



Mini-Store-ERP — Project Documentation Guide

1. Introduction

The **Mini-Store-ERP** is a full-stack MERN-based (MongoDB, Express.js, React.js, Node.js) Enterprise Resource Planning solution designed for small and medium-sized retail stores. This documentation guide provides a comprehensive overview of the system architecture, design decisions, features, technology stack, API structure, and deployment workflow. It serves as a reference manual for developers, testers, and evaluators.

2. Project Overview

2.1 Objective

To build a modern, AI-powered inventory and sales management system that automates daily store operations, provides actionable insights, and ensures accurate stock tracking.

2.2 Key Features

- Secure user authentication with roles (Admin/Cashier)
 - Product catalog & stock management
 - Purchase & supplier tracking
 - POS & sales management with automatic stock deduction
 - PDF invoice generation
 - AI-powered insights using Groq LLM
 - Real-time dashboards & analytics
 - Error handling, validation, and system logging
-

3. System Architecture

The Mini-Store-ERP follows a **3-tier architecture**:

3.1 Frontend

- React + TypeScript
- TailwindCSS for UI
- React Router v6
- Zustand for state management
- Axios for API communication

3.2 Backend

- Node.js + Express.js
- TypeScript-based service layer
- JWT authentication
- Groq API for AI insights
- jsPDF for invoice generation
- Mongoose ORM

3.3 Database

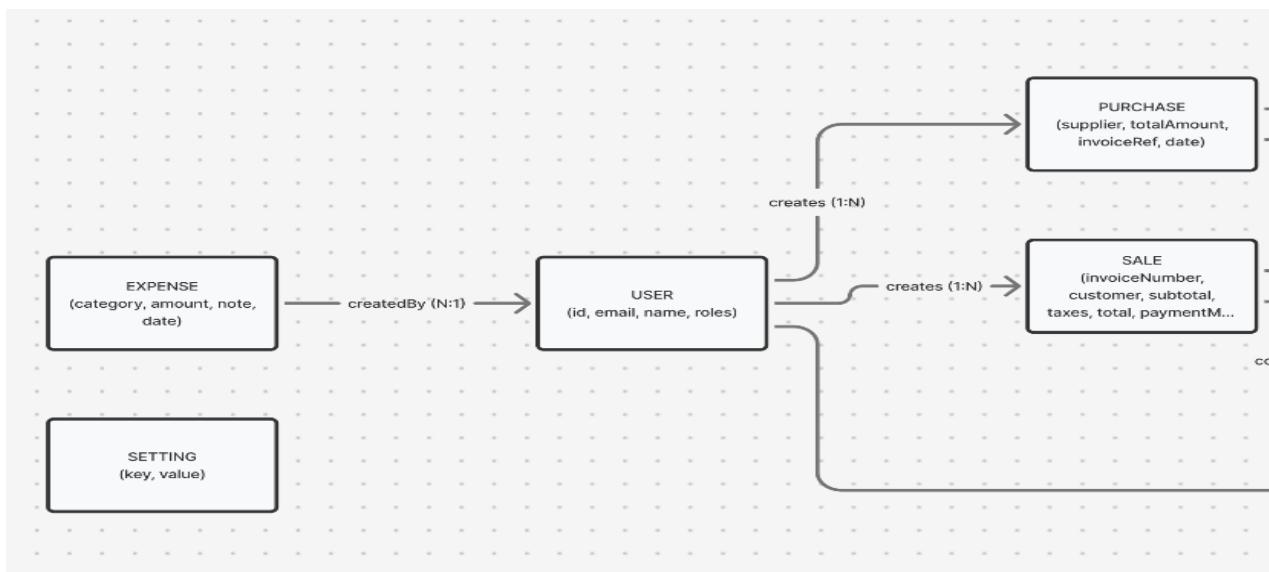
- MongoDB (NoSQL document-oriented database)
-

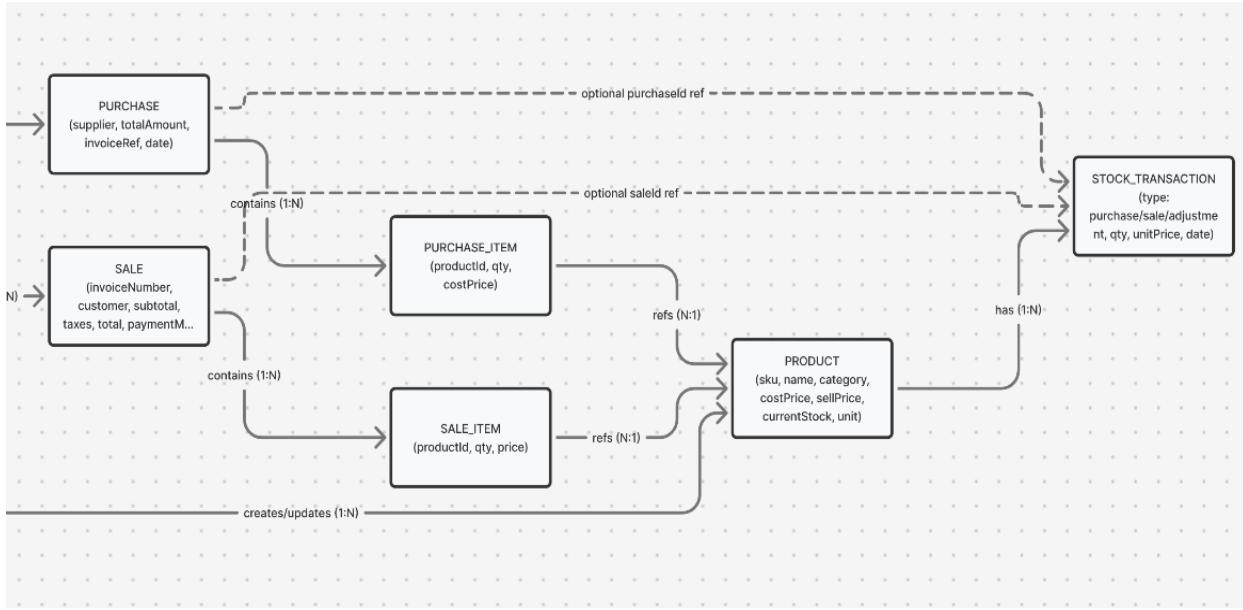
4. ER Diagram & Data Models

Below is the ER diagram summary (as provided):

- **User** ↔ Product (createdBy, updatedBy)
- **Product** ↔ StockTransaction
- **Sale** ↔ StockTransaction
- **Purchase** ↔ StockTransaction
- **Product** ↔ Sale.items[]
- **Product** ↔ Purchase.items[]
- **Expense** linked to User
- **Settings** standalone for system configuration

Full ER diagram is included in the project assets.





5. Backend API Modules

5.1 Authentication API

- Login feature
- Register feature
- Verify token

5.2 Product API

- Create product
- Update product
- Delete product
- Fetch product list

5.3 Purchase API

- Create purchase
- List purchases
- Generate stock additions

5.4 Sales API

- Create sale
- Handle POS checkout
- Auto-generate invoice PDFs

5.5 Stock Transactions API

- Track stock movements
- Adjustment logs

5.6 Expense API

- Add expenses
- Categorize expenses

5.7 AI Insights API

- Natural-language query → JSON insights
 - Smart reorder suggestions
-

6. Frontend and Backend Structure

Key Screens of the Project

- Login/Registration Screen Page
- Dashboard (Revenue, COGS, Top Products)
- Products Page
- Add/Edit Product Page
- Sales POS Page
- Purchases & Suppliers Page
- Reports & Insights Page

Frontend (React + TypeScript) — Complete Structure

```
client/
  └── components/          # Reusable UI components (buttons, modals, tables)
  └── pages/               # Main route pages (Dashboard, Products, POS, etc.)
  └── hooks/               # Custom hooks (useAuth, useFetch, useStore)
  └── context/             # Global context providers
  └── layouts/             # Layout components (Sidebar, Navbar,
ProtectedLayout)
  └── assets/              # Images, icons, static files
  └── utils/               # Helper functions (formatters, validators)
  └── store/               # Zustand store for state management
  └── lib/                 # API handlers, axios wrapper
  └── types/               # TypeScript types
  └── router/              # Route configuration
  └── styles/              # Global styles or theme config
  └── config/              # Environment configs, constants
```

Backend (Node.js + Express + TypeScript) — Complete Structure

```
server/
  └── src/
    ├── config/          # DB config, environment loader
    ├── controllers/    # API controllers for each module
    ├── middleware/     # Auth, error handling, validation
    ├── models/          # Mongoose schemas & models
    ├── services/         # Business logic layer (ProductService, SaleService)
    ├── routes/          # API route definitions
    ├── utils/           # Helpers: tokens, logger, PDF tools
    ├── constants/        # App constants, enums, role permissions
    ├── validators/       # Request validation schemas
    └── app.ts            # Express app initialization
  └── logs/              # Winston or custom logs
  └── public/            # Public assets (invoice templates)
  └── package.json       # Dependencies
```

These structures represent all important directories required for scalable frontend and backend development.

7. AI Integration (Groq LLM)

Use Cases of this Project

1. Natural Language Query Understanding

Users can type queries in plain English, such as:

- “Show me today’s revenue.”
- “Which products are running low?”
- “Top-selling categories this month?”

The backend converts these into structured prompts for the Groq model.

- ✓ **No need for SQL or filters**
- ✓ **Fast real-time responses**

2. Business Insights Generation

Groq processes store data and provides insights such as:

- Revenue trends
- Profit vs expense summary
- Top-selling products
- Low-performing items

- Day-wise, week-wise, month-wise metrics
- Supplier performance
- High-cost vs high-profit products

The model returns a **clean JSON** format to be easily displayed in React charts.

3. Smart Reorder Suggestions

This feature calculates:

- Daily sales movement
- Remaining stock
- Lead time (optional future upgrade)
- Reorder thresholds

The AI suggests following:

- Which items to reorder
- Estimated stock-out days
- Recommended reorder quantities

Useful for inventory optimization.

4. Conversational Analytics

The system behaves like a **business analyst chatbot**:

- Answers follow-up queries
- Remembers the context of the previous question (if implemented)
- Delivers short, precise insights

This improves UX over static dashboards.

5. Backend Prompt Engineering

The backend formats the prompt with:

- Sales and Purchase data
- Stock and Expenses data
- Revenue summaries and Date ranges
- Product names, categories, vendors

Before sending to Groq, it ensures:

- No leakage of raw DB schema
- Model-friendly structure

Groq returns a **stable JSON output**, not plain text.

8. PDF Generation

Invoice PDF

- Invoice number
- Customer details
- Itemized list
- Tax + discount
- Total amount

Tech Used

- jsPDF
 - AutoTable plugin
-

9. Deployment Guide

Backend Deployment (Render)

- Build: `cd server; npm install; tsc`
- Start: `cd server; npm start`
- Add environment variables

Frontend Deployment (Render)

- Build: `cd client; npm run build`
 - Publish folder: `client/dist`
 - Add `VITE_API_URL` env
-

10. Environment Variables

Server .env

```
MONGODB_URI="enter your mongodb url here"
JWT_SECRET="enter your jwt secret here"
GROQ_API_KEY="enter your groq api key here"
GROQ_MODEL="enter the model name"
CORS_ORIGIN="enter backend deployment url(production)"
NODE_ENV="development"
JWT_EXPIRE=24h
JWT_REFRESH_EXPIRE=7d
```

```
RATE_LIMIT_WINDOW_MS=900000  
RATE_LIMIT_MAX_REQUESTS=100
```

Client .env

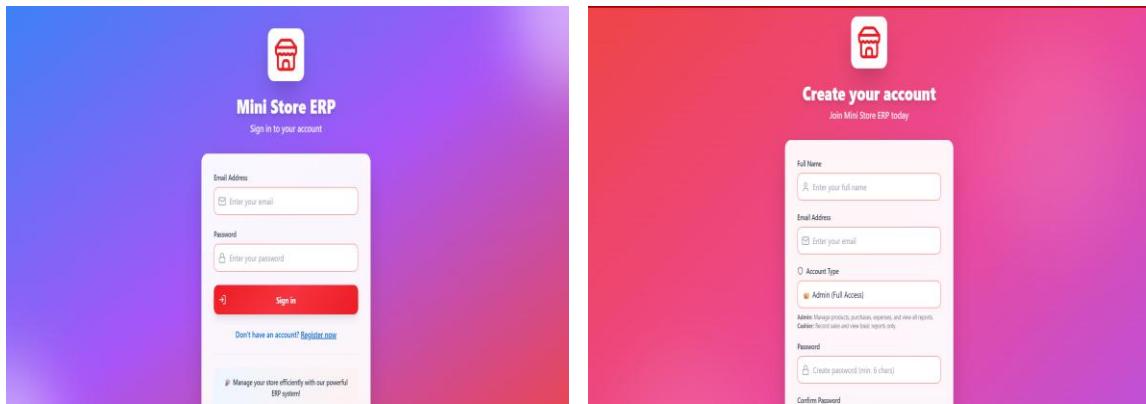
```
VITE_API_BASE_URL="enter backend deployment url(production)"
```

11. Future Enhancements

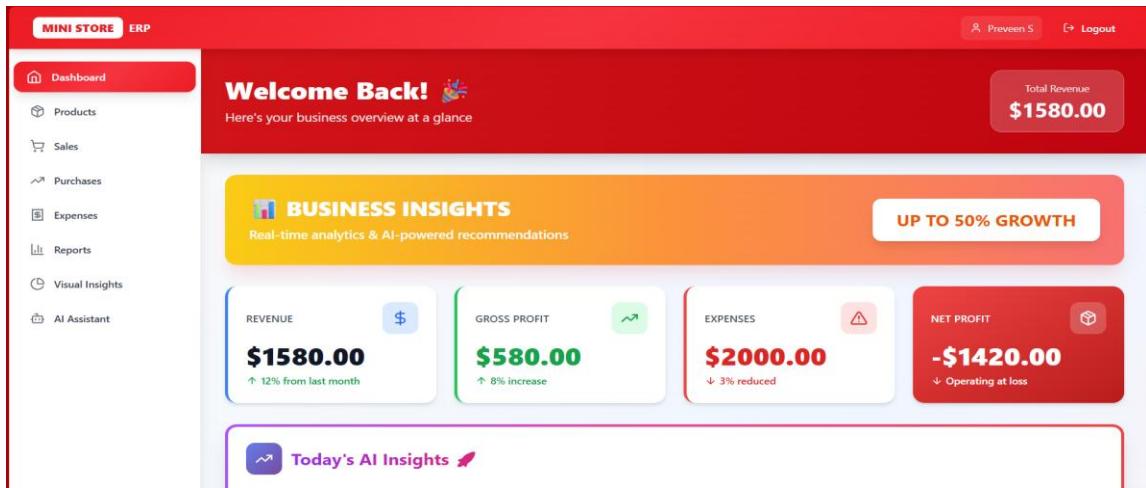
- Multi-branch store support
 - Barcode scanner support
 - Mobile app version
 - Staff attendance & payroll
-

12. Project Screenshots

- Login and Register Page



- Dashboard Page



- **Products Page**

| SKU | NAME | CATEGORY | COST PRICE | SELLING PRICE | STOCK | ACTIONS |
|------------|--------------|-------------|------------|---------------|-------|---------|
| JNS-BLU-M | Jeans Blue | Clothing | \$15.00 | \$45.00 | 50 | |
| COF-001 | Coffee Beans | Groceries | \$5.00 | \$12.00 | 100 | |
| LAP-HP-001 | Laptop HP | Electronics | \$500.00 | \$800.00 | 28 | |

- **Visual Insights Page**

Visual Insights Dashboard
Real-time analytics and business intelligence

Total Revenue **\$1580.00** Profit Margin: 36.7%

Total Sales **1** Avg: \$1580.00 per sale

Gross Profit **\$580.00** Net: \$1420.00

Low Stock Alerts **0** All good

Sales Trend

Revenue Analysis

- **Sales Invoice Page**

Sales

Invoice Details

INVOICE
#INV-2025-00001

Date: 11/11/2025 Time: 7:17:34 pm
Customer: John Smith Phone: 555-1234
Payment Method: Cash

| Item | Qty | Price | Total |
|-----------|-----|----------|-----------|
| Laptop HP | 2 | \$800.00 | \$1600.00 |

Subtotal: \$1600.00
Discount: -\$20.00
Tax: \$0.00
Total: \$1580.00

Thank you for your business!

Average Sale **\$1580.00**

End Date dd-mm-yyyy
\$1580.00

- **Reports Page**

The screenshot shows the 'Reports & Analytics' page. On the left, a sidebar menu includes: Dashboard, Products, Sales, Purchases, Expenses, Reports (highlighted in red), Visual Insights, and AI Assistant. The main area has a title 'Reports & Analytics' and a section 'Want to dig deeper into your data?'. It features a 'Time Period' dropdown set to 'Last 30 Days' and a 'LOAD REPORTS' button. Below this are five summary boxes: Revenue (\$1580.00), Gross Profit (\$580.00), Expenses (\$2000.00), Net Profit (-\$1420.00), and Profit Margin (36.71%). A 'Strategic AI Analysis' section titled 'What Your Numbers Tell You (30d)' contains a message: 'Here's what I found in your data: it seems like your business has had a bit of a tough time over the past 30 days. You've had some sales, but your expenses have been higher than your revenue, which is affecting your bottom line. Let's break it down to see what's going on and what you can do about it.'

- **AI Assistant Page**

The screenshot shows the 'AI Assistant' page. The sidebar menu includes: Purchases, Expenses, Reports, Visual Insights, and AI Assistant (highlighted in red). The main area has a 'Suggested Questions' section with the following prompts: 'What are my top-selling products this month?', 'Show me products with low stock levels', 'What is my current profit margin and financial health?', 'Which expenses are highest and how can I reduce costs?', 'What should I focus on to improve sales and profitability?', 'How much inventory should I reorder for my low stock items?', and 'What are my recent sales trends compared to expenses?'. To the right is a 'Chat with AI Assistant' section with a purple icon, showing '0 messages'. It includes a welcome message: 'Welcome to your AI Business Assistant!', a note: 'Ask me anything about your business data and I'll provide insights based on your real-time store information including sales, inventory, expenses, and financial performance.', example questions: 'What's my current financial performance?', 'Which products need restocking urgently?', 'How can I improve my profit margins?', and 'What are my highest expenses this month?'. At the bottom is a text input field with placeholder 'Ask me about sales, products, inventory, expenses, or business insights...' and a 'SEND' button.

13. Conclusion

This documentation guide provides the complete overview needed for understanding, deploying, evaluating, and extending the Mini-Store-ERP application. For additional details, refer to README files and source code comments.