DMart Clone – Scalable E-Commerce Platform

Status In progress

Timing Jun 1, 2025 to Aug 31, 2025

Owners Saurabh Sonawane

Software Requirements Specification (SRS) Document

Overview

Background

With the rapid growth of digital commerce and increasing consumer preference for online grocery shopping, businesses like DMart require scalable and efficient platforms to serve a large user base. A digital presence not only enhances customer experience but also streamlines operations like inventory and order management.

Problem Statement

DMart's traditional retail operations face limitations in accessibility, convenience, and scalability without a modern e-commerce platform. Existing solutions may lack customization, performance, or alignment with DMart's specific business workflows. A robust, scalable, and modular online solution is essential to meet consumer demands and ensure smooth backend operations.

Proposed Solution

This project aims to develop a full-featured, production-grade **DMart clone web application** that replicates the core functionality of www.dmart.in. The platform will be built using:

- **MySQL** for relational data storage with scalability in mind
- **Spring Boot (Java)** for a robust, modular backend API architecture
- **React.js** for a responsive, user-friendly frontend

The application will support core features like user authentication, product browsing, cart management, order processing, inventory tracking, and admin control. It will be designed using a **database-first approach** to ensure accurate data modeling and maintainability. The system will be scalable, secure, and extensible for future enhancements such as mobile apps, advanced analytics, and third-party integrations.

Note:

This project is developed as a **technical skills showcase and prototype**, following **industry-standard practices** and simulating the **full software development lifecycle (SDLC)** — including requirement analysis, architectural design, database-first development, modular coding, testing, documentation, and deployment.

It is intended to demonstrate proficiency in backend and frontend technologies, clean architecture principles, and production-grade application design aligned with real-world enterprise standards.

Objectives

To design and develop a robust, scalable, and modular e-commerce web application that replicates the business functionalities of www.dmart.in using modern technologies such as Spring Boot (Java), MySQL, and React.js. This system will support online shopping, inventory, order, and user management, optimized for scalability and future expansion.

3. Business Requirements Overview

3.1 Business Need

DMart is a leading retail chain offering groceries and home essentials. With increasing online demand, a clone application is needed to:

- Facilitate online purchases and delivery.
- Improve user experience and engagement.
- Enable operational efficiency in inventory and order management.

3.2 Business Goals

- Enable seamless browsing, cart, checkout, and payment features.
- Provide an admin dashboard for catalog, inventory, and order management.
- Support scalability for high user and transaction volume.
- Facilitate promotional and marketing campaigns.

---

4. Functional Requirements

- **User Module**: Registration, login, role-based access (admin/customer), profile, address management.
- **Product Module**: Category browsing, product details, filtering, product attributes.
- **Cart & Checkout**: Add to cart, quantity management, address selection, coupon application.
- **Order Management**: Place orders, view status, order history, and cancellation.
- **Inventory Management**: Stock levels, transaction logs.
- **Payment Integration**: Support multiple payment methods and status tracking.
- **Reviews & Ratings**: Authenticated user feedback.
- **Promotions**: Discount rules, coupon system, validity tracking.

Requirements for a Dmart Ready E-commerce Website.

1. **Create Accounts** : The User can create accounts and log in to the website.
2. **Contact Details** : Users can store their name and contact details such as email id and phone number.
3. **Addresses** : Users can add one or more addresses to their account and can set a default shipping address

4. **Payment Methods** : - Users can add one or more payment methods to their account and can set a default payment method.
5. **Products** : - The website can store a large number of products.
6. **Categories** : - Each product belongs in a category, and categories can belong to other categories.
7. **Product variations** : Each product can have different variations, such as different colours or sizes. Each of these variations (e.g. colour) can have different values.
8. **Number in Stock** : - The website should keep a track of the number of each product that is in stock.
9. **Shopping cart** : - Visitors can add one or more products to their shopping cart as part of their shopping experience . Shopping carts are not saved in the database unless they are logged in.
10. **Payment details for an Order** : - The User needs to provide their payment details and address as part of placing the order.
11. **Shipping method** : - The user can select a shipping method from a list of methods. Each shipping method has a single standard price.
12. **Order Status : -** The order and shipping process follows several stages once the order is placed, such as created, in transit, and delivered.
13. **Reviews :** - Users can leave reviews for products they have purchased, which include a rating from 1-5 and a text comment.
14. **Promotions : -** The website allows promotion or sales to be run, which allows for one or more product categories to have a specific discount on their price.

5. Non-Functional Requirements

- **Scalability**: Microservice-friendly schema, database indexing.
- **Performance**: Optimized queries, caching strategy.
- **Security**: Password hashing, JWT authentication, role-based access.
- **Maintainability**: Modular code, service-layer abstractions.
- **Documentation**: Swagger for APIs, DBML for schema.

6. Technical Stack

Layer	Technology	Reason
Frontend	React.js	Component-based, fast UI, rich ecosystem
Backend	Spring Boot (Java)	Production-grade framework, REST APIs
Database	MySQL	Relational integrity, indexing, and scalability
ORM	JPA / Hibernate	Mapping DB to Java entities cleanly
Build Tool	Maven	Lifecycle and dependency management
API Docs	Swagger	Interactive API documentation
Schema Modeling	DBML (dbdiagram.io)	Visual schema design
Version Control	Git & GitHub	Code collaboration and tracking

7. Architecture Overview

- Layered architecture: Controller → Service → Repository → DB
- DTOs and Entities are separated for clean boundaries
- RESTful APIs for frontend-backend communication
- Database-first approach using reverse-engineering
- Modular packages: `user`, `product`, `order`, `cart`, `payment`, etc.

8. Strategy and Roadmap

Phase 1 : Planning & Design

- Requirement analysis
- ER diagram & schema design
- Project structure (Spring Boot + Maven)

Phase 2 : Development

- User & Auth modules

- Product and catalog browsing
- Cart and checkout
- Order placement and payments
- Admin panels

Phase 3 : Testing & Deployment

- Unit + Integration tests
- Swagger testing
- Dockerization (optional)
- Deployment to cloud/local environment

9. Future Scope

- Progressive Web App (PWA) support
- Mobile app using React Native
- Microservice conversion (user, product, order services)
- Advanced analytics dashboard for admin
- Loyalty programs, wallet integration, and chat support

11. Appendix

- ER Diagram (DBML shared via dbdiagram.io)
- Source Code Repository (GitHub)
- API Reference (Swagger Docs)

Roles

Role	Name
Owners	Saurabh Sonawane
Approvers	Person Person
Contributors	Hemant Patil Sangram_Rananaware asmitmeshram5@gmail.com

About This Project

This DMart Clone project is created as part of a personal learning initiative and technical portfolio to showcase full-stack development capabilities. It is designed as a realistic prototype with end-to-end functionality, structured like a production-grade enterprise application, and built using best practices from real-world software engineering processes.p