Phase 3: Sales Capture Module - Deployment Guide

Overview

This document provides step-by-step instructions for deploying Phase 3 (Sales Capture Module) to production.

Pre-Deployment Checklist

- [ ] All code changes reviewed and tested locally
- [ ] Database backup created
- [ ] Migration file reviewed: `20251008005257_phase3_sales_module`
- [ ] Environment variables verified
- [ ] Downtime window scheduled (if needed)

Deployment Steps

1. Merge Pull Request

```
# Review and merge PR #73 on GitHub
# Title: "Phase 3: Sales Capture Module - POS, Inventory, Commissions & Reports"
```

2. Pull Latest Changes on Server

```
# SSH into your Easypanel server
ssh user@your-server.com

# Navigate to app directory
cd /path/to/citaplanner

# Pull latest changes
git pull origin main
```

3. Run Database Migration

```
# Navigate to app directory
cd app

# Run Prisma migration
npx prisma migrate deploy

# Generate Prisma client
npx prisma generate
```

Expected output:

```
Applying migration `20251008005257_phase3_sales_module`
The following migration(s) have been applied:
migrations/
  └─ 20251008005257_phase3_sales_module/
    └─ migration.sql
```

4. Verify Database Schema

```
# Connect to PostgreSQL
psql $DATABASE_URL

# Verify new tables exist
\dt

# Should show:
# - product_categories
# - suppliers
# - products
# - sales
# - sale_items
# - inventory_movements
# - professional_commissions

# Verify new enums
\dT+

# Should show:
# - ProductType
# - ProductUnit
# - SaleStatus
# - SalePaymentMethod
# - MovementType
# - CommissionStatus

# Exit psql
\q
```

5. Restart Application

```
# If using PM2
pm2 restart citaplanner

# If using Docker
docker-compose restart

# If using Easypanel
# Restart via Easypanel dashboard
```

6. Verify Deployment

Check API Endpoints

```
# Test products endpoint
curl -X GET https://your-domain.com/api/products \
  -H "Cookie: your-session-cookie"

# Test sales endpoint
curl -X GET https://your-domain.com/api/sales \
  -H "Cookie: your-session-cookie"

# Test dashboard endpoint
curl -X GET https://your-domain.com/api/dashboard/sales?period=month \
  -H "Cookie: your-session-cookie"
```

Check UI Pages

Visit the following pages in your browser:
- [ ] `/sales/pos` - Point of Sale
- [ ] `/inventory/products` - Product Management
- [ ] `/inventory/movements` - Inventory Movements
- [ ] `/dashboard/sales` - Sales Dashboard
- [ ] `/reports/sales` - Sales Reports
- [ ] `/commissions` - Commission Management

7. Initial Data Setup (Optional)

Create Product Categories

```
# Via API or UI
POST /api/product-categories
{
  "name": "Hair Care",
  "description": "Hair care products"
}
```

Create Suppliers

```
# Via API or UI
POST /api/suppliers
{
  "name": "Beauty Supplies Inc",
  "contactPerson": "John Doe",
  "phone": "+1234567890",
  "email": "contact@beautysupplies.com"
}
```

Add Products

```
# Via API or UI
POST /api/products
{
  "name": "Shampoo Premium",
  "sku": "SHP-001",
  "type": "SALE",
  "unit": "PIECE",
  "stock": 50,
  "minStock": 10,
  "costPrice": 150.00,
  "salePrice": 250.00,
  "categoryId": "cat_xxx",
  "supplierId": "sup_xxx"
}
```

Post-Deployment Verification

1. Test Complete Sale Flow

1. Navigate to `/sales/pos`
2. Add a product to cart
3. Add a service with professional
4. Complete sale
5. Verify:
   - Sale created successfully
   - Product stock deducted
   - Inventory movement recorded
   - Commission recorded (for service)

2. Test Inventory Management

1. Navigate to `/inventory/products`
2. Check product stock levels
3. Create stock adjustment
4. Verify movement recorded

3. Test Reports

1. Navigate to `/reports/sales`
2. Generate sales report for current month
3. Verify data accuracy

4. Test Dashboard

1. Navigate to `/dashboard/sales`
2. Verify metrics display correctly
3. Check low stock alerts
4. Check pending commissions

Rollback Plan

If issues occur, rollback using these steps:

1. Revert Code Changes

```
# Revert to previous commit
git revert HEAD
git push origin main
```

2. Rollback Database Migration

```
# This will remove the Phase 3 tables and enums
npx prisma migrate resolve --rolled-back 20251008005257_phase3_sales_module

# Then apply previous migration state
npx prisma migrate deploy
```

3. Restart Application

```
pm2 restart citaplanner
# or
docker-compose restart
```

Troubleshooting

Migration Fails

Error: "Column already exists"

Solution: Check if migration was partially applied. Manually fix schema and mark migration as resolved.

```
npx prisma migrate resolve --applied 20251008005257_phase3_sales_module
```

API Endpoints Return 500

Error: "PrismaClient is not configured"

Solution: Regenerate Prisma client

```
npx prisma generate
pm2 restart citaplanner
```

UI Pages Not Loading

Error: "Module not found"

Solution: Rebuild Next.js application

```
npm run build
pm2 restart citaplanner
```

Stock Not Deducting

Error: Sale completes but stock unchanged

Solution: Check inventory service logs and verify transaction completion

```
# Check application logs
pm2 logs citaplanner

# Verify database transaction
psql $DATABASE_URL
SELECT * FROM inventory_movements ORDER BY "createdAt" DESC LIMIT 10;
```

Monitoring

Key Metrics to Monitor

1. **API Response Times**
   - `/api/sales` POST endpoint
   - `/api/products` GET endpoint
   - `/api/dashboard/sales` GET endpoint

2. **Database Performance**
   - Query execution times
   - Connection pool usage
   - Lock contention

3. **Error Rates**
   - Failed sale transactions
   - Stock validation errors
   - Commission calculation errors

Logging

Monitor application logs for:
- Sale creation errors
- Inventory deduction failures
- Commission calculation issues
- Database connection problems

```
# View real-time logs
pm2 logs citaplanner --lines 100

# Search for errors
pm2 logs citaplanner | grep ERROR
```

Support

If you encounter issues:

1. Check this deployment guide
2. Review application logs
3. Check database schema
4. Verify environment variables
5. Contact development team

Changelog

Phase 3 - Initial Release (2025-10-08)

New Features:
- Point of Sale (POS) system
- Product and inventory management
- Supplier management
- Commission tracking and calculation
- Sales reporting and analytics
- Sales dashboard with key metrics

Database Changes:
- Added 7 new tables
- Added 6 new enums
- Added indexes for performance

API Endpoints:
- 25+ new endpoints for sales, inventory, and reporting

UI Components:
- 6 new pages for POS, inventory, reports, and dashboard

Deployment Date: ___
Deployed By: __
Verified By: ____
Status: [ ] Success [ ] Failed [ ] Rolled Back

Notes: