Phase 3: Sales Capture Module - Pull Request Summary

Status: Ready for Manual Push and PR Creation

All code has been committed to the local branch `feature/phase-3-sales-module`. The GitHub token authentication is not working in this environment, so you'll need to manually push and create the PR.

Manual Steps Required

1. Push the Branch

```
cd /path/to/citaplanner
git checkout feature/phase-3-sales-module
git push origin feature/phase-3-sales-module
```

2. Create Pull Request on GitHub

Go to: https://github.com/qhosting/citaplanner/pulls

Click "New Pull Request" and use the following details:

Pull Request Title

```
Phase 3: Sales Capture Module - POS, Inventory, Commissions & Reports
```

Pull Request Description

```
# Phase 3: Sales Capture Module

This PR implements a comprehensive Sales Capture Module for CitaPlanner, adding Point
of Sale (POS), inventory management, commission tracking, and reporting capabilities.

## 🎯 Overview

The Sales Capture Module enables businesses to:
- Process sales of products and services through a POS system
- Manage product inventory with automatic stock tracking
- Calculate and track professional commissions
- Generate comprehensive sales and inventory reports
- Monitor business performance through a sales dashboard

## 📊 Database Changes

### New Tables (7)
- ✅ `product_categories` - Product categorization
- ✅ `suppliers` - Supplier management
- ✅ `products` - Product catalog with inventory
- ✅ `sales` - Sales transactions
- ✅ `sale_items` - Line items (products/services)
- ✅ `inventory_movements` - Stock movement tracking
- ✅ `professional_commissions` - Commission aggregation

### New Enums (6)
- `ProductType`: SALE, INTERNAL
- `ProductUnit`: PIECE, SERVICE
- `SaleStatus`: PENDING, COMPLETED, CANCELLED
- `SalePaymentMethod`: CASH, CARD, TRANSFER, OTHER
- `MovementType`: IN, OUT, ADJUSTMENT
- `CommissionStatus`: PENDING, PAID

### Migration File
- `20251008005257_phase3_sales_module/migration.sql`

## 🔧 Business Logic Services (5 new)

### 1. ProductService (`src/services/productService.ts`)
- Product CRUD operations
- Stock validation and management
- Low stock alerts
- Automatic stock updates

### 2. SaleService (`src/services/saleService.ts`)
- Sale creation with transaction handling
- Automatic inventory deduction
- Commission calculation
- Sale cancellation with reversals
- Unique sale number generation (YYYYMMDD####)

### 3. InventoryService (`src/services/inventoryService.ts`)
- Movement tracking (IN, OUT, ADJUSTMENT)
- Stock adjustments
- Inventory reports with values

### 4. CommissionService (`src/services/commissionService.ts`)
- Automatic commission aggregation by period
- Commission tracking (pending/paid)
- Professional summaries

### 5. ReportService (`src/services/reportService.ts`)
```

```markdown
- Sales reports with filters
- Top products/services analysis
- Professional performance metrics
- Grouping by day/week/month

## 🌐 API Endpoints (25+ new)

### Products
- `GET/POST /api/products` - List and create
- `GET/PUT/DELETE /api/products/[id]` - Manage product
- `GET /api/products/low-stock` - Low stock alerts

### Suppliers
- `GET/POST /api/suppliers` - List and create
- `GET/PUT/DELETE /api/suppliers/[id]` - Manage supplier

### Product Categories
- `GET/POST /api/product-categories` - Category management

### Sales
- `GET/POST /api/sales` - List and create sales
- `GET /api/sales/[id]` - Sale details
- `POST /api/sales/[id]/cancel` - Cancel with reversals

### Inventory
- `GET /api/inventory/movements` - Movement history
- `POST /api/inventory/adjustment` - Stock adjustments
- `GET /api/inventory/stock-report` - Stock report

### Commissions
- `GET /api/commissions` - List with filters
- `POST /api/commissions/[id]/pay` - Mark as paid
- `GET /api/commissions/summary` - Professional summary

### Reports
- `GET /api/reports/sales` - Comprehensive sales reports
- `GET /api/reports/top-products` - Top products
- `GET /api/reports/top-services` - Top services
- `GET /api/reports/professional-performance` - Performance

### Dashboard
- `GET /api/dashboard/sales` - Sales dashboard KPIs

## 🎨 UI Components (3 new pages)

### 1. Point of Sale (`app/(dashboard)/sales/pos/page.tsx`)
- Quick product/service selection
- Client selection (CRM integration)
- Professional assignment for services
- Real-time cart management
- Multiple payment methods
- Discount and tax support
- Sale completion

### 2. Product Management (`app/(dashboard)/inventory/products/page.tsx`)
- Product list with search
- Stock level indicators
- Low stock alerts
- Quick actions (view, edit)

### 3. Sales Dashboard (`app/(dashboard)/dashboard/sales/page.tsx`)
- Key metrics (sales, average, commissions)
- Period selection (day/week/month)
```

~~~
- Top products and services
- Low stock alerts
- Pending commissions

## ✨ Key Features

### Automatic Processes
- ✅ Stock deduction on sale completion
- ✅ Inventory movement recording
- ✅ Commission calculation and aggregation
- ✅ Low stock detection
- ✅ Unique sale number generation

### Transaction Safety
- ✅ Database transactions for sales
- ✅ Automatic rollback on errors
- ✅ Stock validation before sale
- ✅ Reversals on cancellation

### Integration
- ✅ Uses services from Phase 1 catalog
- ✅ Links to clients from Phase 2 CRM
- ✅ Multi-tenant architecture maintained
- ✅ Non-breaking changes

## 📚 Documentation

- ✅ `docs/PHASE_3_SALES_MODULE.md` - Complete module documentation
- ✅ `PHASE_3_DEPLOYMENT.md` - Deployment guide with rollback plan
- ✅ API endpoint documentation
- ✅ Database schema documentation
- ✅ User guides

## 🚀 Deployment Instructions

### 1. Run Database Migration

```bash
cd app
npx prisma migrate deploy
npx prisma generate
~~~

2. Restart Application

```
# PM2
pm2 restart citaplanner

# Docker
docker-compose restart

# Easypanel
# Restart via dashboard
```

3. Verify Deployment

- [ ] Check API endpoints respond
- [ ] Test POS page loads
- [ ] Test product management
- [ ] Test dashboard displays metrics

4. Initial Setup (Optional)

- Create product categories
- Add suppliers
- Add initial products
- Configure commission rates

✅ Testing Checklist

- [ ] Create product with stock
- [ ] Complete sale with product (verify stock deduction)
- [ ] Complete sale with service (verify commission)
- [ ] Cancel sale (verify reversals)
- [ ] Create stock adjustment
- [ ] View low stock alerts
- [ ] Generate sales report
- [ ] View dashboard metrics
- [ ] Mark commission as paid

🔒 Non-Breaking Changes

- ✅ All new tables and enums
- ✅ Optional relationships to existing tables
- ✅ No modifications to core functionality
- ✅ Backward compatible with Phases 1 and 2

📈 Performance Considerations

- ✅ Indexes added for common queries
- ✅ Transaction-based sale creation
- ✅ Efficient aggregation for commissions
- ✅ Optimized report queries

🔄 Rollback Plan

If issues occur, see `PHASE_3_DEPLOYMENT.md` for detailed rollback instructions.

Quick rollback:

```
git revert HEAD
npx prisma migrate resolve --rolled-back 20251008005257_phase3_sales_module
pm2 restart citaplanner
```

📝 Files Changed

- **56 files changed**
- **8,333 insertions**
- **2,041 deletions**

New Files

- 7 database tables (migration)
- 5 service layer files
- 25+ API route files
- 3 UI page components
- 1 TypeScript types file
- 2 documentation files

🎉 Ready for Review

This PR is ready for review and production deployment. All code follows existing patterns, includes comprehensive error handling, and maintains backward compatibility.

Related Issues: N/A
Depends On: Phase 1 (PR #71), Phase 2 (PR #72)
Breaking Changes: None
Database Migration: Yes (see deployment instructions)
\`\`\`

---

3. After Creating PR

Review the PR

- Check all files are included
- Verify documentation is complete
- Review code changes

Merge When Ready

After review and approval:
1. Merge the PR to main
2. Deploy to production following `PHASE_3_DEPLOYMENT.md`
3. Run database migration
4. Verify all features work

Files Summary

Commit Details

- **Branch:** `feature/phase-3-sales-module`
- **Commit:** `914094b`
- **Files Changed:** 56
- **Insertions:** 8,333
- **Deletions:** 2,041

Key Files Added

1. **Database Schema:** `app/prisma/schema.prisma` (modified)
2. **Migration:** `app/prisma/migrations/20251008005257_phase3_sales_module/migration.sql`
3. **Services:** 5 new service files in `app/src/services/`
4. **API Routes:** 25+ new route files in `app/api/`
5. **UI Pages:** 3 new page components in `app/(dashboard)/`
6. **Types:** `app/src/types/sales.ts`
7. **Documentation:** `docs/PHASE_3_SALES_MODULE.md`, `PHASE_3_DEPLOYMENT.md`

Contact

If you need assistance with the deployment or have questions about the implementation, refer to:
- `docs/PHASE_3_SALES_MODULE.md` - Complete documentation
- `PHASE_3_DEPLOYMENT.md` - Deployment guide

Status: ✅ Code Complete - Ready for Manual Push and PR Creation
Next Step: Push branch and create PR on GitHub