🛒 RevShop - Console E-Commerce Application

RevShop is a secure, console-based e-commerce platform designed for both buyers and sellers. Built with Java and MySQL, it provides complete e-commerce functionality including user registration, product browsing, shopping cart management, order processing, and inventory management through a robust command-line interface.

This project demonstrates professional backend development using layered architecture principles, JDBC connectivity, and industry-standard design patterns.

BACKEND STACK

🛠️ Technology Stack

Java 21

Modern Java with latest features and performance improvements

MySQL Database

Relational database for structured data storage

JDBC

Database connectivity and SQL operations

Log4j2

Comprehensive logging and monitoring

JUnit

Unit testing framework for quality assurance

Maven

Build automation and dependency management

✨ Core Features

👤 Buyer Features

- User registration and secure authentication
- Browse products by category with filtering
- Advanced product search by name
- Shopping cart management (add/remove items)
- Complete checkout with shipping and billing
- Order history tracking and viewing
- Product reviews and rating system
- Favourite products list
- Simulated payment processing

🏪 Seller Features

- Seller registration with business details
- Comprehensive product inventory management
- Order viewing and fulfilment management
- Dynamic MRP and discount price setting
- Inventory threshold alerts and notifications
- Product review monitoring dashboard

Database Setup & Configuration

Setup Steps

01

Install MySQL Server

Download and configure MySQL on your local machine

02

Create Database

Create a new database (e.g., revshop_db)

03

Run Schema Script

Execute src/sql/schema.sql to create all tables

04

Update Credentials

Configure src/main/resources/database.properties

Sample Configuration

```
db.url=jdbc:mysql://localhost:3306/revshop_db
db.username=root
db.password=your_password
```

📁 Project Structure

```
RevShop/
├── src/
│   ├── main/
│   │   ├── java/com/revshop/
│   │   │   ├── MainApplication.java
│   │   │   ├── config/
│   │   │   │   ├── DatabaseConfig.java
│   │   │   │   └── LoggerConfig.java
│   │   │   ├── dao/
│   │   │   │   ├── UserDAO.java
│   │   │   │   ├── ProductDAO.java
│   │   │   │   ├── OrderDAO.java
│   │   │   │   └── CartDAO.java
│   │   │   ├── model/
│   │   │   │   ├── User.java
│   │   │   │   ├── Product.java
│   │   │   │   └── Order.java
│   │   │   ├── service/
│   │   │   │   ├── AuthService.java
│   │   │   │   ├── BuyerService.java
│   │   │   │   └── SellerService.java
│   │   │   ├── util/
│   │   │   └── menu/
│   │   └── resources/
│   └── test/
└── pom.xml
```

Architecture Layers

Presentation Layer

Console menus and user interaction handlers

Service Layer

Business logic and transaction management

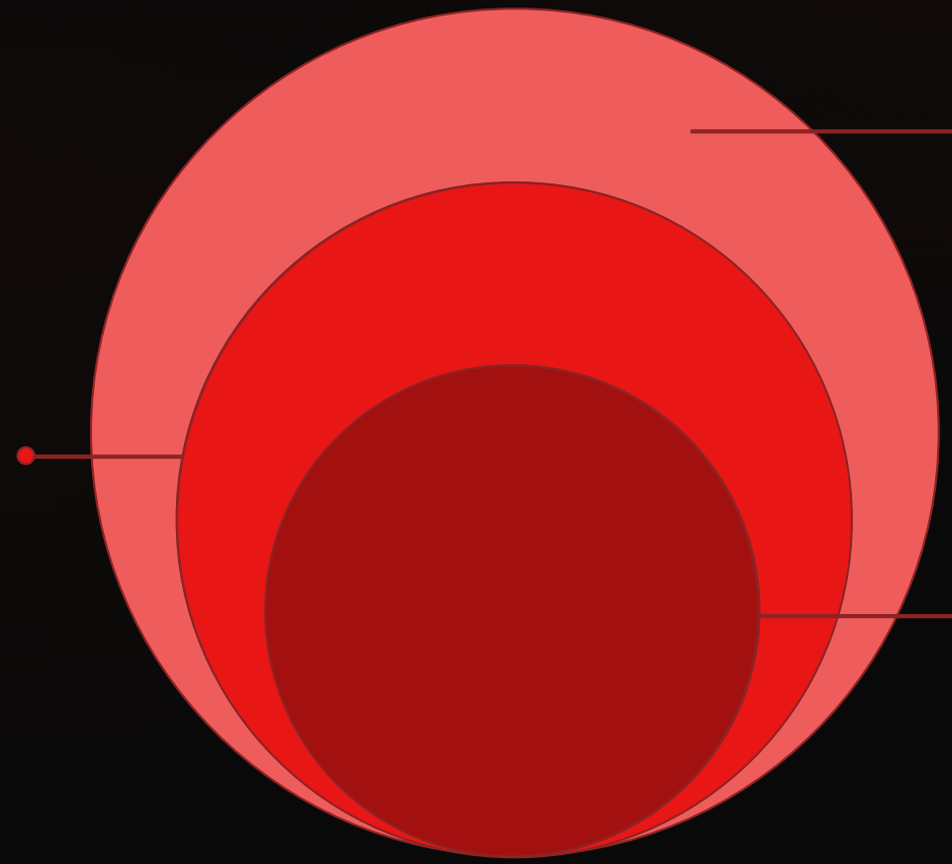
DAO Layer

Data access objects for database operations

Model Layer

Entity classes representing data structures

📐 Class Diagram Architecture



DAO & Persistence

DAOs -> DatabaseUtil -> MySQL storage

MainMenu & Services

MainMenu routes to Auth, Buyer, Seller services

MainApplication

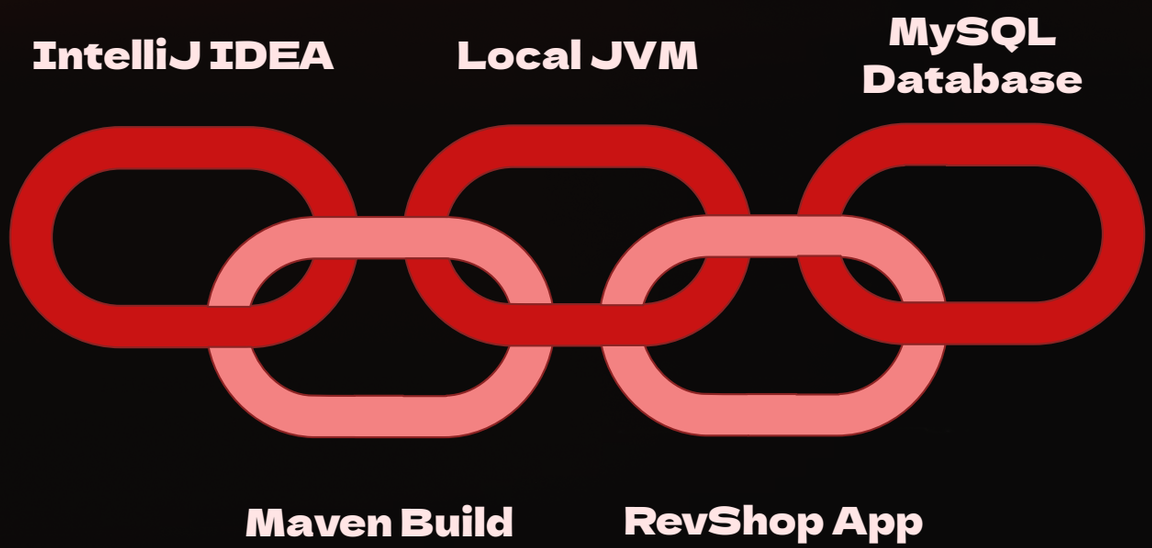
Initialises system and entry point

The application follows a clean layered architecture with clear separation of concerns. The MainApplication initialises the system, while MainMenu orchestrates user interactions. Service classes handle business logic, DAO classes manage database operations, and DatabaseUtil provides connection pooling and resource management.

⚡ Sequence Diagram: User Login Flow

The login flow demonstrates secure authentication with password verification. User credentials are validated through the service layer, with the DAO layer handling database queries. This separation ensures security best practices and maintains clean architecture principles throughout the authentication process.

🚀 Deployment Architecture



Environment Components

Development IDE

IntelliJ IDEA for code development and debugging

Build System

Maven handles compilation, dependencies, and packaging

Runtime Environment

Local JVM executes the compiled application

Data Persistence

MySQL database stores all application data locally

GETTING STARTED

▶️ Running & Testing the Application

Running the Application

1 Configure Database

Update database.properties with your MySQL credentials

2 Build Project

```
mvn clean compile
```

3 Execute Application

```
mvn exec:java -
Dexec.mainClass="com.revshop.MainApplication"
```

🧪 Testing

Run comprehensive unit tests to verify functionality:

```
mvn test
```

🚀 Future Enhancements

- Web-based user interface
- Microservices architecture
- Email notification system
- Advanced payment gateway integration
- AI-powered recommendation engine

Thank You

RevShop Console E-Commerce Application

JAVA BACKEND | MYSQL | MAVEN