

Palm Oil Shop Management System

A comprehensive web-based management system designed specifically for palm oil retail shops in Nigeria. Built with Laravel 11 and featuring a clean, responsive interface using Tailwind CSS.

Features

Core Functionality

- **Inventory Management:** Track palm oil stock in litres and jerrycans with real-time updates
- **Sales Tracking:** Record sales transactions with automatic profit calculations
- **Purchase Management:** Manage supplier purchases and stock replenishment
- **Role-Based Access Control:** Separate interfaces for administrators and sales staff
- **Comprehensive Reporting:** Generate detailed sales, profit, and inventory reports
- **PDF Export:** Export reports to PDF for record-keeping and analysis

User Roles

- **Administrator:** Full system access including inventory management, reporting, and user oversight
- **Salesperson:** Limited access to sales recording and inventory viewing

Key Features

- Real-time stock level monitoring with low stock alerts
- Automatic profit calculation per transaction
- Monthly and yearly performance tracking
- Mobile-friendly responsive design

- Secure authentication system
- Multi-currency support (Nigerian Naira)

Technology Stack

- **Backend:** Laravel 11 (PHP 8.2+)
- **Frontend:** Plain HTML with Tailwind CSS
- **Database:** MySQL 8.0+
- **Authentication:** Laravel Breeze
- **PDF Generation:** TCPDF
- **Styling:** Tailwind CSS 3.x

Requirements

- PHP 8.2 or higher
- MySQL 8.0 or higher
- Composer 2.x
- Node.js 18+ (for asset compilation)
- Web server (Apache/Nginx)

Quick Installation

1. Clone the Repository

Bash

```
git clone <repository-url>  
cd palm-oil-shop
```

2. Install Dependencies

Bash

```
# Install PHP dependencies
composer install

# Install and compile frontend assets
npm install
npm run build
```

3. Environment Setup

Bash

```
# Copy environment file
cp .env.example .env

# Generate application key
php artisan key:generate
```

4. Database Configuration

Edit `.env` file with your database credentials:

Plain Text

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=palm_oil_shop
DB_USERNAME=your_username
DB_PASSWORD=your_password
```

5. Database Setup

Bash

```
# Create database tables
php artisan migrate
```

```
# Seed with sample data  
php artisan db:seed
```

6. Start the Application

Bash

```
# Development server  
php artisan serve  
  
# Production deployment  
# Configure your web server to point to the public directory
```

Default User Accounts

The system comes with pre-configured user accounts for testing:

Administrator Account

- **Email:** admin@palmoil.com
- **Password:** password
- **Role:** Administrator
- **Access:** Full system access

Salesperson Account

- **Email:** sales@palmoil.com
- **Password:** password
- **Role:** Salesperson
- **Access:** Sales recording and inventory viewing



System Architecture

Database Schema

The system uses four main database tables:

1. **Users:** Store user accounts with role-based permissions
2. **Products:** Manage palm oil products (litres/jerrycans)
3. **Purchases:** Track stock purchases from suppliers
4. **Sales:** Record sales transactions with profit calculations

Key Models and Relationships

- User → hasMany → Sales, Purchases
- Product → hasMany → Sales, Purchases
- Sale → belongsTo → User, Product
- Purchase → belongsTo → User, Product



Features Overview

Dashboard

- **Admin Dashboard:** Complete business overview with sales metrics, profit analysis, and stock alerts
- **Sales Dashboard:** Personalized view for sales staff with performance tracking and quick actions

Inventory Management

- Add and manage palm oil products
- Track stock levels in real-time
- Set up low stock alerts

- View stock movement history
- Manual stock adjustments

Sales Management

- Quick sales recording interface
- Customer information tracking
- Automatic profit calculations
- Sales history and analytics
- Performance tracking by salesperson

Purchase Management

- Record supplier purchases
- Automatic stock level updates
- Supplier information management
- Purchase history tracking
- Cost analysis

Reporting System

- **Sales Reports:** Detailed sales analysis with filtering options
- **Profit Reports:** Monthly and yearly profit tracking with trends
- **Inventory Reports:** Stock valuation and restock recommendations
- **PDF Export:** Professional report generation for record-keeping

Configuration

Application Settings

Key configuration files:

- `config/app.php` : Application settings
- `config/database.php` : Database configuration
- `config/auth.php` : Authentication settings

Customization

The system can be customized by:

- Modifying Tailwind CSS classes in views
- Updating business logic in controllers
- Adding new report types
- Extending user roles and permissions



Security Features

- CSRF protection on all forms
- SQL injection prevention through Eloquent ORM
- XSS protection with Blade templating
- Role-based access control
- Secure password hashing
- Session management



Mobile Responsiveness

The system is fully responsive and works seamlessly on:

- Desktop computers
- Tablets
- Mobile phones
- Various screen sizes and orientations

Testing

Manual Testing Checklist

- ☐ User authentication (login/logout)
- ☐ Role-based access control
- ☐ Inventory management operations
- ☐ Sales recording and calculations
- ☐ Purchase recording and stock updates
- ☐ Report generation and PDF export
- ☐ Mobile responsiveness
- ☐ Data validation and error handling

Test Data

The system includes sample data for testing:

- Sample products (palm oil in different units)
- Test sales transactions
- Sample purchase records
- User accounts for different roles

Deployment

Production Deployment

1. Set up web server (Apache/Nginx)
2. Configure PHP and required extensions
3. Set up MySQL database
4. Upload application files
5. Configure environment variables
6. Run migrations and optimizations
7. Set up SSL certificate
8. Configure backup procedures

Performance Optimization

- Enable OPcache for PHP
- Configure database indexing
- Set up caching (Redis/Memcached)
- Optimize images and assets
- Enable gzip compression

Maintenance

Regular Tasks

- Database backups
- Log file rotation
- Security updates

- Performance monitoring
- Data cleanup procedures

Monitoring

- Application logs: `storage/logs/laravel.log`
- Web server logs
- Database performance
- User activity tracking

Troubleshooting

Common Issues

Database Connection Error

- Check database credentials in `.env`
- Verify MySQL service is running
- Ensure database exists

Permission Errors

- Set proper file permissions: `chmod -R 755 storage bootstrap/cache`
- Ensure web server can write to storage directories

Asset Loading Issues

- Run `npm run build` to compile assets
- Check public directory permissions
- Verify asset paths in configuration

Authentication Problems

- Clear application cache: `php artisan cache:clear`
- Regenerate application key: `php artisan key:generate`
- Check session configuration

Support

For technical support or questions:

- Review the troubleshooting section
- Check Laravel documentation
- Examine application logs
- Contact system administrator

License

This project is proprietary software developed for palm oil retail shop management. All rights reserved.

Acknowledgments

- Laravel Framework for robust backend functionality
- Tailwind CSS for responsive design system
- TCPDF for PDF generation capabilities
- The open-source community for various packages and tools

Version: 1.0.0

Last Updated: July 2025

Developed by: Manus AI

Target Market: Palm Oil Retail Shops in Nigeria