active, Used, Expired, Suspended |
| Redemptions | Number of unique customers who used it |
| Note | 100-char optional description |
| Action | Edit button |

**Edit Modal (Action)**

| **Editable Fields** |
| --- |
| Start Date |
| End Date |
| Max Redemptions |
| Note |
| Status |

❌ Not Editable: Promo Code, Promo Type, Usage Counter, Rule Summary

**🔄 Redemptions Counter**

* Auto-incremented on **successful promo redemption**
* Counts **unique customer usage** only (no duplicates)
* Pulls from backend logs (order ID + customer ID match)

**📤 Export (CSV/XLSX)**

| **Detail** | **Info** |
| --- | --- |
| Export Scope | All promo entries (filtered or unfiltered) |
| Format | CSV & XLSX |
| Filename | promos\_export\_YYYYMMDD-HHMMSS.csv |
| Includes | All table data + full rule definition + redemptions count |

**📥 Import (CSV/XLSX)**

**Required Fields for Import:**

| **Field** | **Notes** |
| --- | --- |
| Promo Code | 15-char, case-sensitive, must be unique |
| Type | Must match one of the allowed types |
| Rule Summary | Short summary describing logic (used for display only) |
| Start Date | YYYY-MM-DD |
| End Date | Optional |
| Status | One of: Active, Used, Expired, Suspended |
| Note | Optional |

* Fields like Redemptions, Promo Type logic, or Max Redemptions are not importable
* Import focuses only on registering basic promo entries

**Result Modal:**

19 added, 2 skipped (invalid date), 1 duplicate

**🔄 Pagination**

* Pagination component:
* [ Prev ] [ currentPage / totalPages ] [ Next ]
* Controls are consistent with other modules

**🧠 Backend API (brain/)**

| **Endpoint** | **Description** |
| --- | --- |
| GET /api/promos | Fetch promos with filters |
| POST /api/promos | Add new promo rule |
| PUT /api/promos/:code | Edit editable fields |
| GET /api/promos/export | Export promos to CSV/XLSX |
| POST /api/promos/import | Import basic promo details |
| POST /api/promos/redeem | Record promo usage |
| GET /api/promos/stats | Summary counts for header area |

**🔗 Integrations**

| **Module** | **Purpose** |
| --- | --- |
| Orders (POS) | Applies promo validation at checkout |
| Customers | Promo usage tracked per customer |
| Rewards | Promo-based points optional in future (if planned) |

**👥 Module Prompt: Customers**

**📄 Page Title:**

**Customers**

**🧭 Purpose**

Manage customer profiles with contact info, purchase history, and rewards points. Enable export and import of customer data. Provide search, filter, and sorting capabilities, and allow edits of customer details.

**🧱 Layout Structure**

| **Section** | **Height Ratio** | **Scroll** |
| --- | --- | --- |
| Customer Form Tile | 1/3 | ❌ No scroll |
| Customer Table Tile | 2/3 | ✅ Table scroll |
| Max Width | 1620px | Fixed container |

**🎨 Theme & UX Consistency**

* Neon-blue glassy futuristic style matching other modules
* Responsive, no vertical scroll on form, table scroll enabled
* Fit within 1620×1080 container excluding sidebar

**👤 Customer Form Tile**

**Fields**

| **Field** | **Description** | **Validation/Notes** |
| --- | --- | --- |
| Customer ID | Auto-generated 8-digit unique numeric | Readonly on edit |
| First Name | Text | Required |
| Last Name | Text | Required |
| Email | Text | Optional, must be valid email format |
| Phone Number | Text | Optional, validated digits |
| Date Joined | Auto-filled with current date on add | Readonly on edit |
| Total Purchases | Auto-calculated total value spent | Readonly |
| Rewards Points | Numeric, editable | Must be >= 0 |
| Status | Active / Suspended / Deleted | Default Active |
| Notes | Optional, max 200 chars |  |

**📊 Customer Table Tile**

**Controls (Header Section)**

| **Control** | **Function** |
| --- | --- |
| 🔍 Search Bar | Search by Customer ID, Name, Email |
| 🔽 Filter by Status | Active, Suspended, Deleted |
| 📤 Export Button | Export all customer data |
| 📥 Import Button | Import customers (see required fields) |
| Show List Selector | 10 / 25 / 50 per page |

**Table Columns**

| **Column** | **Description** |
| --- | --- |
| Customer ID | 8-digit unique number |
| First Name | Customer’s first name |
| Last Name | Customer’s last name |
| Email | Email address |
| Phone Number | Contact number |
| Date Joined | Date customer was added |
| Total Purchases | Sum of all purchases |
| Rewards Points | Current points balance |
| Status | Active, Suspended, Deleted |
| Notes | Optional description |
| Action | Edit button |

**✏️ Edit Modal (Action)**

| **Editable Fields** |
| --- |
| First Name |
| Last Name |
| Email |
| Phone Number |
| Rewards Points |
| Status |
| Notes |

❌ Not Editable: Customer ID, Date Joined, Total Purchases (auto)

**📤 Export (CSV/XLSX)**

| **Detail** | **Info** |
| --- | --- |
| Export Scope | All customers (filtered or unfiltered) |
| Format | CSV & XLSX |
| Filename | customers\_export\_YYYYMMDD-HHMMSS.csv |
| Includes | All table fields |

**📥 Import (CSV/XLSX)**

**Required Fields for Import:**

| **Field** | **Notes** |
| --- | --- |
| Customer ID | Optional: if included, must be unique |
| First Name | Required |
| Last Name | Required |
| Email | Must be valid email |
| Phone Number | Optional |
| Rewards Points | Numeric, default to 0 if missing |
| Status | Active / Suspended / Deleted (default Active) |
| Notes | Optional |

* If Customer ID exists, update record, else create new
* Ignore Date Joined & Total Purchases in import (auto-handled)

**Result Modal:**

45 customers imported, 3 skipped (invalid emails), 2 duplicates updated

**🔄 Pagination**

* Standard pagination:
* [ Prev ] [ currentPage / totalPages ] [ Next ]

**🧠 Backend API (brain/)**

| **Endpoint** | **Description** |
| --- | --- |
| GET /api/customers | Fetch customers with filters |
| POST /api/customers | Add new customer |
| PUT /api/customers/:id | Edit customer fields |
| GET /api/customers/export | Export customers CSV/XLSX |
| POST /api/customers/import | Import customer data |

**🔗 Integrations**

| **Module** | **Purpose** |
| --- | --- |
| Orders (POS) | Customer data linked to orders |
| Rewards | Points redemption & accumulation |
| Promos | Promo usage tracking by customer |

**🎯 Module Prompt: Rewards Ledger**

**📄 Page Title:**

**Rewards Ledger**

**🧭 Purpose**

Display detailed transaction history of rewards points per customer, showing points earned, redeemed, the equivalent monetary value at the time of each redemption (using historical conversion rates), and the related Order Number if points were used in a purchase.

**🧱 Layout Structure**

| **Section** | **Height Ratio** | **Scroll** |
| --- | --- | --- |
| Ledger Table | Full page | ✅ Full table scroll |
| Max Width | 1620px | Fixed container |

**🎨 Theme & UX Consistency**

* Neon-blue glassy futuristic style matching other modules
* Table fills container width with vertical scroll if needed
* Filter controls pinned on top for easy access

**🔍 Filter Controls (Pinned above table)**

| **Control** | **Purpose** |
| --- | --- |
| Customer Selector | Dropdown or search for Customer by ID/Name |
| Date Range Picker | Filter transactions by start/end dates |
| Transaction Type | Filter by Earned / Redeemed / All |
| Reset Filters | Clear all filters |
| Export Button | Export filtered ledger data (CSV/XLSX) |

**📊 Rewards Ledger Table Columns**

| **Column** | **Description** |
| --- | --- |
| Transaction ID | Unique transaction identifier |
| Date | Date/time of transaction |
| Customer ID | Customer’s unique ID |
| Customer Name | Customer full name |
| Type | Earned / Redeemed |
| Points | Number of points earned or redeemed |
| Equivalent Value | Monetary equivalent value at transaction time (if redeemed) |
| Conversion Rate | Conversion rate applied at transaction time |
| Order Number | The order ID where points were redeemed (if applicable) |
| Notes | Optional transaction notes |

**📤 Export (CSV/XLSX)**

| **Detail** | **Info** |
| --- | --- |
| Export Scope | All ledger entries matching current filters |
| Format | CSV & XLSX |
| Filename | rewards\_ledger\_export\_YYYYMMDD-HHMMSS.csv |
| Includes | All table columns |

**🧠 Backend API (brain/)**

| **Endpoint** | **Description** |
| --- | --- |
| GET /api/rewards/ledger | Fetch ledger entries with filters |
| GET /api/rewards/ledger/export | Export ledger CSV/XLSX with applied filters |

**🧠 Special Notes**

* Points shown in "Equivalent Value" must use the **conversion rate that was valid at the time of each transaction**, not the current global rate.
* Historical conversion rates must be stored or retrievable for accurate redemption value display.
* When points are redeemed, link to the corresponding **Order Number** so redemption traceability is clear.

📦 **Module Prompt: Inventory**  
📄 **Page Title:** Inventory Management

**🖥️ Layout**

* **Left Tile: Item Groups**
  + Width: ~300px (vertical list)
  + Displays all item groups + a special "All Items" option
  + Clicking a group filters the Item List table to show only items under that group
* **Right Tile: Item List Table**
  + Takes remaining width (~1320px for 1920 screen - 300 sidebar - 300 group tile)
  + Shows items filtered by selected group or all items if no group selected

**🗂 Item Groups Tile**

* Shows a vertical list of item groups with format:  
  **Group Name (Group ID)**
* Clicking on a group filters the item list accordingly
* Special "All Items" option shows all items
* Add/Edit/Delete functionality for Item Groups:
  + Group ID: 2-digit numeric, unique, required
  + Group Name: text, required

**📋 Item List Table**

**Top Controls (Above Table):**

* Search bar (search by Item Name, Item ID, Barcode)
* Filter by Group (dropdown matching Item Groups)
* Export button (CSV/XLSX) — exports all visible data
* Import button (CSV/XLSX) — imports basic data (see below)

**Columns:**

| **Item ID (SKU)** | **Item Name** | **Description** | **Unit Price (₱)** | **Current Stock** | **Reorder Level** | **Item Group (Name (ID))** | **Barcode** | **Action** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |

**Pagination:**

* Next / Prev buttons
* Jump to page input

**➕ Add / Edit Item Modal Form**

Fields and order:

1. **Item Group (Group ID)** – Dropdown/select, required
2. **Item ID (SKU)** – Auto-generated, readonly, 6-digit numeric starting with Group ID (e.g., group 02 → SKU 020001)
3. **Item Name** – Text, required
4. **Description** – Multiline text, optional
5. **Unit Price** – Numeric (currency), required
6. **Current Stock** – Numeric, required
7. **Reorder Level** – Numeric, required
8. **Barcode** – Text, optional
9. **Save button** – validates and saves

**🗃 Item Group Management Modal**

Fields:

* Group ID (2-digit numeric, unique)
* Group Name (text)
* Save button

**📥 Import Details**

**Import Fields (CSV/XLSX):**

* Item Group ID (2-digit) — required
* Item Name — required
* Description — optional
* Unit Price — required
* Current Stock — required
* Reorder Level — required
* Barcode — optional

**Notes:**

* Item ID is auto-generated based on Group ID + sequential counter
* If imported item’s group ID does not exist → reject or create group? (Specify behavior)
* Duplicate check based on Item Group + Item Name + Barcode (if available)
* Import result summary: e.g., "15 items added, 2 skipped (duplicates), 1 failed (invalid price)"

**📤 Export Details**

* Export all visible items with all columns included
* Filename format: inventory\_export\_YYYYMMDD-HHMMSS.csv

**🧠 Backend API Endpoints**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/inventory/groups | Get list of item groups |
| POST | /api/inventory/groups | Add new item group |
| PUT | /api/inventory/groups/:id | Update item group |
| DELETE | /api/inventory/groups/:id | Delete item group |
| GET | /api/inventory/items | Get list of items (filterable) |
| POST | /api/inventory/items | Add new item |
| PUT | /api/inventory/items/:id | Update item |
| DELETE | /api/inventory/items/:id | Delete item |
| POST | /api/inventory/items/import | Bulk import items |
| GET | /api/inventory/items/export | Export items |

**🛠 Validations & Notes**

* Group ID: exactly 2 digits, unique
* Item ID: auto-generated, starts with group ID + 4-digit counter (e.g., 020001)
* Item Name: required, max length 100
* Unit Price, Current Stock, Reorder Level: required, numeric, non-negative
* Barcode: optional, alphanumeric
* On deleting a group: confirm deletion, warn if items exist under group (optionally cascade delete or reject)
* All UI elements styled consistently with global neon-glass-blue theme

**💡 Suggestions for Future Enhancements**

* Batch Edit for items
* Stock adjustment logs/history
* Barcode scanning integration
* Low stock alerts and notifications
* Supplier integration (planned for separate module)

🏭 **Module Prompt: Suppliers**  
📄 **Page Title:** Supplier Management

**🖥️ Layout & Functionality**

* Full-page table listing all suppliers
* Search bar to filter suppliers by **Name**, **Supplier ID**, **Contact Person**, **Email**
* Buttons:
  + **Add Supplier** (opens modal form)
  + **Edit** (per row)
  + **Delete** (per row, with confirmation)
  + **Export** (CSV/XLSX) — exports all supplier data
  + **Import** (CSV/XLSX) — imports supplier data (fields detailed below)

**📋 Supplier List Table Columns**

| **Supplier ID** | **Supplier Name** | **Contact Person** | **Email** | **Phone Number** | **Address** | **Notes** | **Action** |
| --- | --- | --- | --- | --- | --- | --- | --- |

* **Supplier ID**: Unique ID (auto-generated or manual)
* **Notes**: Optional additional info

**➕ Add / Edit Supplier Modal Form**

Fields (required fields marked):

1. **Supplier Name** (required)
2. **Contact Person** (optional)
3. **Email** (optional, validate format)
4. **Phone Number** (required)
5. **Address** (optional, multiline text)
6. **Notes** (optional, multiline text)
7. **Save** button

**📥 Import Details**

**Import Fields (CSV/XLSX):**

* Supplier Name (required)
* Contact Person (optional)
* Email (optional)
* Phone Number (required)
* Address (optional)
* Notes (optional)

**Import Behavior:**

* Duplicate detection based on Supplier Name + Email (if provided)
* Import summary with counts of added, skipped (duplicates), and errors

**📤 Export Details**

* Export all supplier data with all fields
* Filename format: suppliers\_export\_YYYYMMDD-HHMMSS.csv

**🧠 Backend API Endpoints**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/suppliers | Get list of suppliers |
| POST | /api/suppliers | Add new supplier |
| PUT | /api/suppliers/:id | Update supplier |
| DELETE | /api/suppliers/:id | Delete supplier |
| POST | /api/suppliers/import | Bulk import suppliers |
| GET | /api/suppliers/export | Export suppliers data |

**🛠 Validations & Notes**

* Supplier Name and Phone Number are required
* Email format validation if provided
* Confirm on delete to prevent accidental removal

🏢 **Module Prompt: Billers**  
📄 **Page Title:** Maintenance Companies (Billers) Management

**🖥️ Layout & Functionality**

* Full-page table listing all billers (maintenance companies)
* Search bar to filter billers by **Company Name**, **Biller ID**, **Contact Person**, **Email**
* Buttons:
  + **Add Biller** (opens modal form)
  + **Edit** (per row)
  + **Delete** (per row, with confirmation)
  + **Export** (CSV/XLSX) — exports all biller data
  + **Import** (CSV/XLSX) — imports biller data (fields detailed below)

**📋 Biller List Table Columns**

| **Biller ID** | **Company Name** | **Contact Person** | **Email** | **Phone Number** | **Address** | **Notes** | **Action** |
| --- | --- | --- | --- | --- | --- | --- | --- |

* **Biller ID**: Unique ID (auto-generated or manual)
* **Notes**: Optional additional info

**➕ Add / Edit Biller Modal Form**

Fields (required fields marked):

1. **Company Name** (required)
2. **Contact Person** (optional)
3. **Email** (optional, validate format)
4. **Phone Number** (required)
5. **Address** (optional, multiline text)
6. **Notes** (optional, multiline text)
7. **Save** button

**📥 Import Details**

**Import Fields (CSV/XLSX):**

* Company Name (required)
* Contact Person (optional)
* Email (optional)
* Phone Number (required)
* Address (optional)
* Notes (optional)

**Import Behavior:**

* Duplicate detection based on Company Name + Email (if provided)
* Import summary with counts of added, skipped (duplicates), and errors

**📤 Export Details**

* Export all biller data with all fields
* Filename format: billers\_export\_YYYYMMDD-HHMMSS.csv

**🧠 Backend API Endpoints**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/billers | Get list of billers |
| POST | /api/billers | Add new biller |
| PUT | /api/billers/:id | Update biller |
| DELETE | /api/billers/:id | Delete biller |
| POST | /api/billers/import | Bulk import billers |
| GET | /api/billers/export | Export billers data |

**🛠 Validations & Notes**

* Company Name and Phone Number are required
* Email format validation if provided
* Confirm on delete to prevent accidental removal

🏦 **Module Prompt: Banks**  
📄 **Page Title:** Philippine Banks List

**🖥️ Layout & Functionality**

* Simple full-page list/table of Philippine banks
* Search bar to filter by bank name or abbreviation
* Buttons:
  + **Add Bank** (modal form)
  + **Edit Bank** (modal form)
  + **Delete Bank** (with confirmation)
  + **Export** (CSV/XLSX) — export entire list
  + **Import** (CSV/XLSX) — import banks

**📋 Banks Table Columns**

| **Bank Name** | **Short Name (Abbreviation)** | **Action** |
| --- | --- | --- |

* **Bank Name:** Full official bank name (required)
* **Short Name:** Standard Philippine bank abbreviation (required)

**➕ Add / Edit Bank Modal Form**

Fields (all required):

1. **Bank Name** (full official name)
2. **Short Name** (common abbreviation/shortened term)
3. **Save** button

**📥 Import Details**

**Import Fields (CSV/XLSX):**

* Bank Name (required)
* Short Name (required)

**Import Behavior:**

* Duplicate detection based on Bank Name or Short Name
* Import summary with counts of added, skipped (duplicates), and errors

**📤 Export Details**

* Export full bank list with both fields
* Filename format: banks\_export\_YYYYMMDD-HHMMSS.csv

**🧠 Backend API Endpoints**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/banks | Get list of banks |
| POST | /api/banks | Add new bank |
| PUT | /api/banks/:id | Update bank |
| DELETE | /api/banks/:id | Delete bank |
| POST | /api/banks/import | Bulk import banks |
| GET | /api/banks/export | Export banks list |

**🛠 Validations & Notes**

* Both fields are mandatory
* Confirm on delete to avoid accidental removal
* Banks limited to Philippine banks only (validation or manual review recommended)

🛠️ **Module Prompt: 📅 Calendar**  
📅 **Page Title:** Maintenance & Holiday Calendar

**🖥️ Layout & Functionality**

* Full-page calendar view (monthly default, with week/day toggle)
* Display color-coded entries:
  + **Maintenance Activities / Alarms** (custom events)
  + **Holidays** (fetched & synced online)
* Toolbar with:
  + Add Activity/Alarm button
  + Sync Holidays button (manual trigger)
  + View toggle (Month, Week, Day)
  + Filter: Show/Hide Activities, Holidays

**➕ Add / Edit Maintenance Activity or Alarm Modal**

Fields:

1. **Title** (required) — Name of activity/alarm
2. **Description** (optional)
3. **Date & Time** (required) — support single date/time or recurring (daily, weekly, monthly)
4. **Type** (required) — Dropdown: Alarm / Activity
5. **Notify** (optional) — Checkbox to enable notification/reminder
6. **Save** button

**🎉 Holidays Sync**

* Sync holidays automatically from a reliable online Philippine holidays API (or service)
* Holidays displayed on the calendar with distinct color/style
* Manual “Sync Holidays” button to refresh
* Holidays are read-only (cannot be edited/deleted in calendar)

**📋 Calendar Event Behavior**

* Clicking an event opens a detail popup with Edit/Delete options (except holidays which are read-only)
* Drag & drop to reschedule activities/alarms
* Recurring events supported with edit options for single or series

**🧠 Backend API Endpoints**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/maintenance/events | Fetch all calendar events |
| POST | /api/maintenance/events | Create new activity/alarm |
| PUT | /api/maintenance/events/:id | Update existing event |
| DELETE | /api/maintenance/events/:id | Delete activity/alarm |
| GET | /api/maintenance/holidays/sync | Trigger holiday sync |

**🛠 Notes & Validation**

* Activities & alarms can be scheduled with or without notifications
* Recurrence options should support editing one occurrence or entire series
* Holiday sync should happen in background without disrupting user actions
* Ensure timezone consistency for events
* UI should visually distinguish between alarms, activities, and holidays

**🛒 Module Prompt: POS – New Order**

**📄 Page Title:**

**Point of Sale – New Order**

**🧭 Layout Structure**

| **Section** | **Description** |
| --- | --- |
| Left Panel (300px) | Category (Item Groups) |
| Center Panel | Items Grid (per group) |
| Right Panel | Cart + Order Summary + Payment |

**💡 Design Notes**

* **Touchscreen-friendly** layout (large tap targets, responsive buttons)
* **Receipt preview** styled for **80mm thermal printer**
* **Order OR# field** appears if transaction > ₱20
* **Scroll behavior:**
  + 🔁 Cart Item List only (scroll if >8 items)
  + ❌ No scroll on full form or totals
* Fits perfectly within **1920x1080**, minus sidebar (1620px working space)

**🔹 Item Group Panel (Left – 300px)**

* Scrollable list of **Item Groups**
* First item is: **All Items**
* Each group is a **tile/button**
* Clicking a group filters items on center panel

**🔹 Items Grid Panel (Center)**

* Grid layout: 4–5 columns (responsive)
* Each item shown as **large button/tile**:
  + 📦 **Item Name**
  + 🔢 **Item ID** (SKU)
  + 💰 **Unit Price**
* Clicking an item adds it to cart (or increments quantity)

**🛒 Cart + Order Summary Panel (Right)**

**🧾 Cart Panel**

| **Field** | **Behavior** |
| --- | --- |
| Item Rows | Shows up to 8 items; scrollable after that |
| Remove Icon | Remove individual items |
| Qty Control | Increment/decrement or type in |
| Subtotal | Updates per item |

**💰 Totals + Actions**

| **Field** | **Notes** |
| --- | --- |
| Subtotal | Auto-computed |
| Discount (Optional) | Amount or % |
| VAT (Optional toggle) | Add or exclude |
| TOTAL | Final bill |

**🔐 Payment Section**

**Payment Method (Dropdown):**

* 💵 Cash
* 📱 GCash → shows **Reference Number** field
* 🏦 Bank Transfer → shows:
  + **Bank Name** (searchable from Bank Repository)
  + **Reference Number**

**🧾 OR# Field (Official Receipt Number)**

* Only required when total is **above ₱20**
* Input field: Optional by default
* Validated format: Alphanumeric, max 20 characters

**✅ Order Buttons**

* [🧾 Print Receipt Preview]
* [💾 Confirm Order]
* [⏸️ Hold Order]

**🧾 Receipt View (80mm Thermal Format)**

Simulated print preview opens in modal:

[LOGO]  
[STORE NAME]

123 Main St, City, Philippines

Tel: (XXX) XXX-XXXX

Date: 07/18/2025 Time: 03:42 PM

Order #: O-20250718-0001  
Status: Paid

Cashier: Juan Dela Cruz

Sold To:

C-20250718-0002

Maria Santos

09171234567

Blk 3 Lot 12, Barangay Mabuhay,

Quezon City

-------------------------------

Qty Item Subtotal

-------------------------------

2 Siomai ₱60.00

1 Iced Tea ₱30.00

-------------------------------

Subtotal: ₱90.00

VAT (12%): ₱10.80

TOTAL: ₱100.80

Payment Method: GCash

Reference No: G1234567890

OR#: 100102943

Thank you for your purchase!

-------------------------------  
**✅ Receipt Notes**

* **"Sold To"** section is only shown if a customer is selected.
* If no customer selected: this section is omitted.
* Address wraps properly within 80mm printable width (max 2–3 lines).
* Still compatible with **thermal printer formats** (ESC/POS style if needed later).

**📥 Auto-Generated Order ID Format**

**Format:** O-YYYYMMDD-####  
**Resets daily**, stored in backend

**✅ Validations**

| **Field** | **Validation** |
| --- | --- |
| Order Items | Required (at least 1) |
| Payment Method | Required |
| GCash / Bank | Reference Required |
| Bank Transfer | Valid bank required |
| OR# | Required if total > ₱20 |
| Stock | Deducted only on Confirm Order |

**🧠 Backend Considerations**

| **Endpoint** | **Function** |
| --- | --- |
| POST /api/orders | Create order |
| GET /api/inventory/:id | Fetch item info |
| POST /api/orders/receipt | Return printable layout |
| POST /api/orders/deduct | Deduct stock |
| POST /api/or/check | (Optional) OR# conflict check |

**✅ ✅ ✅ Checklist of Built Modules (Menu)**

Here's the current checklist of all modules we've worked on:

**✅ Built Modules**

* **🎁 Rewards**
* **🎟️ Vouchers**
* **💸 Promos**
* **👥 Customers**
* **📜 Rewards Ledger**
* **📚 Inventory**
* **🏦 Suppliers**
* **🧾 Billers**
* **🏛️ Banks**
* **🧹 Maintenance**
* **🛒 POS – New Order** ✅ (current)

**📦 Module Prompt: Orders**

**📄 Page Title:**

**Orders**

**🎯 Purpose**

To display, manage, and track all sales transactions created via the **POS – New Order** module. Enables staff/admins to filter, search, view, and update order-related data such as payment info, order status, and items purchased.

**🖥️ Layout Structure**

| **Section** | **Behavior** |
| --- | --- |
| Top Panel (Filters) | Fixed, no scroll |
| Orders Table | Scrollable |
| Receipt Preview (Modal) | Opens when viewing order |
| Page Width | Max 1620px (300px sidebar excluded) |
| Screen Height | Fullscreen 1080p compatible |

**🔎 Filter Controls (Top Panel)**

* 🔍 **Search Bar** – Order #, Customer ID, Name, Mobile
* 🔽 **Status Filter** – Hold, Paid, Cancelled
* 📅 **Date Range Picker** – Order Date range
* 💳 **Payment Type Filter** – Cash, GCash, Bank Transfer
* 📥 **Export Button** – Exports selected or full order list as CSV/XLSX
* ✅ **Show List** – 10 / 25 / 50 entries per page

**📋 Orders Table Columns**

| **Column** | **Description** |
| --- | --- |
| Order # | Auto-generated (O-YYYYMMDD-####) |
| Order Date | Timestamp of when order was placed |
| Customer | Format: Name (Customer ID)If none, display: Walk-in |
| Total Amount | Final calculated ₱ amount (incl. tax) |
| Payment Method | Cash / GCash / Bank Transfer |
| Reference No. | GCash or Bank Ref # (if applicable) |
| Bank Name | Only if Bank Transfer (full bank name) |
| Status | Hold, Paid, Cancelled |
| OR # | Optional field (Official Receipt #) |
| Action | 👁 View (Receipt), ✏️ Update |

**🛠️ Update Order (Action: Edit)**

Clicking ✏️ **Update** opens modal with:

* Update **Status** → Hold / Paid / Cancelled
* Update **OR #** (optional)
* View **Customer Info** (readonly)
* View **Items Summary** (readonly)
* Update **Payment Type** (if needed)
* Update **Reference Number / Bank Info**

Validation:

* If changing to **Paid**, payment method + reference required.
* Cancelled orders should show optional cancel note field.

**🧾 Receipt View Modal**

Same layout as New Order's receipt:

* 80mm thermal format
* Includes:
  + Store Info
  + Date, Order #
  + **Sold To:** (if customer is selected)
  + Itemized list
  + VAT, Total, Payment info
  + OR # (if exists)
* **Reprint Button** available (simulate or export receipt)

**🔗 Integrations**

| **Module** | **Purpose** |
| --- | --- |
| **POS (New Order)** | Orders originate from here |
| **Customers** | Links customer to order |
| **Inventory** | Adjusts stock level when order is confirmed |
| **Rewards** | Points earned after successful payment |
| **Rewards Ledger** | Transaction logs points earned/used here |

**📥 Export Behavior**

* Export includes:
  + All table fields
  + Item Summary as a joined string (e.g., 2x Siomai, 1x Iced Tea)
* Format: orders\_export\_YYYYMMDD-HHMMSS.csv

**✅ Sample Data Example:**

| **Order #** | **Customer** | **Amount** | **Status** | **Payment** | **OR #** |
| --- | --- | --- | --- | --- | --- |
| O-20250718-0001 | Maria Santos (C-20250718-0002) | ₱100.80 | Paid | GCash | 10293847 |

**📌 Backend Endpoints**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/orders | Get list of orders |
| POST | /api/orders | Add new order (POS) |
| PUT | /api/orders/:id | Update status/payment/ref |
| GET | /api/orders/:id/receipt | Get receipt data for display |
| GET | /api/orders/export | Export order list |

**🧪 Testing Scenarios**

* Add new order → should appear in real-time in Orders page.
* Try filtering by "Hold" → only open transactions shown.
* Change GCash order → validate ref# required.
* Click View → Receipt layout appears with full breakdown.
* Reprint receipt → no logic change, just display.

**🔁 Canceled Orders — Inventory Restock Logic**

When an order’s **Status** is changed to **Canceled**, the system must:

1. **Automatically return item quantities to inventory stock.**
   * Each item’s Current Stock will increase based on the quantity from the canceled order.
   * This applies **only if** the order was previously marked as **Hold** or **Paid**.
   * Prevents double-adding if order was already canceled before.
2. **Rollback Notes in Order Logs**
   * Record a note/timestamp in order’s internal log:

"Order canceled on 07/18/2025. Inventory restocked."

* + Displayed optionally in audit logs or backend reports.

1. **Integration Reminder**
   * Updates will affect the **Inventory module**, modifying Current Stock.
   * Reward points earned for this order (if already credited) **should be rolled back** and reflected in the **Rewards Ledger** as:

"Points reversed due to order cancellation (Order #: O-YYYYMMDD-####)"

**🛠 Module Prompt: POS – Update Order**

**📄 Page Title: Update Order**

This module allows editing of previously placed *Hold* orders. It supports order review, modification, customer assignment, payment updates, and re-issuance.

**⚙️ Purpose**

Enable cashiers or authorized users to:

* Resume, edit, or update a previously placed *Hold* order.
* Modify items, payment method, customer info.
* Convert to finalized sale with receipt preview and issuance.

**📐 Layout**

| **Section** | **Width** | **Height** | **Scroll** |
| --- | --- | --- | --- |
| Left: Search + Order Details Panel | 400px | Full height | ✅ Scrollable |
| Right: Editable Order Panel (Cart + Items) | 1220px | Full height | ✅ Cart list scroll only if >8 items |

**🔍 Left Panel: Search + Load Order**

**🔎 Order Search**

* Search bar to enter:
  + Order Number (O-YYYYMMDD-####)
  + Customer ID
  + Customer Name
* List of *Hold* Orders below with filters:
  + Date range
  + Customer Name
  + Sort by latest or oldest
* Click on a result loads order into right panel

**📋 Right Panel: Edit & Finalize Order**

**🧾 Top Info**

* **Order ID:** O-YYYYMMDD-####
* **Date Created** (Display-only)
* **Cashier/User:** Auto-filled
* **Customer (Optional):**
  + Search/Add via modal
  + Display: ID, Name, Mobile, Address

**🛒 Order Cart**

* Show existing cart items in editable format:
  + Item Name
  + Quantity (+/-)
  + Unit Price
  + Line Total
  + ❌ Remove Item
* **Scroll for cart if more than 8 items**

**🧩 Add More Items**

* Same tile-based **Item Group** and **Items** layout as *New Order*
* Responsive and touch-friendly UI
* Click to add/update quantities in cart

**💳 Payment Section**

* **Total Amount (₱)**
* **Payment Option:**
  + Cash
  + GCash → requires Reference Number
  + Bank Transfer → requires:
    - Bank (search from Bank List)
    - Reference Number
* **Amount Tendered** and **Change Due** auto-calc
* **Add Optional OR# Field**
  + Required only for >₱20 transaction
  + Input is optional otherwise

**📄 Receipt Preview**

Click "Preview Receipt" shows a **mock 80mm thermal receipt layout**:

* Store Header
* OR# (if provided)
* Order #
* Date & Time
* Cashier/User
* Itemized List
* Subtotal, Total, Payment Info
* Customer Info (if provided): ID, Name, Mobile, Address

**✅ Buttons**

* ❌ Cancel Order → Confirmation required; returns item quantity to Inventory
* 🖊️ Save Updates → Applies changes
* 📤 Finalize & Issue Receipt → Marks order as completed, shows receipt, deducts stock

**🔁 Business Logic**

| **Logic** | **Behavior** |
| --- | --- |
| Updating Order | Allowed only on Hold orders |
| Adding Items | Must respect inventory stock |
| Removing Items | Returns item back to temporary available pool |
| Cancel Order | Puts item quantity back to inventory |
| Finalization | Marks order as Completed and logs Rewards, Points, etc. |

**🔒 Permissions**

* Only authorized roles (e.g., Cashier, Manager) can update orders
* Cancel & Finalize require role-based checks

**📂 Backend Routes**

| **Method** | **Endpoint** | **Purpose** |
| --- | --- | --- |
| GET | /api/orders/hold | Fetch all *Hold* orders |
| GET | /api/orders/:id | Get order details |
| PUT | /api/orders/:id/update | Save edits to Hold order |
| POST | /api/orders/:id/finalize | Convert to finalized |
| DELETE | /api/orders/:id | Cancel order and return stocks |

**✅ Order Update Module Checklist**

| **Feature** | **Status** |
| --- | --- |
| Hold order lookup | ✅ |
| Load + modify cart | ✅ |
| Add/remove items | ✅ |
| Update payment details | ✅ |
| Receipt preview (80mm) | ✅ |
| OR# field + logic | ✅ |
| Customer details (sold to) | ✅ |
| Cancel logic (return stock) | ✅ |
| Finalize logic (deduct stock, update rewards) | ✅ |
| Cart scroll on >8 items | ✅ |
| Touch-optimized item layout | ✅ |
| Role-based access | ✅ |

**🧾 Module Prompt: Invoicing > Supplies**

**📄 Page Title:**

**Invoicing – Supplies**

**🎯 Purpose:**

Handles invoice generation for items pulled from Inventory. Acts like a supplier-facing POS module to log, track, and print official supply-related invoices.

Invoices generated here will feed directly into the **Expenses** module (as purchases).

**🖼️ Layout Structure**

* 📌 **Toggle Option**: New Invoice | Update Invoice
  + Toggling switches between modes.
* 📐 **Main Workspace** (POS-like layout):
  + Left: Item Group Selector (tile-style, 300px width)
  + Center: Item Tiles (touch-friendly)
  + Right: Invoice Cart Panel (scroll if items exceed 8 rows)

**📝 New Invoice Section**

**🆔 Auto-Generated Invoice ID**

* Format: I-YYYYMMDD-####
* Starts at 0001 daily

**📥 Input Fields:**

| **Field** | **Required** | **Details** |
| --- | --- | --- |
| Supplier Name | ✅ | Dropdown (from Suppliers list) |
| Invoice Date | ✅ | Auto-filled to current date |
| Payment Type | ✅ | Cash, GCash, Bank Transfer |
| Reference Number | ❌ | Required if GCash or Bank Transfer |
| Bank Name | ❌ | Searchable (if Bank Transfer only, from Bank list) |
| OR Number | ❌ | Optional (user-supplied manual input) |

**📦 Item Selector (Center Panel)**

* Touch-friendly tiles
* Clicking item adds it to invoice cart
* Grouped by Item Group selected on the left
* Quantities adjustable via stepper or input box

**🛒 Invoice Cart Panel (Right)**

| **Column** | **Details** |
| --- | --- |
| SKU | Auto (from Inventory) |
| Item Name | From Inventory |
| Unit Price | From Inventory |
| Qty | Editable |
| Subtotal | Computed |
| ❌ Remove Button | Remove item |

* Scroll allowed if items exceed 8 rows

**💾 Save Invoice Button**

* Validates required fields
* Pushes invoice data to:
  + Local database
  + **Expenses** module
* Opens print preview (optional toggle)

**🔁 Update Invoice Mode**

* Invoice Selector: Search by Invoice ID, Supplier, or Date
* All previous fields become editable
* Cannot change Supplier or Invoice ID
* Cart reloads previous invoice items
* Save Button updates database and reopens invoice preview

**🖨️ Invoice Printing (A5 Format)**

All invoices across **Supplies**, **Maintenance**, and **Billing** use the same format:

* 📄 **Paper Size**: Half A4 (A5, portrait recommended)
* **Header**:
  + Business Name
  + Address, TIN (optional, pulled from Settings)
  + *INVOICE* title
  + Invoice ID, Date
* **Supplier Details**:
  + Supplier Name
  + Contact Info (if available)
* **Item Table**:

| **Item** | **Qty** | **Unit Price** | **Subtotal** |
| --- | --- | --- | --- |

* **Total Amount** at bottom
* Optional Signature / Remarks field
* 🖨️ **Print Trigger**: Auto-opens after Save (optional)
* 📋 **Export-ready Format**: Clean CSS, no background colors

**📤 Export Function**

* Exports all invoice records (CSV/XLSX)
* Includes all fields:
  + Invoice ID, Date, Supplier, Items, Amounts, Payment Type, OR#, etc.

**📥 Import Function**

* Optional future enhancement
* May require structured invoice import (not recommended until billing APIs are introduced)

**📚 Integration Points**

* ✅ **Suppliers Module**: Dropdown list + linkage
* ✅ **Inventory Module**: Item source (live)
* ✅ **Expenses Module**: Each saved invoice creates an Expense record
* ✅ **Banks Module**: Used for Bank Transfer validation
* ✅ **Settings**: For business info on printed invoice

**🧾 Module Prompt: Invoicing > Maintenance**

**📄 Page Title:**

**Invoicing – Maintenance**

**🎯 Purpose:**

Used to log and manage maintenance-related invoices where the items are **manual service entries**, not pulled from Inventory. Examples: A/C cleaning, equipment repair, plumbing, etc.

Each invoice records expense for services, and feeds directly into the **Expenses** module.

**🖼️ Layout Structure**

* 📌 **Toggle Option**: New Invoice | Update Invoice
* 🧭 Layout mimics POS-style invoice:
  + Left: Billers dropdown (not tiles)
  + Center: Manual service entry form (particulars)
  + Right: Invoice Summary + Cart

**📝 New Invoice Section**

**🆔 Auto-Generated Invoice ID**

* Format: I-YYYYMMDD-####
* Daily reset, system-generated

**📥 Input Fields:**

| **Field** | **Required** | **Details** |
| --- | --- | --- |
| Biller Name | ✅ | Dropdown (from Billers list) |
| Invoice Date | ✅ | Auto-filled to today |
| Payment Type | ✅ | Cash, GCash, Bank Transfer |
| Reference Number | ❌ | Required if GCash or Bank Transfer |
| Bank Name | ❌ | Searchable if Bank Transfer |
| OR Number | ❌ | Optional manual input |

**🧾 Service Entry Panel (Center)**

**📄 Add Particular (manual input form)**

| **Field** | **Required** | **Notes** |
| --- | --- | --- |
| Particular Description | ✅ | Multiline, max 200 characters |
| Service Fee | ✅ | ₱ input, numbers only |

* ➕ **Add to Invoice** button adds to cart
* No inventory link required

**🛒 Invoice Cart Panel (Right)**

| **Column** | **Details** |
| --- | --- |
| Description | User-typed |
| Fee | Entered value |
| ❌ Remove Button | Remove item |

* Scroll enabled if items exceed 8 rows

**💾 Save Invoice Button**

* Validates:
  + Biller
  + At least 1 service item
  + Payment method logic
* Saves to:
  + Local database
  + **Expenses** module
* Opens **print preview** after save (A5 invoice format)

**🔁 Update Invoice Mode**

* Select existing invoice (search by Invoice ID, Biller, or Date)
* Edit:
  + Particulars list
  + Fees
  + Payment info
* Cannot change Invoice ID or Biller
* Save button updates the invoice + re-opens print view

**🖨️ Invoice Printing (Universal A5 Format)**

Used across **Supplies**, **Maintenance**, and **Billing**

* **Paper**: A5 (half A4), portrait
* **Header**:
  + Business Name + TIN (from Settings)
  + *INVOICE*
  + Invoice ID, Invoice Date
* **Biller Info**:
  + Name
  + Optional Contact Info
* **Table**:  
  | Particular | Amount |
* **Footer**:
  + Total Amount
  + Optional Notes/Signature

**📤 Export Function**

* Exports all maintenance invoices (CSV/XLSX)
* Includes:
  + Invoice ID, Date, Biller, Particulars, Amounts, OR #, Payment Type, Bank info (if any)

**📚 Integration Points**

* ✅ **Billers Module** → Dropdown
* ✅ **Expenses Module** → Invoice linked as new expense
* ✅ **Banks Module** → Used for bank transfers
* ✅ **Settings Module** → For printed invoice details

**🧾 Module Prompt: Invoicing > Billing**

**📄 Page Title:**

**Invoicing – Billing**

**🎯 Purpose:**

Used to record and manage regular **utility or service bills** such as electricity, internet, water, rent, etc. This module logs invoices for recurring billing and syncs them directly to the **Expenses** module.

* Billers are pulled from the **Billers module**.
* Each invoice is treated as an expense with optional OR# and payment reference.

**🖼️ Layout Structure:**

* 🔘 **Toggle**: New Billing Invoice | Update Billing Invoice
* 📋 Simple entry form with right-side summary
* 🧾 Billing format is minimalist (single line per invoice)

**📥 Input Fields:**

| **Field** | **Required** | **Details** |
| --- | --- | --- |
| Invoice ID | Auto | Format: I-YYYYMMDD-#### |
| Biller Name | ✅ | Dropdown from **Billers** module |
| Billing Period | ✅ | MM/YYYY or select Month-Year |
| Invoice Date | ✅ | Auto-filled to today |
| Due Date | ❌ | Optional |
| Particular / Description | ✅ | Max 200 characters |
| Amount Due | ✅ | ₱ input |
| Payment Type | ✅ | Cash, GCash, Bank Transfer |
| Reference Number | ❌ | Required if GCash or Bank Transfer |
| Bank Name | ❌ | Dropdown/search from **Banks**, required if Bank Transfer |
| OR Number | ❌ | Optional manual input |

**💾 Save Invoice Button**

* ✅ Validates:
  + Biller
  + Amount
  + Billing Period
  + Payment info
* ✅ Saves invoice to:
  + Local database
  + **Expenses module**
* ✅ Opens **A5 invoice preview** upon saving

**🔁 Update Billing Invoice**

* Allows selecting existing billing invoice (Invoice ID, Biller, Period)
* Editable fields:
  + Description
  + Amount
  + Due Date
  + Payment Info
  + OR Number
* Cannot change Invoice ID or Biller
* Auto updates linked expense

**🖨️ Invoice Printing (A5 Format)**

Same universal invoice print layout as Supplies and Maintenance:

* A5 portrait
* Header: Business name, "INVOICE", Invoice ID, Date
* Biller name and optional contact
* Table with:  
  | Description | Amount |
* Footer: Total, Payment info, OR#, signature (optional)

**📤 Export (CSV/XLSX)**

* ✅ Export all or filtered billing invoices
* ✅ Columns:
  + Invoice ID
  + Invoice Date
  + Biller Name
  + Billing Period
  + Due Date
  + Description
  + Amount
  + Payment Method
  + Reference #
  + Bank Name (if any)
  + OR Number

**📚 Integration Points**

| **Module** | **Connection** |
| --- | --- |
| **Billers** | Biller dropdown list |
| **Expenses** | Invoice reflected as expense |
| **Banks** | Bank list used for Bank Transfer |
| **Settings** | Used for invoice printing info (e.g., business header) |

**💰 Module Prompt: Expenses**

**📄 Page Title:**

**Expenses**

**🎯 Purpose:**

This module tracks **all company expenses**, primarily driven by entries from:

* **Invoicing > Supplies**
* **Invoicing > Maintenance**
* **Invoicing > Billing**

It acts as the financial record equivalent to the **Orders** module, but instead of customer purchases, it logs **outflows of money** for products/services acquired.

**🧩 Source of Data:**

* All expenses are automatically inserted when saving an invoice under:
  + **Supplies**
  + **Maintenance**
  + **Billing**

No manual entry is allowed in this module — it's **purely for viewing, filtering, exporting**, and printing.

**🖼️ Layout:**

Full-screen table module (like Orders)

**🔍 Top Controls:**

| **Control** | **Description** |
| --- | --- |
| 🔎 Search Bar | Search by Invoice ID, Particular, Biller/Supplier Name |
| 🗂️ Filter by Source | Supplies, Maintenance, Billing |
| 📅 Filter by Date Range | Invoice Date (From - To) |
| 🏷️ Filter by Biller/Supplier | Dropdown search from linked data |
| 📤 Export | CSV/XLSX of visible data |
| 🖨️ Print Summary | A4/A5 printable summary view |

**📊 Table Columns:**

| **Column** | **Description** |
| --- | --- |
| Invoice ID | Format: I-YYYYMMDD-#### |
| Date | Invoice Date |
| Type | Source: Supplies, Maintenance, Billing |
| Particular / Description | Description of expense |
| Amount | ₱ value |
| Payment Method | Cash, GCash, Bank Transfer |
| Reference # | If GCash or Bank Transfer |
| Bank Name | If Bank Transfer |
| OR Number | Optional BIR receipt number |
| Linked Entity | Biller or Supplier Name |
| Action | View button (for invoice preview) |

**🖨️ View/Print Invoice**

Clicking **View** shows invoice print preview (from the originating module):

* Follows **A5 invoice format** with full details
* Printable directly via standard printers

**📤 Export (CSV/XLSX)**

* Export all or filtered data
* Fields included:
  + Invoice ID
  + Date
  + Type
  + Particular
  + Amount
  + Payment Method
  + Ref #
  + Bank Name
  + OR Number
  + Biller/Supplier

**🧩 Integration Map**

| **Module** | **Integration** |
| --- | --- |
| Invoicing – Supplies | Inserts expense when saved |
| Invoicing – Maintenance | Inserts expense when saved |
| Invoicing – Billing | Inserts expense when saved |
| Banks | Bank Transfer validation |
| Billers | Used in Billing entries |
| Suppliers | Used in Supplies entries |

**🧪 Scenarios to Consider**

| **Scenario** | **Expected Result** |
| --- | --- |
| Supplier invoice is saved | Expense auto-recorded |
| Maintenance with bank payment | Bank info shown in Expense |
| Filter by Billing | Only billing invoices show |
| Export all expenses this month | Export works with date range |

**✅ Revision to: Particular / Description Column**

| **Before** | **After** |
| --- | --- |
| Particular / Description | Notes / Remarks |

**📝 New Behavior:**

* Column name: **"Notes / Remarks"**
* Will display:
  + For **Supplies** or **Maintenance**: A brief note or custom summary (optional during invoice entry)
  + For **Billing**: Usually a summary or the bill period (e.g., MERALCO Bill - July 2025)
* Field will be a **multiline text field** during invoice creation, stored along with invoice metadata.

**📌 Justification:**

* **Supplies** invoices have **multiple items** — so a catch-all "Description" is not accurate.
* This aligns more with traditional invoicing structure, where **line items** belong in the detailed invoice, and a **"Notes" field** supports general context.

**🖨️ Invoice Preview Update:**

The **printable invoice view** will display:

* **Full line items** (from Supplies or Maintenance)
* **Notes** section at the bottom (populated from this field)

**📈 Module Prompt: Reports**

**📄 Page Title: Reports**

**🔰 Purpose:**

Provide summarized and exportable data across various operational areas, such as:

* Sales
* Expenses
* Inventory movements
* Customer performance
* Reward usage
* Invoicing trends

This module serves as a data review and decision-support tool, showing trends, summaries, and drill-downs.

**🖥️ Layout:**

* **Left Sidebar Panel (Filter & Report Type)** – Approx. 300px width
* **Main Report View Panel** – Flexible width, responsive, scrollable if content exceeds

No 1/3 : 2/3 layout enforced. Scroll allowed only in **Report View panel**.

**🗂️ Report Categories:**

Each report is selectable on the **left panel**, with date filters and other contextual filters (dropdowns):

| **Report Name** | **Description** |
| --- | --- |
| 🔹 Sales Report | View daily, weekly, monthly, or custom sales trends; filter by payment method, customer, order status |
| 🔹 Expenses Report | Summary of all expenses (pulled from invoices marked as Expenses) |
| 🔹 Customer Report | Customer-wise sales, orders, rewards earned and redeemed |
| 🔹 Inventory Movements | Logs of stock-ins, stock-outs, reorders; filter by item, group, or action |
| 🔹 Rewards Usage | Reward points issued, redeemed, expired; filter by date, customer |
| 🔹 Promo Usage | Which promos were used, by whom, how many times |
| 🔹 Voucher Usage | Similar to promos — view redeemed/unused/expired voucher data |
| 🔹 Invoicing Summary | Supplies, Maintenance, Billing trends with totals and invoice count |
| 🔹 On Hold Orders | List and metrics of all currently held orders and their values |

**🧮 Filters (dynamic based on selected report):**

* 📆 Date range (From, To)
* ⏳ Quick Selects: Today, This Week, This Month, Custom
* 🧑 Customer (autocomplete dropdown)
* 📦 Item / Group
* 💳 Payment Method
* 🎟️ Promo / Voucher Code
* 📄 Invoice Type: Supplies / Maintenance / Billing
* 📂 Export Type: CSV, Excel

**📊 Report View Panel:**

Each report loads:

* A **summary banner** (Total Sales, # of Orders, Avg. Value, etc.)
* A **chart** (bar, pie, or line depending on report type)
* A **table of raw data** (scrollable, paginated)

All data can be **exported**.

**📤 Export Behavior:**

* 🔘 Export only what's filtered and visible (not full database)
* ⏳ Include timestamp and applied filters in export filename
* 📄 Filename format: reportname\_YYYYMMDD-HHMMSS.csv

**⚙️ Backend API Requirements (brain/):**

| **Endpoint** | **Purpose** |
| --- | --- |
| GET /api/reports/sales | Fetch filtered sales data |
| GET /api/reports/expenses | Fetch expense records |
| GET /api/reports/customers | Customer performance stats |
| GET /api/reports/inventory | Inventory logs |
| GET /api/reports/rewards | Rewards data |
| GET /api/reports/promos | Promo usage summary |
| GET /api/reports/vouchers | Voucher redemption history |
| GET /api/reports/invoices | Invoice trends |
| GET /api/reports/holdorders | Current hold order data |

**✅ Features Recap:**

* ✅ Modular, filter-based reporting
* ✅ Charts + Tables
* ✅ CSV/Excel export
* ✅ Integrated with backend filters
* ✅ Drill-down per customer/item/date
* ✅ Covers all key modules

**🛠️ Module Prompt: Settings**

**📄 Page Title:**

Settings

**🎯 Purpose:**

Manage essential application-wide preferences that affect business identity, POS behavior, rewards logic, user access (future), printing, and system metadata — all stored locally with no cloud dependencies.

**🧱 Layout Overview:**

* **Left Panel (Sidebar – 300px fixed)**  
  Tab navigation styled with glassmorphic buttons:
  + General Settings
  + Rewards Conversion
  + Printer Config
  + ID Formats
  + System Info
  + *(Future)* User Roles
* **Right Panel (Content – 1320px width)**  
  Loads section content via AJAX  
  Scroll only enabled on right panel  
  Modular save buttons per section

**🔹 1. General Settings**

| **Field** | **Type** | **Behavior** |
| --- | --- | --- |
| Business Name | Text | Required, printed on receipts/invoices |
| Branch Name | Text | Optional |
| Contact Number | Text | Optional |
| Business Address | Textarea (3 lines) | Printed on documents |
| Timezone | Dropdown | Default: GMT+8 (Philippines); affects all dates/times |
| Currency Symbol | Text | Default: ₱ (display only, does not affect math) |

✅ **Save Button** – validates and saves locally.

**🔹 2. Rewards Conversion Settings**

| **Field** | **Description** |
| --- | --- |
| Points to Amount | Numeric – e.g., 1 Point = ₱1 |
| Effective Date | Date picker – default to current date |
| Current Display | Example: "1 Point = ₱1 since 07/01/2025" |

* “Set New Rate” opens a modal with confirmation
* History of past rates is stored in backend (for Rewards Ledger accuracy)

✅ **Conversion only affects future point redemptions.**

**🔹 3. Printer Configuration**

| **Field** | **Type** | **Details** |
| --- | --- | --- |
| POS Printer Type | Dropdown | 80mm Thermal / Half A4 / Full A4 |
| Invoice Print Size | Dropdown | Half A4 or Full A4 |
| Auto-Print on Save | Toggle | Yes / No |
| Enable Print Preview | Toggle | Yes / No |

💡 Defaults to 80mm for POS, Half A4 for invoices  
✅ Save settings persist locally and affect printing behavior across all modules.

**🔹 4. ID Format Preview (Read-only)**

Displays how system-generated IDs are formatted:

| **Entity** | **Format** |
| --- | --- |
| Customer | C-YYYYMMDD-#### |
| Order | O-YYYYMMDD-#### |
| Invoice | I-YYYYMMDD-#### |
| Supplier | S-YYYYMMDD-#### |
| Biller | B-YYYYMMDD-#### |

🔒 All IDs are auto-generated via backend; no manual entry or override.

**🔹 5. System Info (Read-only Diagnostic Panel)**

| **Field** | **Description** |
| --- | --- |
| App Version | Example: v1.0.0 |
| Last Updated | Timestamp of latest code update |
| Local DB File Path | Example: D:\Portable\_POS\_System\memory\main.db |
| Storage Used | Size in MB of local data |
| System Mode | Always "Offline" (no cloud sync) |

**🔒 *(Future Section)* User Roles & Access**

*(Disabled in v1.0.0 but UI ready)*  
Designed for future extensibility.

| **Field** | **Details** |
| --- | --- |
| User Roles | Admin, Cashier, Manager (customizable) |
| Permissions | Multi-select per module (e.g., Inventory, Orders) |
| Users | List with Username, Role, Status |
| Password Policy | Optional strength rules (future) |

**🎨 UI/UX Notes**

* Fully glassmorphic, sleek neon-blue UI
* Drag-and-drop logo upload (optional future feature)
* Section headers sticky for quick access
* Modals used for setting confirmation (e.g., conversion rate)

**🧠 Backend API Structure (brain/ folder)**

| **Method** | **Endpoint** | **Purpose** |
| --- | --- | --- |
| GET | /api/settings | Fetch all current settings |
| POST | /api/settings/update | Update any settings section |
| GET | /api/settings/conversion | View historical conversion rates |
| POST | /api/settings/conversion/update | Set new reward rate |
| GET | /api/settings/printer | Load printer preferences |
| POST | /api/settings/printer | Update printer settings |

**✅ Checklist Summary**

| **Feature** | **Status** |
| --- | --- |
| Business identity & receipt data | ✅ |
| Reward points logic | ✅ |
| POS/invoice printing behavior | ✅ |
| Secure and local-only | ✅ |
| Modular section saving | ✅ |
| AJAX module load | ✅ |
| Role & user logic (future-ready) | 🕒 Placeholder |

Would you like me to generate this as an actual HTML/CSS/JS module file or as a documentation export (e.g., Markdown/PDF)?

**🖨️ 6. Print Preview & Formatting Section**

**🎯 Purpose:**

Allow users to preview and adjust the layout of printed **Receipts (POS)** and **Invoices (Supplies, Maintenance, Billing)** before actual printing. Ideal for aligning printouts, modifying headers/footers, or toggling optional fields.

**🧱 Layout Structure:**

* **Toggle Tabs (within section):**
  + 🧾 Receipt Preview (POS)
  + 📄 Invoice Preview (A5/A4)

Each tab has:

* A **live preview pane** (rendered on-screen using CSS)
* An **editable formatting control panel**
* A **"Test Print"** button (triggers browser/system print)

**🧾 Receipt Preview (POS – 80mm Thermal)**

**🔧 Formatting Options:**

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| Store Logo | Toggle | Show/hide (if uploaded) |
| Business Header | Editable | Business Name + Address |
| Date & Time Format | Dropdown | MM/DD/YYYY hh:mm A or YYYY-MM-DD HH:mm |
| Include Customer Info | Toggle | Show "Sold To" section if customer exists |
| Include VAT Breakdown | Toggle | Shows VAT line on total |
| Footer Message | Textarea | Thank-you note, return policy, etc. |
| Show OR Number | Conditional | Required if total > ₱20 |

✅ Save settings to local file  
✅ Print preview updates live

**🔍 Preview Area:**

Simulated **80mm receipt** (ESC/POS style) with:

* Items
* Total, VAT
* Payment Method + Ref #
* Custom header/footer

🔘 Button: **[Test Print (Receipt)]**

**📄 Invoice Preview (A5 or A4)**

**🔧 Formatting Options:**

| **Field** | **Type** | **Notes** |
| --- | --- | --- |
| Print Size | Dropdown | Half A4 (A5) or Full A4 |
| Show Logo | Toggle | Top-left or center |
| Show Branch Info | Toggle | Branch name + contact |
| Header Title | Text | Default: "INVOICE" (can change to "BILL", etc.) |
| Include OR Number | Toggle | Optional |
| Include Signature Line | Toggle | Bottom of invoice |
| Show Notes Section | Toggle | From invoice metadata |
| Border Lines | Toggle | Light border under each table row |

✅ Applies to:

* Supplies Invoices
* Maintenance Invoices
* Billing Invoices

**🔍 Preview Area:**

Simulated **A5/A4 invoice layout**, showing:

* Business header
* Invoice #, Date, Supplier/Biller
* Items/services
* Totals + Optional Notes
* Signature (if enabled)

🔘 Button: **[Test Print (Invoice)]**

**🧠 Backend API (Additions):**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/settings/print | Load current print formatting settings |
| POST | /api/settings/print/update | Save updated layout settings |
| POST | /api/print/preview/receipt | Render receipt HTML |
| POST | /api/print/preview/invoice | Render invoice HTML |

**✅ Features Recap:**

* Modular toggles for formatting receipts/invoices
* Test print buttons for real layout validation
* Editable text fields for headers and footers
* Live preview in-browser (no real printer needed)
* Supports all major formats: 80mm, A5, A4

**📊 Module Prompt: Dashboard**

**📄 Page Title:**

Dashboard

**🎯 Purpose:**

Provide a real-time overview of business performance using summary widgets, trend graphs, and calendar-based insights. The Dashboard includes **four main tabs**:

1. **Sales**
2. **Expenses**
3. **Overall**
4. **Calendar View**

Each tab loads data dynamically and supports light analytics at-a-glance.

**🖥️ Layout Structure:**

* **Sidebar Tab Navigation** *(Fixed Left, 300px)*
  + Tabs: Sales | Expenses | Overall | Calendar
  + Neon-highlighted tab active
  + AJAX-style module load per tab
* **Main Panel (Right, 1320px)**
  + Content changes per tab
  + Consistent glassmorphic theme
  + All charts responsive inside 1620×1080

**🔹 1. Sales Tab**

**Header Widgets (Top Row – 1/3 height):**

* 🧾 Total Sales Today
* 💳 Sales by Payment Type (Cash / GCash / Bank)
* 📈 Highest-Selling Item
* 🕒 Latest Order Time

**Main Chart Section (2/3 height):**

* **Sales Trend Chart:** Line graph for Today / This Week / This Month
* Filter controls:
  + Date Range Picker
  + Payment Method
  + Customer (autocomplete)
* Export Button (CSV/XLSX)

**🔹 2. Expenses Tab**

**Header Widgets:**

* 💸 Total Expenses Today
* 🔍 Expense by Source (Supplies / Maintenance / Billing)
* 🏢 Highest Spending Supplier
* 🧾 Latest Invoice

**Main Chart Section:**

* **Expense Breakdown:** Pie chart by Type or Supplier
* Filter controls:
  + Date Range
  + Expense Type
  + Supplier/Biller
* Export Button (CSV/XLSX)

**🔹 3. Overall Tab**

**Widgets Section (Full Width):**

* 📊 Sales vs Expenses (Today / Week / Month)
* 🟢 Net Balance: Sales - Expenses
* ⚠️ Top 3 Expense Categories
* 🧠 Reward Points Issued vs Redeemed (mini chart)

**Chart Section:**

* **Comparison Line Chart:** Overlayed sales and expenses over time
* Filters:
  + Date range
  + Group by: Day, Week, Month

**Note:** This tab acts as a "Financial Pulse" of the business.

**🔹 4. Calendar Tab**

**Calendar View:**

* Full-width calendar with toggle for **Month / Week / Day**

**Events Displayed:**

* 🛠 Maintenance Activities (from Maintenance module)
* 🎉 Philippine Holidays (synced, read-only)
* 🔔 Alarm-type Reminders
* 🧾 High-sales days (colored heatmap style based on sales volume)

**Toolbar Controls:**

* ➕ Add Event (opens modal)
* 🔁 Sync Holidays
* 📅 View Toggle (Month, Week, Day)
* ☑️ Filters (checkboxes): Show Activities, Show Holidays, Show Sales Peaks

**📤 Export & Integration Support**

* **Each tab** supports:
  + CSV / XLSX export
  + Optional print view of summary
* **Integrated With:**
  + Orders Module (Sales)
  + Expenses Module
  + Calendar API
  + Rewards Module (for points summary)

**🧠 Backend API Suggestions**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/dashboard/sales-summary | Fetch sales stats |
| GET | /api/dashboard/expenses-summary | Fetch expenses stats |
| GET | /api/dashboard/overall-summary | Combined stats |
| GET | /api/calendar/events | Fetch calendar events |
| GET | /api/orders/trend | Sales trend chart data |
| GET | /api/expenses/trend | Expenses trend data |

Would you like this as an HTML structure next, or converted into a module file for the system?