Electrical Parts Agency - Stock Management System

# Complete Django Application Documentation

# Table of Contents

1. 1. System Overview
2. 2. Technical Architecture
3. 3. Features and Functionalities
4. 4. Database Schema
5. 5. API Endpoints
6. 6. User Guide
7. 7. Installation and Setup
8. 8. Testing and Validation
9. 9. Deployment Guidelines
10. 10. Troubleshooting

# 1. System Overview

The Electrical Parts Agency Stock Management System is a comprehensive Django-based web application designed to manage inventory, sales, and stock movements for an electrical parts distribution business. The system provides a complete solution for tracking products, managing stock levels, processing sales, and generating reports.

Key Features:

* • Product catalog management with categories and SKU tracking
* • Real-time stock level monitoring with low-stock alerts
* • Comprehensive sales tracking and customer management
* • Stock movement tracking (in/out/adjustments)
* • Responsive web interface with Bootstrap 5
* • Admin panel for system administration
* • Data visualization with charts and graphs
* • Automated fake data generation for testing

# 2. Technical Architecture

The application is built using the following technology stack:

|  |  |
| --- | --- |
| Component | Technology |
| Backend Framework | Django 4.2.7 |
| Database | SQLite (development) |
| Frontend | Bootstrap 5, HTML5, CSS3, JavaScript |
| Charts | Chart.js |
| Authentication | Django built-in authentication |
| Image Handling | Pillow |
| Fake Data | Faker library |
| Documentation | python-docx |

## 2.1 Application Structure

The Django project consists of the following apps:

* • products\_app: Product catalog management, categories, and product details
* • stock\_app: Stock level tracking, movements, and alerts
* • sales\_app: Sales processing, customer management, and analytics
* • dashboard\_app: Main dashboard with overview statistics
* • accounts\_app: User authentication and profile management

# 3. Features and Functionalities

## 3.1 Product Management

The product management module provides comprehensive functionality for managing electrical parts inventory:

* • Create, read, update, and delete products
* • Product categorization with custom categories
* • SKU (Stock Keeping Unit) management
* • Price and cost price tracking with profit margin calculation
* • Product image upload and management
* • Product status management (active/inactive)
* • Search and filter functionality
* • Bulk operations support

## 3.2 Stock Management

The stock management module tracks inventory levels and movements:

* • Real-time stock level monitoring
* • Stock movement tracking (in, out, adjustments, transfers)
* • Low stock and out-of-stock alerts
* • Minimum and maximum stock level configuration
* • Stock movement history and audit trail
* • Automated stock updates based on sales
* • Stock level reports and analytics

## 3.3 Sales Management

The sales management module handles customer transactions and sales analytics:

* • Customer management and contact information
* • Sales order creation and processing
* • Multi-item sales support
* • Sales status tracking (pending, completed, cancelled)
* • Sales analytics and reporting
* • Top-selling products tracking
* • Revenue and sales performance metrics
* • Customer sales history

## 3.4 Dashboard and Analytics

The dashboard provides a comprehensive overview of business operations:

* • Key performance indicators (KPIs)
* • Stock status overview
* • Sales performance metrics
* • Low stock alerts
* • Recent activity feed
* • Quick action buttons
* • Revenue tracking
* • Top-selling products display

# 4. Database Schema

The application uses the following main database models:

## 4.1 Products App Models

* • Category: Stores product categories with name and description
* • Product: Main product model with SKU, pricing, and category relationship

## 4.2 Stock App Models

* • StockLevel: Tracks current stock levels, minimum/maximum thresholds
* • StockMovement: Records all stock movements with type, quantity, and reference

## 4.3 Sales App Models

* • Customer: Customer information and contact details
* • Sale: Sales transactions with status and total amount
* • SaleItem: Individual items within a sale with quantity and pricing

# 5. API Endpoints

The application provides the following main URL patterns:

## 5.1 Products URLs

|  |  |  |
| --- | --- | --- |
| URL | Method | Description |
| /products/ | GET | Product list view with search and filtering |
| /products/create/ | GET, POST | Create new product |
| /products/<id>/ | GET | Product detail view |
| /products/<id>/update/ | GET, POST | Update product |
| /products/<id>/delete/ | GET, POST | Delete product |
| /products/categories/ | GET | Category list view |
| /products/categories/create/ | GET, POST | Create new category |

## 5.2 Stock URLs

|  |  |  |
| --- | --- | --- |
| URL | Method | Description |
| /stock/ | GET | Stock dashboard with overview |
| /stock/list/ | GET | Stock level list with filtering |
| /stock/movements/ | GET | Stock movements history |
| /stock/movements/add/ | GET, POST | Add new stock movement |
| /stock/level/<id>/update/ | GET, POST | Update stock level settings |
| /stock/alerts/ | GET | Low stock alerts view |
| /stock/api/ | GET | Stock data API endpoint |

## 5.3 Sales URLs

|  |  |  |
| --- | --- | --- |
| URL | Method | Description |
| /sales/ | GET | Sales dashboard with analytics |
| /sales/sales/ | GET | Sales list with filtering |
| /sales/sales/create/ | GET, POST | Create new sale |
| /sales/sales/<id>/ | GET | Sale detail view |
| /sales/sales/<id>/add-item/ | GET, POST | Add item to sale |
| /sales/customers/ | GET | Customer list |
| /sales/customers/create/ | GET, POST | Create new customer |
| /sales/analytics/ | GET | Sales analytics and charts |
| /sales/api/ | GET | Sales data API endpoint |

# 6. User Guide

## 6.1 Admin User Guide

As an admin user, you have access to all system features:

* • Access the admin panel at /admin/ to manage all data
* • Use the main dashboard at /dashboard/ for business overview
* • Navigate through products, stock, and sales modules using the sidebar
* • Create and manage product categories before adding products
* • Set up stock levels and minimum/maximum thresholds
* • Process sales and track customer information
* • Monitor low stock alerts and take appropriate actions
* • Generate reports and analyze business performance

## 6.2 Staff User Guide

Staff users can perform day-to-day operations:

* • View and search products in the catalog
* • Process stock movements (in/out/adjustments)
* • Create and manage sales transactions
* • Add new customers and update their information
* • Monitor stock levels and respond to alerts
* • View sales reports and analytics
* • Update product information as needed

# 7. Installation and Setup

Follow these steps to set up the application:

1. 1. Ensure Python 3.8+ is installed on your system
2. 2. Install required packages: pip install django pillow faker python-docx django-plotly-dash black
3. 3. Run database migrations: python manage.py migrate
4. 4. Create a superuser: python manage.py createsuperuser
5. 5. Generate fake data: python manage.py generate\_fake\_data
6. 6. Collect static files: python manage.py collectstatic
7. 7. Start the development server: python manage.py runserver
8. 8. Access the application at http://localhost:8000/

## 7.1 Automated Setup

For convenience, use the provided setup script:

chmod +x setup.sh

./setup.sh

This script will automatically install dependencies, run migrations, create a superuser, generate fake data, and run tests.

# 8. Testing and Validation

The application includes comprehensive testing:

* • Automated test suite (test\_app.py) validates all core functionality
* • Database model testing ensures data integrity
* • View testing verifies all pages are accessible
* • Admin interface testing confirms administrative access
* • Data relationship testing validates model connections
* • Fake data generation provides realistic test scenarios

Run tests using: python3 test\_app.py

# 9. Deployment Guidelines

For production deployment:

1. 1. Set DEBUG = False in settings.py
2. 2. Configure a production database (PostgreSQL recommended)
3. 3. Set up proper static file serving
4. 4. Configure media file storage
5. 5. Set up SSL/HTTPS
6. 6. Configure proper ALLOWED\_HOSTS
7. 7. Set up logging and monitoring
8. 8. Use a production WSGI server (Gunicorn recommended)
9. 9. Set up reverse proxy (Nginx recommended)

# 10. Troubleshooting

Common issues and solutions:

* • Database errors: Run python manage.py migrate to apply migrations
* • Static files not loading: Run python manage.py collectstatic
* • Permission errors: Check file permissions and ownership
* • Import errors: Ensure all dependencies are installed
* • Template not found: Check TEMPLATES setting in settings.py
* • Media files not serving: Configure MEDIA\_URL and MEDIA\_ROOT
* • CSRF errors: Ensure CSRF middleware is enabled
* • Login issues: Verify user exists and password is correct

# Conclusion

The Electrical Parts Agency Stock Management System provides a comprehensive solution for managing electrical parts inventory, sales, and stock movements. The system is built with Django best practices, includes responsive design, and provides extensive functionality for both administrative and operational users.

The application is ready for immediate use and can be easily extended with additional features as needed. All code is well-documented and follows Django conventions for maintainability and scalability.