PREMIER INSULATION - COMPREHENSIVE IMPLEMENTATION PLAN v2.0

Last Updated: December 5, 2024
Project: Quote & Job Management System
Tech Stack: Next.js 14 + React + TypeScript + Tailwind CSS + Supabase + Vercel
Repository: whizroc-premier (Private)

TABLE OF CONTENTS

---

1. EXECUTIVE SUMMARY

Business Goal

Replace legacy MYCS Windows/SQL system to save 300-500 staff hours monthly while providing:

- Easy-to-use interface for VAs and Premier sales staff
- Seamless workflow from enquiry → recommendation → quote → job → invoice
- Professional quote design and generation
- Real-time inventory management

User Types

1. **Virtual Assistants (VAs)** - Create product recommendations (NO pricing access)
2. **Premier Users (Sales Reps)** - Create quotes with pricing, convert recommendations
3. **Admin** - Full system access including inventory and settings

Core Workflow

```
Customer Contact → Premier User Creates Enquiry → Assigns to VA →
VA Creates Product Recommendation → Premier User Converts to Quote (with pricing) →
Quote Accepted → Job Created → Inventory Adjusted → Invoice Generated
```

Deployment Strategy

- **Phase 1:** Deploy to Vercel (free tier), direct URL access for users
- **Phase 2:** Embed in GoHighLevel (GHL) via iframe for unified CRM experience

---

2. CURRENT SYSTEM STATE

Folder Structure

```
src/
├── lib/
│   └── supabase.ts          ✅ Supabase client configured
├── app/
│   ├── layout.tsx           ✅ Root layout with Sidebar
│   ├── page.tsx             ✅ Landing/home page
│   ├── globals.css          ✅ Global styles
│   ├── components/
│   │   ├── Sidebar.tsx          ✅ Navigation (10 menu items)
│   │   ├── PageHeader.tsx       ✅ Reusable header component
│   │   ├── QuotesTable.tsx      ✅ Quotes list table
│   │   ├── QuoteModal.tsx       ✅ Quote detail modal
│   │   └── CustomerModal.tsx    ✅ Customer detail modal
│   ├── dashboard/
│   │   └── page.tsx         ✅ Dashboard page (basic)
│   ├── customers/
│   │   ├── page.tsx         ✅ Customers list
│   │   └── new/
│   │       └── page.tsx     ✅ Add new customer form
```

```
|   ├── quotes/
|   |   ├── page.tsx          ✅ Quotes list (with modals)
|   |   ├── new/
|   |   |   └── page.tsx      ✅ Create quote form
|   |   └── [id]/
|   |       └── page.tsx      ⚠️ Quote detail (needs verification)
|   ├── va-workspace/
|   |   ├── page.tsx          ⚠️ NEEDS MAJOR UPDATES
|   |   └── new/
|   |       └── page.tsx      ⚠️ NEEDS MAJOR UPDATES
|   └── Incidents/
|       └── page.tsx          ✅ Incidents management (H&S)
└── components/
    └── ClientDetailsForm.tsx      ✅ Reusable client form
```

Database Tables (48 Total in Supabase)

✅ Core Tables (Exist)

- `clients` - Customer records
- `companies` - Corporate client companies
- `regions` - NZ regions (Auckland metro areas)
- `sales_reps` - Sales representatives
- `app_types` / `application_types` - Insulation application types with colors
- `products` - Inventory products (SKU, R-value, pricing, etc.)
- `pricing_tiers` - Retail/Trade/VIP/Custom pricing tiers
- `job_types` - Types of jobs

✅ Enquiry Tables (EXIST - Good News!)

- `enquiries` - Main enquiry records
- `enquiry_sections` - Sections within enquiries
- `enquiry_line_items` - Line items per section
- `enquiry_tags` - Tags for categorization
- `pipeline_stages` - Enquiry pipeline stages

✅ VA Recommendation Tables (Exist)

- `product_recommendations` - VA recommendations
- `recommendation_sections` - Sections in recommendations
- `recommendation_items` - Products per section

✅ Quote Tables (Exist)

- `quotes` - Quote headers
- `quote_sections` - Sections in quotes
- `quote_line_items` - Line items per section
- Multiple views: `quote_complete_view`, `quote_details_view`, `quote_header_view`, `quote_line_items_detail`, `quote_sections_summary`

✅ Job Tables (Exist)

- `jobs` - Job records
- `job_line_items` - Actual materials used
- `job_details_view` - Consolidated job view

✅ Invoice Tables (Exist)

- `invoices` - Invoice records
- `invoice_line_items` - Invoice line items
- `invoice_aging_view` - Aging analysis
- `payments` - Payment records

✅ Inventory Tables (Exist)

- `stock_levels` - Current stock quantities
- `stock_movements` - Stock in/out transactions
- `stock_alerts` - Low stock alerts
- `suppliers` - Supplier records
- `purchase_orders` - PO headers
- `purchase_order_items` - PO line items

✅ Support Tables (Exist)

- `documents` - File attachments
- `activities` - Activity log
- `workflow_log` - Workflow tracking
- `workflow_rules` - Automation rules
- `incidents` - Safety incidents (H&S)
- `incident_actions`, `incident_attachments`, `incident_history`, `incident_witnesses`

---

3. DATABASE SCHEMA ANALYSIS

Key Observations

✅ EXCELLENT NEWS: Enquiry System Already Exists!

The database already has a complete enquiry management system:

- `enquiries` table for tracking customer enquiries
- `enquiry_sections` and `enquiry_line_items` for detailed product specifications
- `pipeline_stages` for tracking enquiry status
- This means **we don't need to create new tables** - just build the UI!

⚠️ Potential Schema Issues to Verify

1. **Enquiry → Recommendation Linking**
   - Does `product_recommendations` have an `enquiry_id` foreign key?
   - Need to verify the relationship structure
2. **User Roles & Permissions**
   - No visible `users` or `roles` table (might be using Supabase Auth metadata)
   - Need to verify RLS (Row Level Security) policies for VA vs Premier access
3. **Email Notifications**
   - No `notifications` or `email_queue` table
   - Will use Supabase Edge Function + Resend for emails
4. **Stock Reservation**
   - When quote is created, should stock be reserved?

- Need business logic for `stock_levels` updates

---

4. WHAT'S COMPLETED ✅

Pages Built and Functional

✅ Core Navigation

- **Sidebar.tsx** - 10 menu items with icons, active state highlighting
- **Layout.tsx** - Root layout with sidebar integration
- **Landing Page** - Basic home/dashboard

✅ Customers Module

- **Customers List** (`/customers`) - With search, filters, pagination
- **Add New Customer** (`/customers/new`) - Form with validation
- **Customer Modal** - Quick view customer details
- Components: `CustomerModal.tsx`, `ClientDetailsForm.tsx`

✅ Quotes Module (Partial)

- **Quotes List** (`/quotes`) - Table with 16+ columns, search, status filters
- **Create Quote** (`/quotes/new`) - Full quote creation form
- **Quote Detail** (`/quotes/[id]`) - View/edit quote (needs verification)
- **Quote Modal** - Quick view quote details
- Components: `QuotesTable.tsx`, `QuoteModal.tsx`

✅ Dashboard

- **Dashboard Page** (`/dashboard`) - Basic dashboard layout (needs enhancement)

✅ Health & Safety Module

- **Incidents Page** (`/Incidents`) - Accident and near-miss reporting

✅ Shared Components

- **PageHeader.tsx** - Consistent page headers with actions

- **Sidebar.tsx** - Navigation with Premier branding
- **ClientDetailsForm.tsx** - Reusable client input form

Technical Infrastructure

✅ **Supabase Integration**

- Client configured at `src/lib/supabase.ts`
- Connection to 48 database tables
- RLS policies (need review for VA restrictions)

✅ **Styling System**

- Tailwind CSS configured
- Global styles in `globals.css`
- Consistent blue (`🔵 #0066CC`) theme for Salesforce-style UI

✅ **TypeScript Setup**

- Proper type definitions for Quote, Customer interfaces
- Type-safe Supabase queries

---

5. WHAT NEEDS REVIEW ⚠️

Critical Items Requiring Verification

⚠️ **VA Workspace Pages (URGENT)**

Files: `src/app/va-workspace/page.tsx`, `src/app/va-workspace/new/page.tsx`

Current Issues:

1. **List Page** (`page.tsx`):
   - Only shows recommendation number, no client details
   - Missing columns: Client Name, Company, Site Address
   - No "Assigned Enquiries" section
   - Stats cards are text-only (should be visual cards)

- Missing Calculator button

2. **Create Page** (`new/page.tsx`):
   - Unknown structure/functionality
   - Needs to be rebuilt based on workflow:
     - Step 1: Select from assigned enquiries OR create new client
     - Step 2: Add sections (application types with color coding)
     - Step 3: Search products, calculate packs (NO PRICING)
     - Step 4: Review and submit

Action Required: Complete rewrite of both VA workspace pages

⚠️ Quote Pages

Files: `src/app/quotes/new/page.tsx`, `src/app/quotes/[id]/page.tsx`

Items to Verify:

1. Does quote creation integrate with `products` table for pricing tiers?
2. Is labour auto-generated ($1.50 cost, $3.00 sell per m²)?
3. Does it support pricing tiers (Retail/Trade/VIP/Custom)?
4. Is waste % (default 10%) applied to pack calculations?
5. GST 15% calculation correct?
6. PDF generation functional?

Action Required: Code review and testing

⚠️ Quote from Recommendation Flow

Expected Flow: VA submits recommendation → Premier user reviews → Converts to quote

Missing:

- Page at `/quotes/from-recommendation/[id]` to convert recommendations to quotes
- Button on recommendation detail page: "Convert to Quote"

Action Required: Build conversion workflow

⚠️ Database Relationships

Need to Verify:

1. `product_recommendations.enquiry_id` → `enquiries.id` (does this FK exist?)
2. `quotes.recommendation_id` → `product_recommendations.id` (link quotes to recommendations)
3. User roles stored in Supabase Auth metadata or separate table?
4. RLS policies prevent VAs from seeing pricing columns?

Action Required: Schema inspection and RLS policy review

⚠️ Authentication & Authorization

Current State: Unknown

Requirements:

- Supabase Auth for login
- Role-based access:
  - VA: Only VA Workspace, no pricing visibility
  - Premier: Full access except inventory management
  - Admin: Everything including settings

Action Required: Implement auth flow and role checking

---

6. WHAT NEEDS IMPLEMENTATION - PHASE 1

Priority Order

PHASE 1A: ENQUIRY SYSTEM (Week 1)

🔧 Task 1.1: Verify Enquiry Schema

Time: 2 hours

Actions:

- Query `enquiries` table structure
- Verify foreign keys to `clients`, `pipeline_stages`

- Check if `enquiry_sections` and `enquiry_line_items` are properly linked
- Document any missing columns

SQL to Run:

```sql
-- Get enquiries table structure
SELECT column_name, data_type, is_nullable
FROM information_schema.columns
WHERE table_name = 'enquiries';

-- Check foreign keys
SELECT * FROM enquiries LIMIT 5;
SELECT * FROM enquiry_sections LIMIT 5;
SELECT * FROM pipeline_stages;
```

🔧 Task 1.2: Create Enquiry Management Page

Location: `src/app/enquiries/page.tsx`
Time: 1 day

Features:

- List all enquiries (sortable table)
- Columns: Enquiry #, Client Name, Company, Site Address, Created Date, Status, Assigned To, Actions
- Search bar (30% width) - fuzzy search on client name, address, enquiry number
- Status filter buttons (New, Assigned, In Progress, Completed, Cancelled)
- Blue table header (`#0066CC`)
- Pagination at bottom
- Actions: View, Edit, Delete, Assign to VA

Components to Create:

- `src/app/components/EnquiryTable.tsx`
- `src/app/components/EnquiryModal.tsx` (quick view)

🔧 Task 1.3: Create "New Enquiry" Page

Location: `src/app/enquiries/new/page.tsx`
Time: 1 day

Form Sections:

Section 1: Client Selection

- Search existing clients (dropdown with fuzzy search)
- OR "Create New Client" inline form
- Fields: First Name, Last Name, Email, Phone, Company, Site Address, Region

Section 2: Enquiry Details

- Site address (if different from client address)
- Project description (textarea)
- Priority (dropdown: Low, Medium, High, Urgent)
- Due date (date picker)
- Notes (textarea)

Section 3: Assignment

- Assign to VA (dropdown of VA users)
- Auto-generate enquiry number (ENQ-2025-XXX)
- Status auto-set to "New"

Actions:

- [Save as Draft] - saves but doesn't notify VA
- [Create & Assign] - saves and sends email to VA

🔧 Task 1.4: Add "Enquiries" to Sidebar

Location: `src/app/components/Sidebar.tsx`
Time: 30 minutes

Changes:

```typescript
const navItems = [
  { href: '/dashboard', label: 'Dashboard', icon: Home },
  { href: '/customers', label: 'Customers', icon: Users },
  { href: '/enquiries', label: 'Enquiries', icon: ClipboardList }, // NEW
  { href: '/quotes', label: 'Quotes', icon: FileText },
  { href: '/va-workspace', label: 'VA Workspace', icon: Package },
  // ... rest
]
```

🔧 Task 1.5: Email Notification System

Location: `supabase/functions/send-enquiry-email/index.ts`
Time: 3 hours

Setup:

1. Create Supabase Edge Function
2. Install Resend SDK
3. Configure email template
4. Trigger on enquiry INSERT where `assigned_to_va_id IS NOT NULL`

Email Content:

```
Subject: New Enquiry Assigned - [Client Name]

Hi [VA Name],

You have been assigned a new enquiry:

Enquiry #: ENQ-2025-XXX
Client: [Client Name]
Company: [Company Name]
Site Address: [Address]
Priority: [Priority]

[View Enquiry in Premier App]

Thanks,
Premier Insulation System
```

Environment Variables:

- `RESEND_API_KEY`

PHASE 1B: VA WORKSPACE REBUILD (Week 2)

🔧 Task 2.1: Rebuild VA Workspace List Page

Location: `src/app/va-workspace/page.tsx`
Time: 1 day

Layout Changes:

Header Section:

- Title: "Product Recommendations"
- Buttons: [Calculator] [Create Recommendation]

Stats Cards (Visual):

```
┌──────────────┐ ┌──────────────┐ ┌──────────────┐ ┌──────────────┐
│ Created: 10  │ │ Draft: 3     │ │ Submitted: 5 │ │ Awaiting: 2  │
└──────────────┘ └──────────────┘ └──────────────┘ └──────────────┘
```

Search Bar: 30% width, fuzzy search

Status Filter Buttons: All, Draft, Submitted, Awaiting Review

Section 1: MY ASSIGNED ENQUIRIES

- Pull enquiries where `assigned_to_va_id = current_user_id` AND `status != 'Completed'`
- Display: Enquiry #, Client Name, Company, Site Address, Priority, Due Date
- Button: [CREATE RECOMMENDATION →]

Section 2: MY RECOMMENDATIONS

- Table columns: Rec #, Client, Company, Site Address, Created Date, Status
- Actions: [Edit] [Delete] (if Draft) OR [View] (if Submitted/Converted)

Database Query:

```
typescript
```

```
// Get assigned enquiries
const { data: enquiries } = await supabase
  .from('enquiries')
  .select(`
    id,
    enquiry_number,
    site_address,
    priority,
    due_date,
    clients (first_name, last_name, companies (name))
  `)
  .eq('assigned_to_va_id', currentUserId)
  .neq('status', 'Completed');

// Get my recommendations
const { data: recommendations } = await supabase
  .from('product_recommendations')
  .select(`
    id,
    recommendation_number,
    version,
    status,
    created_at,
    clients (first_name, last_name, address_line_1, companies (name))
  `)
  .eq('created_by_user_id', currentUserId)
  .order('created_at', { ascending: false });
```

🔧 Task 2.2: Create Recommendation Form (from Enquiry)

Location: `src/app/va-workspace/new/page.tsx`
Time: 2 days

URL Pattern: `/va-workspace/new?enquiry_id=xxx`

Step 1: Pre-fill Client Data

- If `enquiry_id` in URL, fetch enquiry and client data
- Auto-populate: Client Name, Company, Site Address, Region
- Display enquiry details at top (read-only)

Step 2: Add Sections

```
┌─ SECTION 1 ─────────────────────────┐
│ Application Type: [External Walls ▼]    │ ← Dropdown with color indicator
│ OR Custom Name: [_____________]         │ ← If not standard type
│                                     │
│ ┌─ PRODUCTS ───────────────────────┐   │
│ │ Area m² │ Product Search │ Packs │   │
│ │   89    │ [R2.4 90mm 🔍] │  10   │   │ ← Fuzzy search
│ │         │ Bale: 8.81m²   │       │   │ ← Auto-calc
│ │         │ Stock: ✓ In Stock      │   │ ← Stock status
│ │                                  │   │
│ │ [+ Add Product]                  │   │
│ └──────────────────────────────────┘   │
│                                     │
│ [Remove Section]                        │
└─────────────────────────────────────┘


[+ Add Section]
```

Key Features:

- Product search: Fuzzy search by R-value, SKU, description
- Auto-calculate packs: `CEIL((area * (1 + waste%)) / bale_size)`
- Show waste % from products table (default 10%, editable)
- Stock status indicators:
  - ✓ In Stock (green)
  - ⚠️ Low Stock (orange)
  - ❌ Out of Stock (red)
- NO PRICING VISIBLE (no cost, sell price, or margin)

Step 3: Review & Submit

- Summary of all sections and products
- Total packs per product
- Total area covered
- Stock warnings

Actions:

- [Save Draft] - status = 'Draft'
- [Submit for Review] - status = 'Submitted', notify Premier user

---

🔧 Task 2.3: Recommendation Detail/Edit Page

Location: `src/app/va-workspace/[id]/page.tsx`
Time: 1 day

Features:

- View mode if status = 'Submitted' or 'Converted'
- Edit mode if status = 'Draft'
- Same layout as create page
- Show history: Created date, submitted date, converted date (if applicable)

PHASE 1C: PREMIER QUOTE CONVERSION (Week 2-3)

🔧 Task 3.1: Review Existing Quote Pages

Files: `src/app/quotes/new/page.tsx`, `src/app/quotes/[id]/page.tsx`
Time: 4 hours

Checklist:

- [ ] Verify connection to `quotes` table
- [ ] Check pricing tier implementation (Retail/Trade/VIP/Custom)
- [ ] Verify labour auto-generation ($1.50 cost, $3.00 sell per m²)
- [ ] Check waste % application (default 10%)
- [ ] Test GST 15% calculation
- [ ] Verify PDF generation (Template A/B)
- [ ] Check version control (Q-001.01, Q-001.02)

If Issues Found: Document and add to task list

🔧 Task 3.2: Create "Convert Recommendation to Quote" Page

Location: `src/app/quotes/from-recommendation/[id]/page.tsx`
Time: 1 day

Flow:

1. Load recommendation data (client, sections, products, areas, packs)
2. Pre-populate quote form with recommendation data
3. Add Premier-specific fields:
   - Pricing tier selector (Retail/Trade/VIP/Custom)
   - Labour rate (default $3.00/m², editable)
   - Discount % (optional)
   - Quote validity days (default 30)
   - Sales rep selection
   - PDF template selection (A/B/Both)
4. System auto-calculates:
   - Product sell prices based on tier
   - Labour rows (area × $1.50 cost, area × $3.00 sell)
   - Subtotals per section
   - Total Ex GST
   - GST amount (15%)
   - Total Inc GST
   - Gross Profit %
5. Actions:
   - [Save Draft]
   - [Generate Quote] - creates quote, updates recommendation status to "Converted to Quote"

🔧 Task 3.3: Add "Convert to Quote" Button

Location: `src/app/va-workspace/[id]/page.tsx`
Time: 1 hour

Conditions:

- Only show if:
  - User role = 'Premier' or 'Admin'
  - Recommendation status = 'Submitted'
  - Not already converted

Button:

```tsx
{canConvert && (
  <button
    onClick={() => router.push(`/quotes/from-recommendation/${recommendationId}`)}
    className="px-4 py-2 bg-[#0066CC] text-white rounded-lg"
  >
    Convert to Quote →
  </button>
)}
```

PHASE 1D: INVENTORY MANAGEMENT (Week 3)

🔧 **Task 4.1: Create Inventory Page**

Location: `src/app/inventory/page.tsx`
Time: 1 day

Layout: Three Tabs

Tab 1: Products

- List all products from `products` table
- Columns: SKU, Product Name, Category, R-Value, Current Stock, Reorder Level, Status
- Stock level indicators:
  - Green: Stock > Reorder Level
  - Orange: Stock ≤ Reorder Level (⚠️ Low Stock)
  - Red: Stock = 0 (❌ Out of Stock)
- Actions: [Add Stock] [Subtract Stock] [Edit Product]

Tab 2: Purchase Orders

- List from `purchase_orders` table
- Columns: PO #, Supplier, Order Date, Expected Date, Status, Total
- Actions: [New PO] [Receive PO] [View Details]

Tab 3: Suppliers

- List from `suppliers` table
- Columns: Name, Contact, Phone, Email, Payment Terms
- Actions: [Add Supplier] [Edit] [Delete]

🔧 Task 4.2: Stock Adjustment Modals

Components: `src/app/components/StockAdjustmentModal.tsx`
Time: 3 hours

Add Stock Modal:

- Product name (display only)
- Current stock (display only)
- Quantity to add (input)
- Reference (PO number or note)
- Update `stock_levels.quantity`
- Insert into `stock_movements` (movement_type = 'receive')

Subtract Stock Modal:

- Same as above but negative quantity
- Movement type = 'adjustment' or 'sale'

🔧 Task 4.3: Stock Alerts Integration

Location: Multiple pages
Time: 2 hours

Quote Creation Page:

- When product is selected, check `stock_levels`
- Show warning if stock < required packs
- Allow override with note

VA Recommendation Page:

- Show stock status for each product
- Warning if insufficient stock

Dashboard:

- Widget showing low stock alerts from `stock_alerts` table

PHASE 1E: AUTHENTICATION & AUTHORIZATION (Week 3-4)

🔧 Task 5.1: Implement Supabase Auth

Location: `src/app/login/page.tsx`, `src/lib/auth.ts`
Time: 1 day

Features:

- Email/password login
- Password reset flow
- Session management
- Protected routes (middleware)

🔧 Task 5.2: Role-Based Access Control

Location: `src/lib/rbac.ts`
Time: 1 day

User Roles:

```typescript
```

```
enum UserRole {
  VA = 'va',
  PREMIER = 'premier',
  ADMIN = 'admin'
}
```

Access Matrix:

Page	VA	Premier	Admin
Dashboard	✅	✅	✅
Customers	❌	✅	✅
Enquiries	❌	✅	✅
VA Workspace	✅	✅	✅
Quotes	❌	✅	✅
Jobs	❌	✅	✅
Inventory	❌	❌	✅
Settings	❌	❌	✅

Implementation:

- Store role in Supabase Auth user metadata
- Create `useAuth()` hook to check role
- Create `<ProtectedRoute>` component
- Conditional rendering based on role

🔧 Task 5.3: RLS Policies for Pricing Protection

Location: Supabase Dashboard → Authentication → Policies
Time: 2 hours

Critical Policies:

Products Table:

```sql
sql
```

```sql
-- VAs can see products but NOT pricing columns
CREATE POLICY "VAs can read products without pricing"
ON products FOR SELECT
TO authenticated
USING (
  auth.jwt() ->> 'role' = 'va'
)
WITH CHECK (
  -- Hide: cost_price, retail_price, pack_price columns
  -- Only show: sku, name, r_value, bale_size, waste_percentage
);

-- Premier users can see all product data
CREATE POLICY "Premier can read all product data"
ON products FOR SELECT
TO authenticated
USING (
  auth.jwt() ->> 'role' IN ('premier', 'admin')
);
```

Quotes Table:

```sql
-- VAs cannot access quotes at all
CREATE POLICY "VAs cannot access quotes"
ON quotes FOR SELECT
TO authenticated
USING (
  auth.jwt() ->> 'role' IN ('premier', 'admin')
);
```

PHASE 1F: TESTING & POLISH (Week 4)

🔧 Task 6.1: End-to-End Workflow Testing

Time: 2 days

Test Scenarios:

Scenario 1: Full Enquiry → Quote Flow

1. Premier user creates enquiry, assigns to VA
2. VA receives email notification
3. VA logs in, sees enquiry in "Assigned Enquiries"
4. VA creates recommendation with products
5. VA submits recommendation
6. Premier user receives notification
7. Premier user converts to quote with pricing
8. Quote PDF generated successfully

Scenario 2: Stock Management

1. Admin adds new product with stock level
2. VA creates recommendation using product
3. Stock warning shows if low
4. Premier converts to quote
5. Admin adjusts stock (simulate delivery)
6. Stock alerts update

Scenario 3: Role-Based Access

1. VA attempts to access Quotes page → Blocked
2. VA views product in recommendation → No prices shown
3. Premier views same product in quote → Prices visible
4. VA attempts to access Inventory → Blocked

🔧 Task 6.2: UI/UX Polish

Time: 2 days

Checklist:

- [ ] All pages follow master template (blue headers, 30% search, etc.)
- [ ] Loading states on all async operations
- [ ] Error messages user-friendly

- [ ] Success toasts after actions
- [ ] Mobile responsiveness (basic)
- [ ] Consistent button styles
- [ ] Proper form validation
- [ ] Confirmation dialogs for destructive actions

🔧 Task 6.3: Data Import

Time: 1 day

Required Data:

- [ ] Import existing clients (CSV → `clients` table)
- [ ] Import products (CSV → `products` table)
- [ ] Import stock levels (CSV → `stock_levels` table)
- [ ] Create initial users (VA, Premier, Admin roles)
- [ ] Set up regions (Auckland metro areas)
- [ ] Configure pricing tiers

---

7. FUTURE PHASES

PHASE 2: GHL INTEGRATION (Week 5-6)

Features:

- n8n workflow: GHL contact → Supabase client sync
- n8n workflow: New enquiry → GHL opportunity
- n8n workflow: Quote accepted → GHL deal won
- Webhook handlers in Next.js
- Bidirectional data sync

Deployment:

- Embed Premier app in GHL iframe
- Custom domain setup
- SSO if needed

PHASE 3: ADVANCED FEATURES (Week 7+)

Jobs Module:

- Convert quote to job
- Job scheduling
- Crew assignment
- Photo upload (before/after)
- Completion certificates
- Actual vs quoted materials tracking

Invoicing:

- Auto-generate from completed jobs
- Payment tracking
- Overdue reminders
- Export to Xero/MYOB

Analytics Dashboard:

- Sales metrics
- VA performance
- Stock turnover
- Quote conversion rates
- Revenue forecasting

Mobile App:

- React Native or PWA
- Field crew access
- Photo capture
- Job completion on-site

---

8. TECHNICAL ARCHITECTURE

Deployment

```
┌─────────────────────────────────────────────┐
│              VERCEL                         │
│  ┌───────────────────────────────────────┐  │
│  │ Next.js 14 App (React + TypeScript)   │  │
│  │ - Server Components                   │  │
│  │ - Client Components                   │  │
│  │ - API Routes                          │  │
│  └───────────────────────────────────────┘  │
└─────────────────────────────────────────────┘
              ↕️
┌─────────────────────────────────────────────┐
│            SUPABASE                         │
│  ┌───────────────────────────────────────┐  │
│  │ PostgreSQL Database (48 tables)       │  │
│  │ - RLS Policies                        │  │
│  │ - Foreign Keys                        │  │
│  │ - Views                               │  │
│  ├───────────────────────────────────────┤  │
│  │ Auth (JWT)                            │  │
│  │ - Email/Password                      │  │
│  │ - User Metadata (roles)               │  │
│  ├───────────────────────────────────────┤  │
│  │ Edge Functions                        │  │
│  │ - Email notifications                 │  │
│  │ - Webhook handlers                    │  │
│  ├───────────────────────────────────────┤  │
│  │ Storage                               │  │
│  │ - PDFs, images, documents             │  │
│  └───────────────────────────────────────┘  │
└─────────────────────────────────────────────┘
              ↕️
┌─────────────────────────────────────────────┐
│             RESEND                          │
│ - Email delivery (VA notifications)         │
│ - Quote emails to clients                   │
└─────────────────────────────────────────────┘
```

Future Architecture (Phase 2)

```
┌─────────────────────────────────────────────┐
│          GOHIGHLEVEL (GHL)                  │
│  - CRM & Conversations                      │
│  - Marketing automation                     │
│  - Premier App embedded (iframe)            │
└─────────────────────────────────────────────┘
              ↕️
┌─────────────────────────────────────────────┐
│        n8n (Self-hosted Whizrock)           │
│  - GHL → Supabase sync workflows            │
│  - Supabase → GHL sync workflows            │
│  - Email automation                         │
└─────────────────────────────────────────────┘
```

---

9. DEVELOPMENT WORKFLOW

Git Workflow

```bash
# Current branch
main (production)

# Feature branches
feature/enquiry-system
feature/va-workspace-rebuild
feature/quote-conversion
feature/inventory-management
feature/auth-rbac

# Workflow
1. Create feature branch from main
2. Develop and test locally
3. Push to GitHub
4. Vercel auto-deploys preview
5. Test preview deployment
6. Merge to main → Production deployment
```

Local Development

```bash
# Install dependencies
npm install

# Run dev server
npm run dev

# Build for production
npm run build

# Run Supabase locally (optional)
supabase start
supabase db reset  # Reset local DB
```

Environment Variables

```env
# .env.local
NEXT_PUBLIC_SUPABASE_URL=https://xxx.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=eyJxxx...
SUPABASE_SERVICE_ROLE_KEY=eyJxxx...  # Server-side only
RESEND_API_KEY=re_xxx...
```

Testing Checklist

- [ ] All pages load without errors
- [ ] Database queries return expected data
- [ ] RLS policies block unauthorized access
- [ ] Email notifications send successfully
- [ ] PDF generation works
- [ ] Stock calculations accurate
- [ ] Role-based UI rendering correct

---

10. IMMEDIATE NEXT STEPS (This Week)

Priority 1: Database Verification (TODAY)

1. ✅ Run schema inspection queries

2. ✅ Verify `enquiries` table structure
3. ✅ Check foreign key relationships
4. ✅ Document any missing columns/tables

Priority 2: Enquiry System (Days 1-2)

1. Create Enquiry list page
2. Create "New Enquiry" form
3. Add to Sidebar
4. Set up Resend account
5. Create email notification edge function

Priority 3: VA Workspace (Days 3-4)

1. Rebuild VA Workspace list page
2. Add "Assigned Enquiries" section
3. Create recommendation form (from enquiry)
4. Test end-to-end VA workflow

Priority 4: Quote Conversion (Day 5)

1. Review existing quote pages
2. Create "Convert to Quote" page
3. Test conversion workflow

Priority 5: Inventory (Days 6-7)

1. Create Inventory page with 3 tabs
2. Add stock adjustment modals
3. Integrate stock checks in quote/recommendation pages

APPENDIX A: KEY DESIGN DECISIONS

✅ Confirmed Decisions

1. **Frontend:** Next.js deployed on Vercel (free tier)

2. **Database:** Supabase as single source of truth
3. **Email:** Resend for notifications
4. **Enquiry Structure:** Separate `enquiries` table (not status on client)
5. **VA Workflow:** Enquiry → Recommendation → Quote (with Premier conversion)
6. **Automation:** n8n (self-hosted Whizrock) for GHL sync (Phase 2)
7. **Access Control:** Role-based (VA, Premier, Admin) with RLS policies
8. **Deployment Strategy:** Phase 1 = Direct URL, Phase 2 = GHL iframe

---

APPENDIX B: CONTACTS & RESOURCES

Key Stakeholders

- **Client:** Premier Insulation (New Zealand)
- **Developer:** Sukumar (with Claude AI assistance)
- **Users:** 5 staff (VAs + Premier sales reps)

Technical Resources

- **GitHub Repo:** whizroc-premier (Private)
- **Supabase Project:** [URL needed]
- **Vercel Project:** [URL needed]
- **n8n Instance:** [URL needed]
- **Domain:** [URL needed]

Documentation

- Implementation Guide (Google Doc)
- Extended System Design (Google Doc)
- This Implementation Plan (Markdown)

DOCUMENT HISTORY

Version	Date	Author	Changes
1.0	Nov 24, 2024	Sukumar	Initial implementation guide
2.0	Dec 5, 2024	Sukumar + Claude	Comprehensive plan after code review

---

END OF IMPLEMENTATION PLAN v2.0

QUICK REFERENCE: FILE LOCATIONS

Pages to Create

```
✅ src/app/enquiries/page.tsx
✅ src/app/enquiries/new/page.tsx
✅ src/app/enquiries/[id]/page.tsx
⚠️ src/app/va-workspace/page.tsx (REBUILD)
⚠️ src/app/va-workspace/new/page.tsx (REBUILD)
✅ src/app/quotes/from-recommendation/[id]/page.tsx
✅ src/app/inventory/page.tsx
✅ src/app/login/page.tsx
```

Components to Create

```
✅ src/app/components/EnquiryTable.tsx
✅ src/app/components/EnquiryModal.tsx
✅ src/app/components/StockAdjustmentModal.tsx
✅ src/app/components/ProtectedRoute.tsx
```

Utilities to Create

```
✅ src/lib/auth.ts
✅ src/lib/rbac.ts
✅ src/lib/email.ts (Resend wrapper)
```

Supabase Functions

- [x] supabase/functions/send-enquiry-email/index.ts

This document should be referenced for all development sessions to maintain consistency and track progress.