

Experiment No:2

Student Name: Sumanyu Singh
Branch: BE CSE
Semester: 6th
Subject Name: System Design
SubjectCode:23CSH-314

UID:23BCS10801
Section/Group:KRG-3B
Date of Performance:14/01/2026

Aim: To design an online E-commerce platform similar to Amazon/Flipkart for browsing and purchasing products like mobiles, laptops, cameras, and clothes. To implement Kafka, Elasticsearch, and a CDC pipeline for real-time data processing, fast search, and scalability.

Objective:

- To develop a scalable online E-commerce system for product listing, search, and order management.
- To use Apache Kafka for real-time event streaming and inter-service communication.
- To implement Elasticsearch for fast and efficient product search.
- To integrate a CDC pipeline for real-time data synchronization.

Tool required:

- Programming Language: Java / Python / JavaScript
- Backend Framework: Spring Boot / Express.js / Flask
- Database: MySQL / PostgreSQL / MongoDB
- API Testing Tool: Postman
- Design Tool: Draw.io
- Web Browser
- Apache Kafka
- Elasticsearch
- CDC Connector

Functional Requirements :

- The user should be able to search products by name or title.
- The user should be able to view product details such as description, images, price, quantity, and reviews.
- The user should be able to add products to the cart and update quantities.
- The user should be able to perform checkout and make payments.
- The user should be able to track the order status.
- The system should manage purchases for products with limited stock.

Non - Functional Requirements :

- **Target Scale:** Support up to **100 million DAU** with **10 orders per second**.
- **Availability:** System must be available **24/7**, with **high availability for product search**.
- **Consistency:** Strong consistency required for:
 - Payment and order placement
 - Inventory management
- **Latency:** Average response time should be **~200 ms**.
- **Scalability:** Support **horizontal and vertical scaling**.

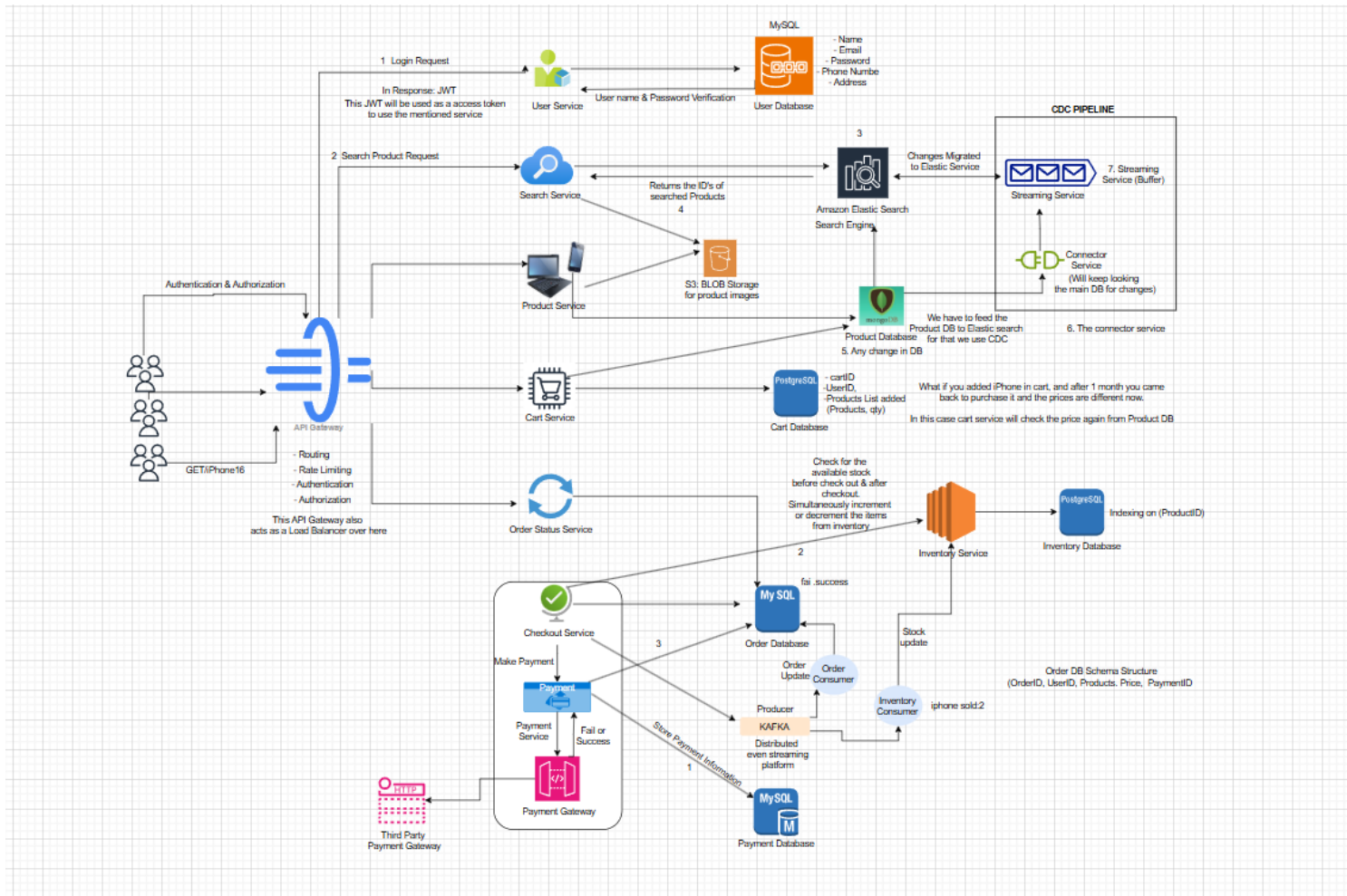
Core-Entites of the System:

1. User / Client
2. Product
3. Cart
4. Orders
5. Checkout followed by Payment

API Designing (LLD)

1. **GET API – Product Search**
`https://localhost/products/search_item?query={search_keywords}`
 - Used to search products by keyword
 - Pagination is applied to reduce latency
2. **GET API – View Product Details**
`https://localhost/products/{product_id}`
 - Fetches complete details of a selected product
3. **POST API – Add Item to Cart**
`https://localhost/cart/add_products`
 - Adds selected products to the user's cart
4. **PUT API – Update Cart Item**
 - Used to update product quantity in the cart
5. **DELETE API – Remove Item from Cart**
 - Removes a product from the cart
6. **POST API – Checkout & Payment**
 - Checkout: `https://localhost/checkout`
 - Payment: `https://localhost/payment`
 - Places order and processes payment
7. **GET API – Order Status**
`https://localhost/order_status?order_id={order_id}`
 - Retrieves current order status

HLD(High level Diagram) -> FINAL :



Learning outcomes:

- Understand the architecture of a scalable E-commerce system.
- Gain hands-on experience with Apache Kafka for real-time data streaming.
- Learn to implement fast and efficient product search using Elasticsearch.
- Understand Change Data Capture (CDC) for real-time data synchronization.
- Develop skills in designing distributed systems with high availability and scalability.
- Learn how microservices communicate using event-driven architecture.
- Understand API design principles for large-scale applications.
- Gain knowledge of handling high traffic and concurrent user requests.
- Learn techniques for reducing latency and improving system performance.
- Understand data consistency, availability, and fault tolerance in distributed systems.