MHO B2B E-commerce Platform

A modern, scalable B2B e-commerce platform with real-time Zoho Inventory integration, built with Next.js 16, Prisma, and Docker.

!Next.js
!TypeScript
!Prisma
!Docker

Features

Modern UI/UX

- **Beautiful Product Catalog** - Grid/List views with smooth animations
- **Shadcn/Radix UI** - Modern component library with accessibility
- **Responsive Design** - Mobile-first approach
- **Stock Badges** - Color-coded inventory status (In Stock/Low Stock/Out of Stock)
- **Product Modals** - Quick view with detailed information

Zoho Integration

- **Real-time Sync** - Automatic product synchronization from Zoho Inventory
- **Zoho Books Integration** - Automatic sales orders and invoice generation
- **Branch Auto-registration** - Branches automatically synced as customers/contacts
- **Invoice PDF Generation** - Automatic PDF creation and viewing
- **OAuth 2.0** - Secure authentication with token refresh
- **Comprehensive Data** - 50+ product fields synced (pricing, stock, images, etc.)
- **Multi-organization** - Support for multiple Zoho organizations
- **Service Account** - Background sync service for scheduled updates

B2B Features

- **Shopping Cart** - Branch-specific carts with guest support
- **Order Management** - Complete order lifecycle from cart to invoice
- **Invoice System** - Automatic invoice generation with PDF viewing
- **Quotations** - Create, manage, and convert quotations to orders

- **Multi-branch Support** - Manage multiple business locations
- **Organization Management** - Company and branch hierarchy
- **Role-based Access** - Admin, Manager, Customer roles
- **Branch Switching** - Easy switching between locations
- **Recent Activity Dashboard** - Real-time activity feed for orders, invoices, and branches

Analytics & Reporting

- **Comprehensive Analytics Dashboard** - Real-time insights into spending and orders
- **Financial Metrics** - Total spent amount, average order value, cost per employee
- **Interactive Charts** - Revenue over time, category breakdown, top products
- **Time Range Filters** - Last 7 days, 30 days, 90 days, 1 year, or custom range
- **Branch Filtering** - Filter analytics by specific branch
- **Export Capabilities** - Export analytics to PDF or CSV
- **Latest Orders & Quotations** - Quick view of recent activity
- **Status Breakdown** - Visual breakdown of orders by status
- **AED Currency** - All financial data displayed in UAE Dirhams

Authentication & Security

- **NextAuth.js** - Secure authentication with session management
- **Redis Sessions** - Fast, scalable session storage
- **Protected Routes** - Role-based access control
- **Secure API** - JWT-based authentication

Performance

- **Docker Compose** - Complete containerized environment
- **PostgreSQL** - Robust relational database
- **Redis Caching** - Fast session and data caching
- **Image Optimization** - Sharp for image compression
- **Optimistic Loading** - Skeleton states for better UX

Architecture

```
MHO/
├── apps/web/                 # Next.js frontend & API
│   ├── app/                  # Next.js 16 App Router
│   │   ├── api/              # API routes
│   │   │   ├── auth/         # Authentication endpoints
│   │   │   ├── products/     # Product management
│   │   │   ├── cart/         # Shopping cart endpoints
│   │   │   ├── orders/       # Order management
│   │   │   ├── invoices/     # Invoice endpoints
│   │   │   ├── analytics/    # Analytics & reporting
│   │   │   ├── quotations/   # Quotation management
│   │   │   ├── zoho/         # Zoho integration
│   │   │   └── orgs/         # Organization management
│   │   ├── products/         # Product catalog page
│   │   ├── cart/             # Shopping cart page
│   │   ├── checkout/         # Checkout page
│   │   ├── orders/           # Orders list & details
│   │   ├── quotations/       # Quotations list & management
│   │   ├── analytics/        # Analytics dashboard
│   │   ├── dashboard/        # Admin dashboard
│   │   └── onboarding/       # User onboarding flow
│   ├── components/           # React components
│   │   ├── ui/                    # Shadcn UI components
│   │   └── analytics/             # Analytics chart components
│   ├── lib/                  # Utilities & helpers
│   │   ├── auth.ts           # NextAuth configuration
│   │   ├── prisma.ts         # Database client
│   │   ├── redis.ts          # Redis client
│   │   └── zoho.ts           # Zoho API client
│   └── prisma/               # Database schema & migrations
├── services/zoho-sync/       # Standalone Zoho sync service
│   ├── src/
│   │   ├── index.ts          # Service entry point
│   │   ├── sync.ts           # Sync logic
│   │   └── zoho-client.ts    # Zoho API wrapper
│   └── Dockerfile
└── docker-compose.yml        # Container orchestration
```

Quick Start

Prerequisites

- Docker & Docker Compose
- Node.js 20+ (for local development)
- pnpm
- Zoho Inventory account

1. Clone the Repository

```
git clone https://github.com/Lanioque/MHO-B2B-Ecommerce.git
```

```
cd MHO-B2B-Ecommerce
```

2. Environment Setup

```
cp apps/web/env.example apps/web/.env
```

Edit `apps/web/.env` with your credentials:

```
# Database
DATABASE_URL=postgresql://postgres:postgres@localhost:5432/ecommerce_dev

# Redis
REDIS_URL=redis://localhost:6379

# NextAuth
NEXTAUTH_URL=http://localhost:3000
NEXTAUTH_SECRET=your-secret-key-here

# Zoho (Get from Zoho API Console)
ZOHO_CLIENT_ID=your_client_id
ZOHO_CLIENT_SECRET=your_client_secret
ZOHO_ORGANIZATION_ID=your_org_id
ZOHO_REGION=eu  # or us, in, au
ZOHO_SCOPE=ZohoInventory.fullaccess.all,ZohoBooks.contacts.CREATE,ZohoBooks.contacts.READ,ZohoB
  ooks.salesorders.CREATE,ZohoBooks.invoices.CREATE,ZohoBooks.invoices.READ
ZOHO_BOOKS_ORGANIZATION_ID=your_books_org_id
```

3. Start with Docker

```
docker-compose up -d
```

The application will be available at:

- **Web App**: http://localhost:3000
- **Prisma Studio**: http://localhost:5555
- **PostgreSQL**: localhost:5432
- **Redis**: localhost:6379

4. Initial Setup

Visit http://localhost:3000/register to create an account
Complete the onboarding process
Go to http://localhost:3000/test-zoho to connect Zoho
Click "Sync to Database" to import products

Development

Local Development (without Docker)

```
# Install dependencies
pnpm install

# Start PostgreSQL & Redis
docker-compose up postgres redis -d

# Run migrations
cd apps/web
pnpm prisma migrate dev

# Start dev server
pnpm dev
```

Database Management

```
# Generate Prisma client
pnpm prisma generate

# Create migration
pnpm prisma migrate dev --name your_migration_name

# Open Prisma Studio
pnpm prisma studio

# Seed database
pnpm db:seed
```

Building for Production

```
# Build all services
docker-compose build

# Start in production mode
docker-compose up -d
```

Tech Stack

Frontend

- **Next.js 16** - React framework with App Router
- **React 19** - Latest React with React Compiler
- **TypeScript** - Type safety

- **Tailwind CSS 4** - Utility-first CSS
- **Shadcn UI** - Component library
- **Radix UI** - Headless UI primitives
- **Lucide Icons** - Beautiful icon set

Backend

- **Next.js API Routes** - RESTful API
- **NextAuth.js** - Authentication
- **Prisma** - Type-safe ORM
- **PostgreSQL** - Primary database
- **Redis** - Session & cache store
- **Zod** - Runtime validation

Analytics & Charts

- **Recharts** - Interactive charts (line, bar, pie)
- **date-fns** - Date manipulation and formatting
- **jsPDF & jsPDF-AutoTable** - PDF generation for exports
- **Papaparse** - CSV export functionality

DevOps

- **Docker** - Containerization
- **Docker Compose** - Multi-container orchestration
- **pnpm** - Fast package manager
- **ESLint & Prettier** - Code quality

API Endpoints

Authentication

- `POST /api/auth/register` - User registration
- `POST /api/auth/signout` - Sign out

Products

- `GET /api/products` - List products (paginated)
- `GET /api/products/:id` - Get product details
- `POST /api/products` - Create product (admin)

Shopping Cart

- `GET /api/cart` - Get user's cart (with branch filter)
- `POST /api/cart` - Add items to cart
- `DELETE /api/cart` - Clear cart
- `PATCH /api/cart/items/:itemId` - Update cart item quantity
- `DELETE /api/cart/items/:itemId` - Remove item from cart

Orders

- `GET /api/orders` - List orders (with filters)
- `POST /api/orders` - Create order from cart
- `GET /api/orders/:id` - Get order details
- `PATCH /api/orders/:id` - Update order status

Invoices

- `GET /api/invoices/:id` - Get invoice details
- `GET /api/invoices/:id/pdf` - Download invoice PDF

Analytics

- `GET /api/analytics` - Get organization analytics (with date range and branch filters)
- `GET /api/analytics/export` - Export analytics to CSV or PDF

Quotations

- `GET /api/quotations` - List quotations (with filters)
- `POST /api/quotations` - Create quotation
- `GET /api/quotations/:id` - Get quotation details
- `PATCH /api/quotations/:id` - Update quotation status or convert to order
- `DELETE /api/quotations/:id` - Delete quotation

Zoho Integration

- `GET /api/zoho/oauth/start` - Start OAuth flow
- `GET /api/zoho/oauth/callback` - OAuth callback
- `GET /api/zoho/products` - Fetch from Zoho (live)
- `POST /api/zoho/sync` - Sync to database

Organizations

- `GET /api/orgs` - List organizations
- `POST /api/orgs` - Create organization
- `GET /api/branches` - List branches

Environment Variables

See `apps/web/env.example` for all required variables.

Database Schema

Key models:

- **User** - Authentication & profile
- **Organization** - Company entities
- **Branch** - Business locations (with Zoho contact sync)
- **Membership** - User-org relationships with roles
- **Product** - Product catalog (global)
- **Cart** - Shopping carts (branch-specific, guest support)
- **CartItem** - Cart line items
- **Order** - Customer orders (with status tracking)
- **OrderItem** - Order line items
- **Invoice** - Generated invoices (linked to orders)
- **Quotation** - Quotations with status lifecycle (DRAFT, SENT, APPROVED, REJECTED, EXPIRED, CONVERTED)
- **QuotationItem** - Quotation line items
- **Customer** - Customer records
- **ZohoConnection** - OAuth tokens & settings

All Features Done

Core Features

- [x] **User Authentication & Authorization**
- NextAuth.js with Redis sessions
- Role-based access control (OWNER, ADMIN, STAFF, CUSTOMER)
- Secure password hashing
- Session management

- [x] **Organization & Branch Management**
- Multi-organization support
- Branch creation and management
- Employee management per branch
- Address management (billing & shipping)
- Branch-specific operations

- [x] **Product Catalog**
- Product listing with pagination
- Detailed product information (50+ fields)
- Product categories
- Stock status tracking (In Stock/Low Stock/Out of Stock)
- Product images support
- Dietary information and tags
- Product search and filtering

- [x] **Shopping Cart System**
- Branch-specific carts
- Guest cart support (session-based)
- Authenticated user carts
- Quantity management
- Real-time cart updates
- Cart drawer UI component

- [x] **Order Management**
- Order creation from cart
- Order status tracking (PENDING, AWAITING_PAYMENT, PAID, FAILED, CANCELLED, REFUNDED)
- Order history
- Order details view
- Order items tracking

- [x] **Invoice System**
- Automatic invoice generation

- Invoice PDF viewing
- Invoice status tracking (DRAFT, ISSUED, SENT, PAID, OVERDUE, VOID)
- Linked to orders

- [x] **Quotation Management**
- Quotation creation
- Quotation status lifecycle (DRAFT, SENT, APPROVED, REJECTED, EXPIRED, CONVERTED)
- Convert quotations to orders
- Quotation history
- Expiry date management

- [x] **Analytics & Reporting**
- Comprehensive analytics dashboard
- Revenue over time charts
- Category breakdown (bar charts)
- Top products visualization
- Status pie charts (orders, quotations)
- Time range filters (7 days, 30 days, 90 days, 1 year, custom)
- Branch filtering
- Export to PDF/CSV
- Latest orders and quotations tables
- Financial metrics (total spent, average order value, cost per employee)
- AED currency display

Zoho Integration

- [x] **OAuth 2.0 Integration**
- Secure OAuth flow
- Token refresh mechanism
- Multi-organization support
- Region support (EU, US, IN, AU)

- [x] **Zoho Inventory Sync**
- Real-time product synchronization
- 50+ product fields synced
- Stock level updates
- Product image sync
- Scheduled sync service
- Manual sync trigger

- [x] **Zoho Books Integration**

- Automatic sales order creation
- Automatic invoice generation
- Branch auto-registration as contacts/customers
- Invoice PDF generation
- Customer management sync

User Interface

- [x] **Modern UI Components**
- Shadcn/Radix UI components
- Responsive design (mobile-first)
- Dark mode support (implicit via theme)
- Beautiful animations
- Skeleton loading states
- Toast notifications

- [x] **Pages & Routes**
- Product catalog page
- Shopping cart page
- Checkout page
- Order listing and details
- Quotation management
- Analytics dashboard
- Admin dashboard
- Organization management
- Branch management
- User onboarding flow
- Login/Register pages

Technical Features

- [x] **Database & ORM**
- PostgreSQL database
- Prisma ORM with type safety
- Database migrations
- Seed scripts

- [x] **Caching & Performance**
- Redis session storage
- Redis caching

- Image optimization with Sharp
- Optimistic UI updates

- [x] **API & Backend**
- RESTful API routes
- Request validation (Zod)
- Error handling middleware
- Type-safe API responses

- [x] **DevOps & Deployment**
- Docker containerization
- Docker Compose setup
- Environment configuration
- Production build setup

User Stories

As a **Customer (Branch Manager)**

As a customer, I want to browse products in a catalog so that I can see available items with prices and stock status.
As a customer, I want to add products to my cart so that I can prepare an order.
As a customer, I want to manage cart items (quantity, remove) so that I can adjust my order before checkout.
As a customer, I want to place orders from my cart so that I can purchase items for my branch.
As a customer, I want to view my order history so that I can track my past purchases.
As a customer, I want to view order details including status so that I know when my order will be delivered.
As a customer, I want to view invoices so that I can keep records for accounting.
As a customer, I want to create quotations so that I can prepare quotes before placing orders.
As a customer, I want to convert quotations to orders so that I can quickly place orders from approved quotes.
As a customer, I want to switch between branches so that I can manage orders for different locations.

As an **Admin/Staff**

As an admin, I want to manage organizations so that I can set up new client companies.
As an admin, I want to manage branches so that I can configure multiple locations per organization.
As an admin, I want to view all orders across the organization so that I can monitor business activity.
As an admin, I want to update order status so that customers know the progress of their orders.
As an admin, I want to view analytics dashboard so that I can understand business performance.
As an admin, I want to export analytics data so that I can create reports for stakeholders.

As an admin, I want to filter analytics by branch and date range so that I can analyze specific periods and locations.
As an admin, I want to manage quotations so that I can track sales opportunities.
As an admin, I want to view employee information so that I can manage workforce.
As an admin, I want to access role-based features so that I can perform administrative tasks.

As a **System Administrator**

As a system admin, I want to connect Zoho Inventory so that products are automatically synced.
As a system admin, I want products to sync automatically so that inventory stays up-to-date.
As a system admin, I want orders to be created in Zoho Books so that accounting is automated.
As a system admin, I want invoices to be generated automatically so that billing is streamlined.
As a system admin, I want branches to be registered in Zoho Books so that they appear as customers.
As a system admin, I want to manage Zoho OAuth tokens so that integration stays connected.

As a **Guest User**

As a guest, I want to browse products so that I can see what's available before registering.
As a guest, I want to add items to a cart so that I can prepare an order before creating an account.

Roadmap - 6 Week Sprint Plan

Week 1: Email Notifications & Basic Search

Goal: Improve user communication and product discovery

- [ ] **Email Notifications** (Priority: High)
- Order confirmation emails
- Invoice delivery emails
- Order status update notifications
- Quotation sent/approved emails

- [ ] **Basic Product Search** (Priority: High)
- Full-text product search
- Product filtering by category and brand
- Product sorting options
- Search results highlighting

Week 2: Payment Processing Foundation

Goal: Enable online payments

- [ ] **Payment Gateway Integration** (Priority: Critical)
- Integration with payment gateway (Stripe/PayPal/Telr)
- Payment method selection
- Payment status tracking
- Payment confirmation handling

- [ ] **Payment Management**
- Payment history view
- Payment receipt generation
- Payment failure handling

Week 3: Advanced Search & Filters

Goal: Enhanced product discovery

- [ ] **Advanced Search Features**
- Search autocomplete/suggestions
- Saved search queries
- Search filters (price range, stock status, tags)
- Search history

- [ ] **UI/UX Enhancements**
- Improved product grid/list views
- Quick filter sidebar
- Recent searches display

Week 4: Bulk Operations & Workflow Automation

Goal: Improve efficiency for admins

- [ ] **Bulk Operations**
- Bulk order creation
- Bulk product import/export (CSV)
- Bulk quotation generation
- Bulk status updates

- [ ] **Workflow Automation**
- Automated quotation expiration
- Inventory low stock alerts
- Automated order status transitions

- Automated invoice generation rules

Week 5: API Documentation & Webhooks

Goal: Enable integrations

- [ ] **API Documentation** (Priority: Medium)
- Swagger/OpenAPI documentation setup
- API endpoint documentation
- Interactive API explorer
- Code samples and examples

- [ ] **Webhook System**
- Webhook configuration UI
- Event subscriptions (order created, invoice paid, etc.)
- Webhook delivery and retry mechanism
- Webhook logs and monitoring

Week 6: Polish & Testing

Goal: Production readiness

- [ ] **Quality Assurance**
- End-to-end testing of all features
- Performance optimization
- Security audit
- Bug fixes and refinements

- [ ] **Documentation & Training**
- User manual completion
- Admin guide
- API documentation finalization
- Video tutorials (optional)

- [ ] **Deployment Preparation**
- Production environment setup
- Database migration scripts
- Backup and recovery procedures
- Monitoring and alerting setup

Future Roadmap (Post-6 Weeks)

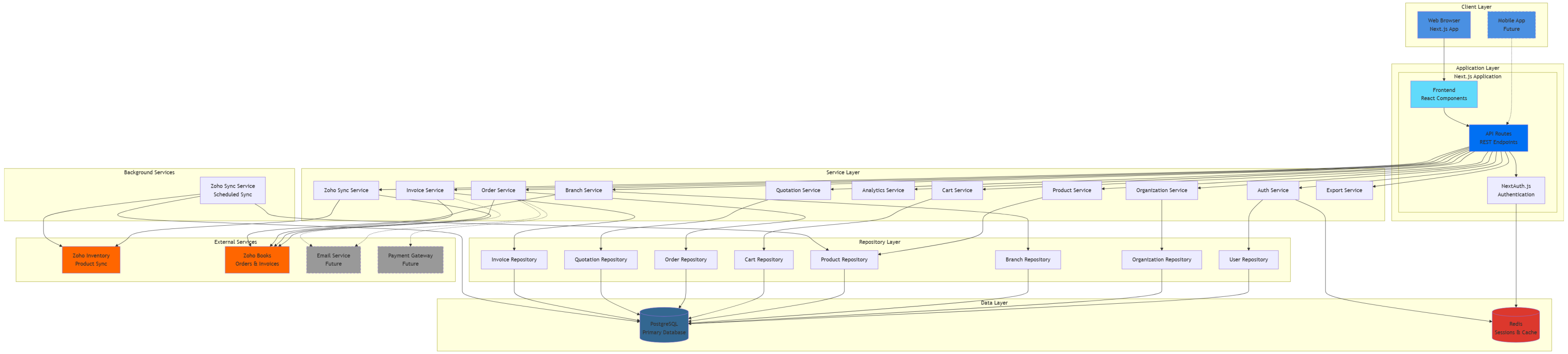
Next Phase: Enhanced Features

- [ ] Enhanced Analytics (predictive analytics, forecasting)
- [ ] Product Recommendations (AI-powered suggestions)
- [ ] Multi-currency Support
- [ ] Internationalization (i18n) - Multi-language support
- [ ] Additional Integrations (Salesforce, QuickBooks, Xero)

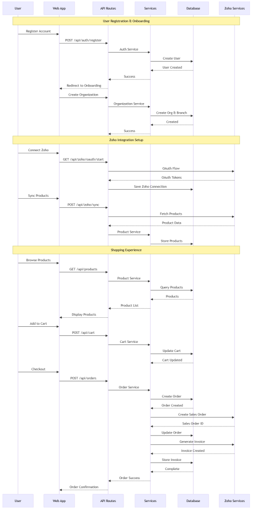
Advanced Features

- [ ] Mobile Application (iOS/Android)
- [ ] Subscription Management (recurring orders)
- [ ] Loyalty Program (points, rewards)
- [ ] Advanced Inventory (multi-location, warehouse management)
- [ ] Collaboration Features (comments, activity feed)
- [ ] Advanced AI features (chatbot, natural language queries)

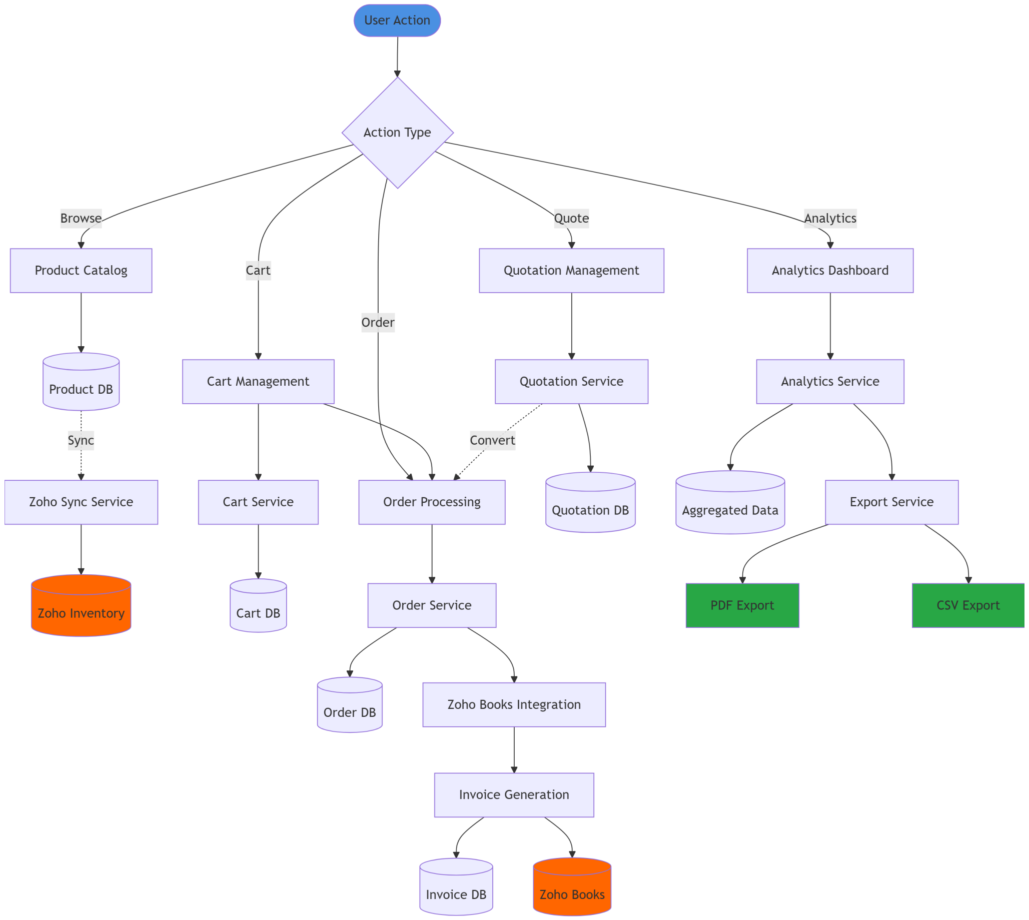
System Architecture Diagram



User Journey Flow



Data Flow Diagram



Contributing

Fork the repository
Create a feature branch (`git checkout -b feature/amazing-feature`)
Commit your changes (`git commit -m 'Add amazing feature'`)
Push to the branch (`git push origin feature/amazing-feature`)

Open a Pull Request

License

This project is licensed under the MIT License.

Acknowledgments

- Next.js
- Shadcn UI
- Prisma
- Zoho Inventory

Support

For issues and questions, please open an issue on GitHub.

Built with using Next.js & TypeScript

Export as PDF

To export this documentation as PDF with rendered Mermaid diagrams, you have several options:

Option 1: Browser Export (Easiest - Recommended)

Visit your GitHub Pages documentation: https://lanioque.github.io/MHO-B2B-Ecommerce/
Wait for all Mermaid diagrams to fully render (check browser console if needed)
Press `Ctrl+P` (Windows/Linux) or `Cmd+P` (Mac)
Select "Save as PDF" as destination
Click "More settings" → Enable **"Background graphics"**
Adjust margins if needed
Click "Save"

Pros: Works immediately, preserves all formatting and diagrams
Cons: Manual process

Option 2: Python Script (Local MD File - Recommended)

Install dependencies:

```
pip install md2pdf-mermaid playwright markdown
python -m playwright install chromium
```

Run the export script:

```
python scripts/export-pdf.py
```

This automatically:
• Reads your local `README.md` file
• Renders all Mermaid diagrams
• Generates `README.pdf` in the project root

Option 3: Automated Script (GitHub Pages - Puppeteer)

For exporting from GitHub Pages instead:

Install Puppeteer:

```
npm install puppeteer --save-dev
```

Run the export script:

```
node scripts/export-readme-pdf.js
```

See `scripts/README.md` for more options including Pandoc, VS Code extensions, and online services.