E-Commerce CLI Application

Complete User Manual & Setup Guide

A comprehensive guide for setting up and running the
E-Commerce Command Line Interface Application

Version: 1.0.0

Date: December 2024

Platform: Windows / macOS / Linux

📋 Table of Contents

1. Introduction

2. System Requirements

3. Installing Java 17

4. Installing PostgreSQL

5. Database Setup

6. Running the Application

7. Using the CLI Interface

8. Verifying Data in Database

9. Complete Workflow Example

10. Troubleshooting Guide

11. Quick Reference

1. Introduction

The E-Commerce CLI Application is a Java Spring Boot application that provides a command-line interface for managing an online shopping experience. Users can register, browse products, add items to cart, place orders, and process payments.

Key Features

- User registration and authentication with JWT tokens
- Product catalog browsing
- Shopping cart management
- Order placement and tracking
- Payment processing simulation
- PostgreSQL database for data persistence

2. System Requirements

Component	Requirement
Operating System	Windows 10/11, macOS 10.14+, or Linux
Java	JDK 17 or higher
Database	PostgreSQL 12 or higher
RAM	Minimum 4 GB (8 GB recommended)
Disk Space	500 MB for application + database
Network	Port 8080 (application) and 5432 (PostgreSQL)

3. Installing Java 17

3.1 For Windows

1 Download Java 17

Visit: `https://adoptium.net/temurin/releases/?version=17`

Select **Windows x64** and download the **.msi** installer.

2 Run the Installer

Double-click the downloaded file and follow the installation wizard.

Accept the default settings and complete installation.

3 Verify Installation

Open **Command Prompt** and run:

```
java -version
```

Expected output: `openjdk version "17.x.x"`

3.2 For macOS

Install using Homebrew:

```
brew install openjdk@17
echo 'export PATH="/opt/homebrew/opt/openjdk@17/bin:$PATH"' >> ~/.zshrc
source ~/.zshrc
java -version
```

3.3 For Linux (Ubuntu/Debian)

```
sudo apt update
sudo apt install openjdk-17-jdk
java -version
```

4. Installing PostgreSQL

4.1 For Windows

1 Download PostgreSQL

Visit: `https://www.postgresql.org/download/windows/`

Download the Windows installer.

2 Run the Installer

- Accept default installation directory
- Select all components
- **Important:** Remember the password you set for `postgres` user
- Keep default port: **5432**
- Complete the installation

3 Verify Installation

Open **Command Prompt** and run:

```
psql -U postgres
```

Enter your password. If you see `postgres=#` , PostgreSQL is working.

4.2 For macOS

```
brew install postgresql@15
brew services start postgresql@15
```

```
echo 'export PATH="/opt/homebrew/opt/postgresql@15/bin:$PATH"' >> ~/.zshrc
source ~/.zshrc
```

4.3 For Linux (Ubuntu/Debian)

```
sudo apt update
sudo apt install postgresql postgresql-contrib
sudo systemctl start postgresql
sudo systemctl enable postgresql
```

5. Database Setup

5.1 Create the Database

1 Connect to PostgreSQL

Open Command Prompt (Windows) or Terminal (macOS/Linux):

```
psql -U postgres
```

Enter your postgres password when prompted.

2 Create Database

```
CREATE DATABASE e_com;
```

You should see: `CREATE DATABASE`

3 Verify Database Creation

```
\l
```

Look for `e_com` in the list of databases.

4 Set Password (Optional)

To match the application configuration:

```
ALTER USER postgres WITH PASSWORD 'sherupass';
```

Or update the application config file with your password (see Section 5.2).

5 Exit PostgreSQL

```
\q
```

5.2 Configure Application Password

If you want to use your own PostgreSQL password instead of `sherupass` :

1 Open Configuration File

Navigate to: `src/main/resources/application.properties`

2 Update Password

Find this line:

```
spring.datasource.password=sherupass
```

Change to your password:

```
spring.datasource.password=YOUR_PASSWORD
```

5.3 Add Sample Data

Connect to the e_com database and add sample products:

```
psql -U postgres -d e_com
```

Run these SQL commands:

Add Categories:

```
INSERT INTO category_table (category_id, category_name) VALUES (1, 'Electronics');
INSERT INTO category_table (category_id, category_name) VALUES (2, 'Clothing');
INSERT INTO category_table (category_id, category_name) VALUES (3, 'Books');
```

Add Products:

```
INSERT INTO product_table (product_id, product_name, description, price, stock_present, cate
VALUES (1, 'iPhone 15', 'Latest Apple smartphone', 999.99, 50, 1, NOW());

INSERT INTO product_table (product_id, product_name, description, price, stock_present, cate
VALUES (2, 'Samsung Galaxy S24', 'Android flagship phone', 899.99, 30, 1, NOW());

INSERT INTO product_table (product_id, product_name, description, price, stock_present, cate
VALUES (3, 'Cotton T-Shirt', 'Comfortable cotton t-shirt', 29.99, 100, 2, NOW());

INSERT INTO product_table (product_id, product_name, description, price, stock_present, cate
VALUES (4, 'Java Programming Book', 'Learn Java from scratch', 49.99, 25, 3, NOW());
```

Update Sequences:

```
SELECT setval('category_table_category_id_seq', 10);
SELECT setval('product_table_product_id_seq', 10);
```

Exit with: `\q`

6. Running the Application

6.1 Navigate to Project Directory

Windows (Command Prompt):

```
cd C:\path\to\ecommerce4-Updated
```

macOS/Linux (Terminal):

```
cd /path/to/ecommerce4-Updated
```

6.2 Start the Application

Windows:

```
mvnw.cmd spring-boot:run
```

macOS/Linux:

```
./mvnw spring-boot:run
```

⚠ **First Run:** The first time you run the application, it will download dependencies. This may take 2-5 minutes depending on your internet speed.

6.3 Successful Startup

When the application starts successfully, you'll see the CLI menu:

```
==== E-COMMERCE CLI ====
1. Register
2. Login
3. View All Products
4. Add Product to Cart
5. View Cart
6. Checkout / Place Order
7. Initiate Payment
8. Verify Payment
9. Exit
Select option:
```

7. Using the CLI Interface

7.1 Menu Options

Option	Description	Login Required
1. Register	Create a new user account	No
2. Login	Log into your account	No
3. View All Products	Display all available products	No
4. Add Product to Cart	Add a product to shopping cart	Yes
5. View Cart	View items in your cart	Yes
6. Checkout	Place an order from cart items	Yes
7. Initiate Payment	Start payment for an order	Yes
8. Verify Payment	Complete payment verification	Yes
9. Exit	Close the application	No

7.2 Typical User Flow

> **Recommended Order:**
>
> Register → Login → View Products → Add to Cart → Checkout → Initiate Payment → Verify Payment

8. Verifying Data in Database

Open a **second terminal window** to monitor database changes while using the CLI.

8.1 Connect to Database

```
psql -U postgres -d e_com
```

8.2 Useful SQL Commands

Purpose	SQL Command
List all tables	`\dt`
View all users	`SELECT * FROM user_table;`
View all products	`SELECT * FROM product_table;`
View categories	`SELECT * FROM category_table;`
View addresses	`SELECT * FROM addresses;`
View shopping carts	`SELECT * FROM carts;`
View cart items	`SELECT * FROM cart_items;`
View orders	`SELECT * FROM orders;`
View order items	`SELECT * FROM order_items;`
View payments	`SELECT * FROM payments;`
Exit psql	`\q`

9. Complete Workflow Example

Step A: Register a User

CLI Input:

```
Select option: 1
Enter name: John Doe
Enter email: john@example.com
Enter password: password123
Enter phone: 9876543210
```

Verify in Database:

```
SELECT id, name, email FROM user_table;
```

Step B: Login

CLI Input:

```
Select option: 2
Enter email: john@example.com
Enter password: password123
```

Output: Login successful! Your User ID = 1

Step C: View Products

CLI Input:

```
Select option: 3
```

Products will be displayed with their IDs.

Step D: Add Product to Cart

CLI Input:

```
Select option: 4
Enter product ID to add: 1
```

Verify in Database:

```
SELECT * FROM carts;
SELECT * FROM cart_items;
```

Step E: Add Address (Before Checkout)

Important: You need to add an address before checkout. Run this in psql:

```
INSERT INTO addresses (street, city, state, zipcode, country, user_id)
VALUES ('123 Main St', 'New York', 'NY', '10001', 'USA', 1);
```

Step F: Checkout

CLI Input:

```
Select option: 6
Enter addressId: 1
```

```
Enter payment method (COD / UPI / CARD): UPI
```

Verify in Database:

```
SELECT * FROM orders;
SELECT * FROM order_items;
```

Step G: Initiate Payment

CLI Input:

```
Select option: 7
Enter Order ID to initiate payment: 1
Enter Payment method: UPI
```

Note: Save the Payment ID and Transaction ID shown.

Step H: Verify Payment

CLI Input:

```
Select option: 8
Enter Payment ID: 1
Enter Transaction ID: [from previous step]
Was the payment successful? (yes/no): yes
```

Verify in Database:

```
SELECT * FROM payments;
SELECT order_id, status FROM orders;
```

10. Troubleshooting Guide

10.1 Database Connection Errors

Error: `FATAL: database "e_com" does not exist`

Solution: Create the database:

```
psql -U postgres
CREATE DATABASE e_com;
\q
```

Error: `Connection refused`

Solution (Windows):

1. Press `Win + R`, type `services.msc`
2. Find **postgresql-x64-XX**
3. Right-click → Start

Error: `Authentication failed`

Solution: Update password in `application.properties`:

```
spring.datasource.password=YOUR_ACTUAL_PASSWORD
```

10.2 Application Errors

Error: `Port 8080 already in use`

Solution: Stop other applications using port 8080, or change port in `application.properties`:

```
server.port=8081
```

Error: `Please login first`

Solution: You must login (Option 2) before using cart, checkout, or payment features.

10.3 Data Issues

Issue: No products showing

Solution: Add sample products using SQL commands in Section 5.3

11. Quick Reference Card

Application Commands

Action	Windows	macOS/Linux
Start Application	`mvnw.cmd spring-boot:run`	`./mvnw spring-boot:run`
Stop Application	`Ctrl + C`	`Ctrl + C`

PostgreSQL Commands

Action	Command
Connect to PostgreSQL	`psql -U postgres`
Connect to e_com database	`psql -U postgres -d e_com`
List databases	`\l`
List tables	`\dt`
Describe table	`\d table_name`
Exit psql	`\q`

Database Tables

Table Name	Description
user_table	User accounts
product_table	Product catalog
category_table	Product categories

addresses	User addresses
carts	Shopping carts
cart_items	Items in carts
orders	Placed orders
order_items	Items in orders
payments	Payment records

Configuration File

Location: `src/main/resources/application.properties`

```
spring.application.name=ecommerce
spring.datasource.url=jdbc:postgresql://localhost:5432/e_com
spring.datasource.username=postgres
spring.datasource.password=sherupass
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

E-Commerce CLI Application - User Manual

Version 1.0.0 | December 2024