Backend

- Complete doc

Complete doc

Overview

OnMart POS Backend is a full-featured RESTful API for managing retail operations. It provides comprehensive features for product management, transaction processing, inventory tracking, cashier shift management, and customer relationship management.

Key Features

- **JWT Authentication & Authorization** - Secure role-based access control
- **Multi-Role User Management** - MasterAdmin, SuperAdmin, Admin, Cashier
- **Multi-Outlet Support** - Manage multiple store locations
- **Product Management** - Categories, brands, barcodes, and inventory
- **Transaction Processing** - Complete POS functionality with void/refund
- **Shift Management** - Track cashier shifts and sales
- **Promotional System** - Flexible discount management
- **Customer Management** - Customer profiles and transaction history
- **Dashboard & Reporting** - Sales analytics and insights
- **Invoice Generation** - PDF receipt printing
- **Import/Export** - CSV and Excel support

Quick Start

Prerequisites

- Node.js v16 or higher
- PostgreSQL v12 or higher
- npm or yarn

Installation

1. **Clone the repository**

```
git clone <repository-url>
cd onmart-pos-backend
```

2. **Install dependencies**

```
npm install
```

3. **Configure environment variables**

```
cp .env.example .env
# Edit .env with your configuration
```

4. **Set up database**

```
# Create database
createdb onmart_pos_dev

# Run migrations
npx sequelize-cli db:migrate

# Seed initial data
npx sequelize-cli db:seed:all
```

5. **Start the server**

```
npm start
```

The server will be running at `http://localhost:3000` □□

Default Login Credentials

After seeding, you can log in with these accounts:

Role	Email	Password
MasterAdmin	masteradmin@mail.com	password
SuperAdmin	superadmin@mail.com	password
Admin	admin@mail.com	password
Cashier	cashier@mail.com	password

⚠ **Important:** Change these passwords in production!

□□ Documentation

Comprehensive documentation is available in the `/docs` folder:

- **Backend Documentation** - Complete API reference, authentication, and development guide
- **Database Documentation** - Database schema, relationships, and query examples

Quick Links

- Installation Guide
- API Endpoints
- Database Schema
- Environment Variables

□□ Project Structure

```
onmart-pos-backend/
├── config/             # Database configuration
├── controllers/        # Business logic
├── docs/               # Documentation
│   ├── BACKEND.md      # Backend & API documentation
│   └── DATABASE.md     # Database documentation
├── helpers/            # Utility functions
├── middlewares/        # Express middlewares
├── migrations/         # Database migrations
├── models/             # Sequelize models
├── routes/             # API routes
├── seeders/            # Database seeders
├── uploads/            # File uploads
├── utils/              # Utilities
├── index.js            # Entry point
└── package.json        # Dependencies
```

□□ API Endpoints

Base URL: `http://localhost:3000/api`

Core Endpoints

Category	Endpoint	Description
Auth	`POST /auth/login`	User login
Auth	`GET /auth/me`	Get current user
Products	`GET /products`	List products
Products	`POST /products`	Create product
Transactions	`POST /transactions`	Create transaction
Cashier	`POST /cashier/shifts/open`	Open shift
Dashboard	`GET /dashboard/summary`	Dashboard stats

For complete API documentation, see BACKEND.md.

□□ Tech Stack

- **Runtime:** Node.js
- **Framework:** Express v5.1.0
- **Database:** PostgreSQL
- **ORM:** Sequelize v6.37.7
- **Authentication:** JWT (jsonwebtoken)
- **Security:** Helmet, CORS, bcrypt
- **File Processing:** Multer, Sharp
- **PDF Generation:** PDFKit
- **Data Processing:** fast-csv, xlsx
- **Logging:** Morgan

□□ Security Features

- □ JWT-based authentication
- □ Password hashing with bcrypt

- □ Role-based authorization
- □ Security headers (Helmet)
- □ CORS protection
- □ Input validation
- □ SQL injection prevention (Sequelize ORM)

□□ Development

Database Commands

```
# Run migrations
npx sequelize-cli db:migrate

# Undo last migration
npx sequelize-cli db:migrate:undo

# Run seeders
npx sequelize-cli db:seed:all

# Database reset
npx sequelize-cli db:migrate:undo:all
npx sequelize-cli db:migrate
npx sequelize-cli db:seed:all
```

Environment Variables

Key environment variables (see `.env.example` for all options):

```
PORT=3000
NODE_ENV=development
DB_USERNAME=postgres
DB_PASSWORD=your_password
DB_NAME=onmart_pos_dev
DB_HOST=localhost
JWT_SECRET=your_jwt_secret
ORIGIN_CORS=http://localhost:3711
```

□□ Database Schema

The system uses 16 tables with proper relationships:

- **Users & Roles** - Authentication and authorization
- **Products** - Inventory with categories and brands
- **Transactions** - Sales records with items
- **Shifts** - Cashier working periods
- **Customers** - Customer information
- **Promos** - Promotional campaigns
- **Outlets** - Store locations

For detailed schema information, see DATABASE.md.

□□ Contributing

1. Fork the repository
2. Create a feature branch (`git checkout -b feature/amazing-feature`)
3. Commit your changes (`git commit -m 'Add amazing feature'`)
4. Push to the branch (`git push origin feature/amazing-feature`)
5. Open a Pull Request

□□ License

This project is licensed under the ISC License.

□□ Support

For questions or support:

- Open an issue in the repository
- Contact the development team

□□ For New Developers

If you're taking over this project, start with these documents:

1. **Backend Documentation** - Understand the API architecture and endpoints
2. **Database Documentation** - Learn the database structure and relationships
3. **Environment Setup** - Follow the Quick Start guide above
4. **Test the API** - Use the default credentials to explore the system

Recommended Reading Order

1. Installation & Setup
2. Database Schema Overview
3. Authentication Flow
4. API Endpoints Reference
5. Code Structure & Organization

Version: 1.0.0
Last Updated: February 9, 2026