



## **Experiment - 2**

**Student Name:** Sameep Das

**UID:** 23BCS11375

**Branch:** BE-CSE

**Section/Group:** KRG-1A

**Semester:** 6<sup>th</sup>

**Date of Performance:** 12/01/26

**Subject Name:** System Design

**Subject Code:** 23CSH-314

### **1. Aim:**

To design and analyze an E-commerce System by identifying its functional and non-functional requirements and representing the system architecture using a draw.io diagram.

### **2. Objectives:**

1. To understand the working of a E-commerce system
2. To identify functional requirements of the system
3. To identify non-functional requirements such as performance and scalability
4. To design a high-level system flow using draw.io
5. To design a DB schema for the system using ERD.
6. To improve understanding of real-world system design concepts

### **3. Procedure-**

- Studied real-world E-commerce platforms such as Amazon and Flipkart.
- Identified core components including users, products, carts, orders, inventory, and payments.
- Listed functional requirements required for smooth E-commerce operations.
- Analyzed non-functional requirements like low latency, scalability, and fault tolerance.
- Designed a structured system design diagram using draw.io.
- Designed a DB schema using DB scripts and ERD.
- Reviewed the design to ensure scalability, data consistency, and clarity.

### **4. Functional Requirements -**

- User registration and authentication
- Browse and search products by category
- Add products to cart
- Update or remove items from cart
- Place orders and process payments
- Manage inventory and product availability
- View order history
- Provide product reviews and ratings



## 5. Non-functional Requirements

- Low latency response time (< 300 ms)
- High availability (99.9% uptime)
- Scalability to support millions of concurrent users
- Strong consistency for inventory and payments
- Secure handling of user and payment data

## 6. High Level Design (HLD)

The system consists of Client, Frontend, Backend Services (Product, Cart, Order, Payment, Inventory), and Database layers. High availability is ensured for search, while consistency is enforced for payments and inventory. The system supports horizontal and vertical scaling.

## 7. Low-Level Design (LLD)

### Product Module

- Stores product details and inventory.
- Supports keyword-based search with pagination.

### Cart Module

- Maintains user-specific cart.
- Supports add, update, and delete operations.

### Order and Payment Module

- Generates order ID after checkout.
- Confirms payment and updates order status.

### Inventory and Race Condition Handling

- Manages limited stock.
- Uses locking or transactions during flash sales

## 8. Outcome / Result -

- Successfully designed an E-commerce system using draw.io.
- Identified functional and non-functional requirements clearly.
- Understood scalability and consistency challenges in large-scale systems.
- Gained hands-on experience in system design and architectural thinking.