DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment: 2

Student Name: Jay Shankar Kumar UID: 23BCS10408

Branch: B.E.- C.S.E. Section/Group: 23BCS_KRG-1A

Semester: 6 Date of Performance: 12-01-26

Subject Name: System Design Subject Code: 23CSH-314

1. Aim: To design and analyze the system architecture of an E-Commerce platform covering product listing, user and product storage, cart functionality, and payment processing using an event-driven Kafka mechanism.

2. Objective:

    1. To understand the architecture of an E-Commerce platform
    2. To design High Level Design (HLD) and Low Level Design (LLD)
    3. To study user, product, and cart management workflows
    4. To understand asynchronous payment processing using Kafka
    5. To analyze scalability, reliability, and performance of the system

Requirements (Tools):

- Draw.io for system design diagrams
- Backend services (conceptual – microservices based)
- Apache Kafka (for payment event handling)
- Database (SQL / NoSQL – conceptual)
- API testing tool (Postman)

3. Procedure and Output:
    1. Functional Requirements:

- User should be able to register and log in
- User should be able to view product listings
- User should be able to add or remove products from cart
- System should store user, product, and cart data
- Order confirmation should be generated after successful payment

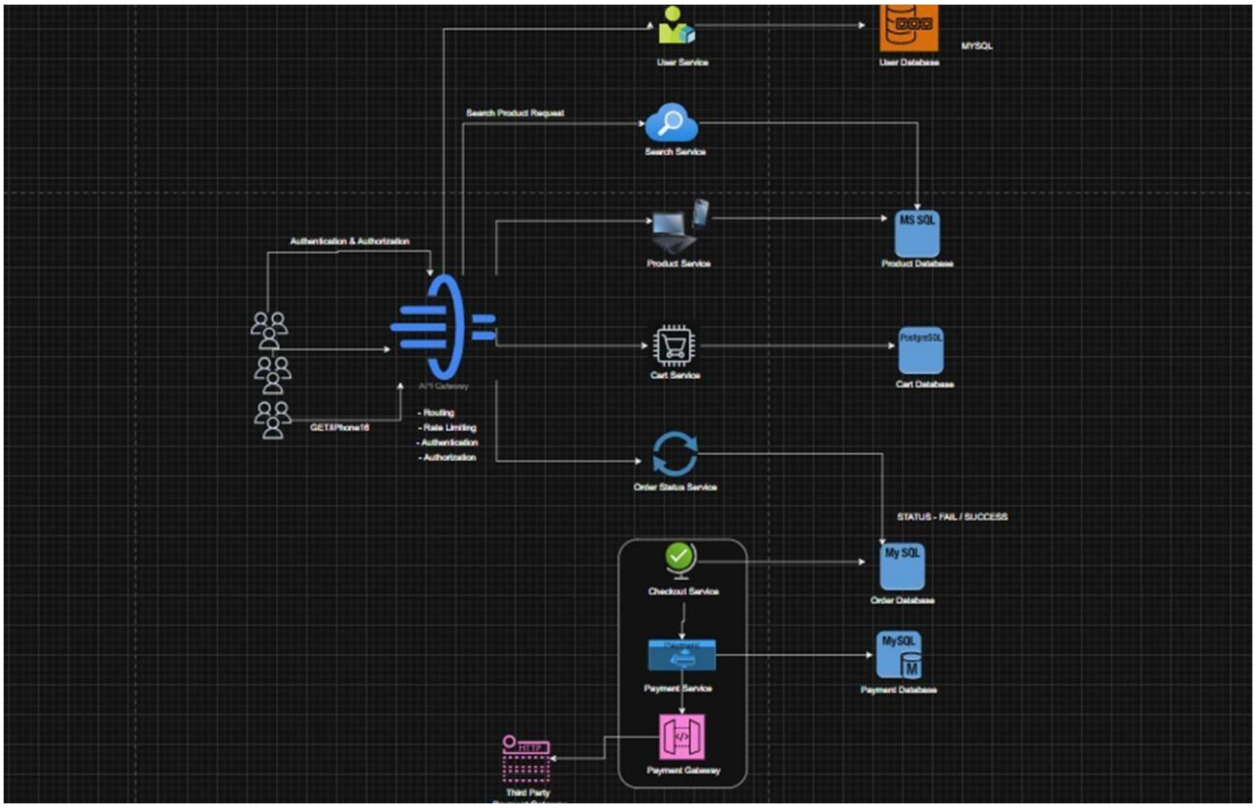
    2. Non-functional requirements:

- Latency: Low response time for product listing and cart operations
- Availability: 24 × 7 system availability
- Scalability: Horizontal scaling of services
- Reliability: Payment processing should be fault-tolerant
- Consistency: Accurate order and payment data
- Security: Secure payment and user authentication

3. High Level Design (HLD)

The High Level Design of the E-Commerce platform consists of the following components:

- Client (Web / Mobile): Interacts with the platform
- API Gateway: Routes requests to appropriate services
- User Service: Manages user data and authentication
- Product Service: Handles product listing and inventory
- Cart Service: Manages cart operations
- Payment Service: Initiates payment processing
- Kafka Broker: Handles asynchronous payment events
- Database: Stores user, product, cart, and order data



4. Low Level Design(LLD)

User Service

- Handles user registration and login
- Stores user details in the User Database

Product Service

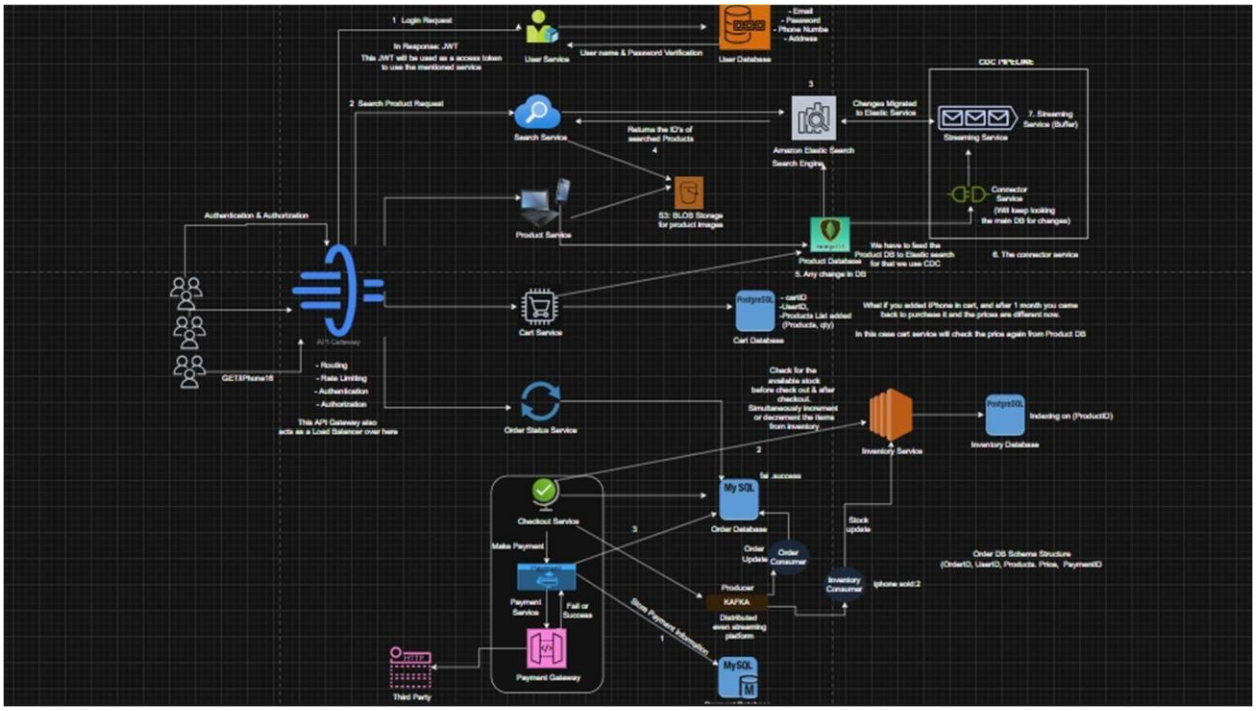
- Fetches product listings
- Stores product details and inventory Cart Service

- Adds and removes products from cart
- Calculates total price

Payment Service (Kafka-based)

- Produces payment request events to Kafka
- Listens for payment status events
- Updates order status after successful payment

Kafka decouples the payment service from other services, improving system reliability and scalability.



4. Learning Outcomes:

Understood E-Commerce system architecture

Learned microservices-based design approach

Gained knowledge of Kafka and event-driven systems

Designed scalable and reliable payment workflows