Experiment No: 02

Student Name: Asmit Prabhakar
UID: 23BCS10063
Branch: BE-CSE
Section/Group: KRG_2B
Semester: 6th
Date of Performance: 14/01/2026
Subject Code: 23CSH-314
Subject Name: System Design

Aim:

To design and implement system design of an e-commerce website

Objective:

- To understand the overall architecture and workflow of an e-commerce website.
- To design a scalable and reliable system for product browsing, order processing, and payments.
- To create High-Level Design (HLD) and Low-Level Design (LLD) diagrams using draw.io.
- To analyze key system design decisions and their impact on performance, scalability, and reliability.
- To understand how components such as frontend, backend services, database, and third-party integrations interact in an e-commerce platform.

Procedure:

- Study the working principles of an e-commerce website and identify its core components such as user interface, product catalog, order management, payment processing, and database.
- Design the High-Level Design (HLD) representing the overall system architecture, including frontend, backend services, database, and third-party integrations, using draw.io.
- Create the Low-Level Design (LLD) illustrating database schemas, APIs, service interactions, and internal workflows using draw.io.
- Define all the functional and non-functional requirements.
- Analyse system latency, throughput, and response time under varying load conditions.
- Analyse the collected performance metrics and evaluate how system design choices impact scalability, performance, and reliability.

Functional Requirements:

- The system shall allow users to register, log in, and log out securely.
- The system shall allow users to browse products by category and view product details such as price, description, and availability.
- The system shall provide a search functionality to find products using keywords.
- The system shall allow users to add products to a shopping cart and update or remove items from the cart.
- The system shall allow users to place orders by completing the checkout process.

- The system shall support online payment processing through integrated payment gateways.
- The system shall allow users to view order history and order status.
- The system shall maintain inventory levels and update stock after each order

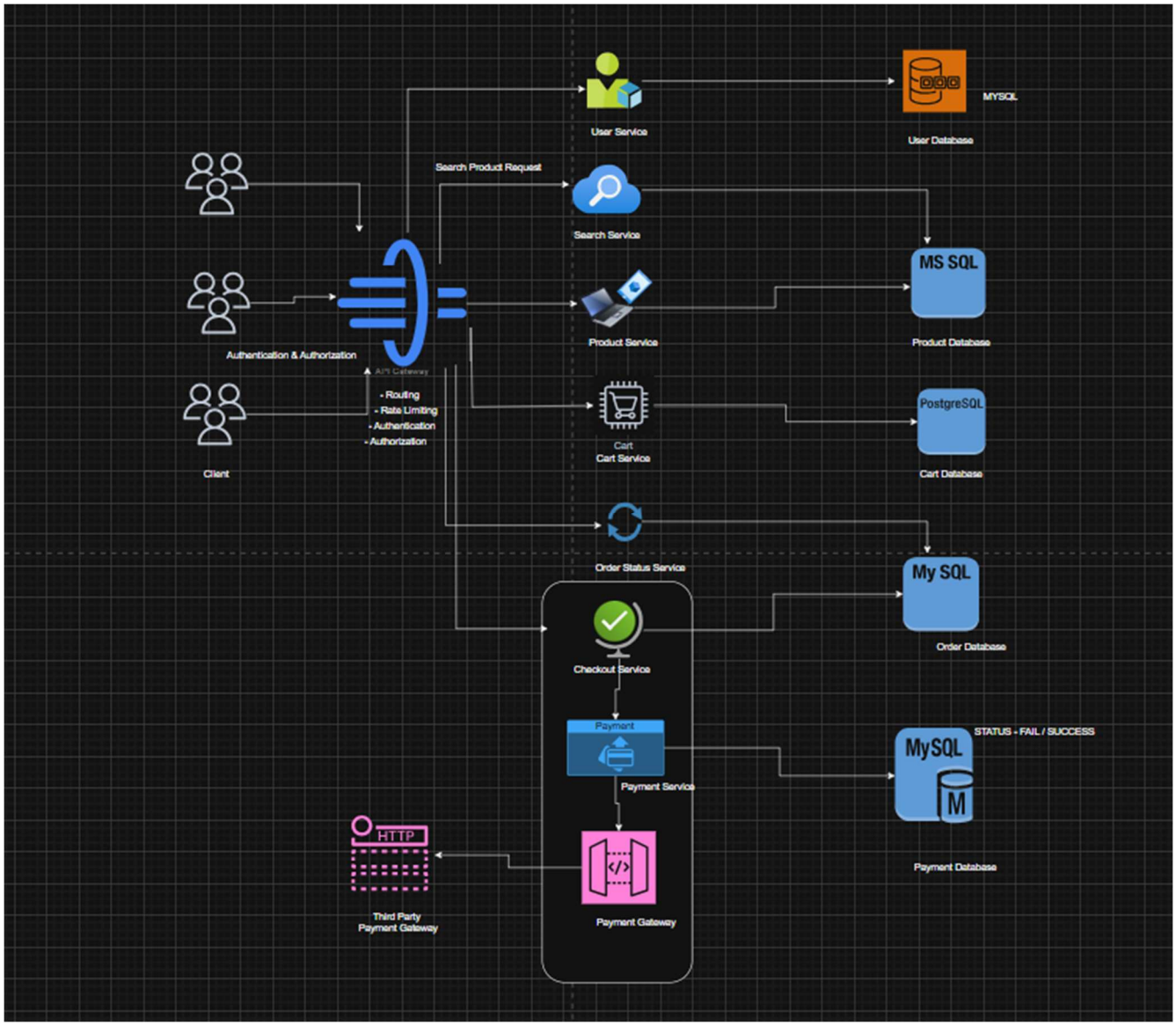
Non-Functional Requirements:

- The system shall support a target scale of 100 million daily active users (DAU) with 10 orders processed per second.
- The system shall balance consistency and availability based on the criticality of each module.
- The product search module shall ensure high availability to allow users to browse products efficiently at all times.
- The payment and order placement modules shall ensure strong consistency to prevent incorrect transactions.
- The inventory management module shall maintain high data consistency to avoid stock mismatches.
- The system shall maintain an average response latency of approximately 200 ms.
- The system shall support both horizontal and vertical scaling to handle traffic growth.

Outcome:

- Designed a scalable microservices-based e-commerce architecture covering search, cart, order, payment, and inventory modules.
- Implemented secure authentication and authorization using an API Gateway and User Service.
- Achieved high availability for product search using Elasticsearch and CDC-based data synchronization.
- Ensured strong consistency for critical workflows such as order placement, payment processing, and inventory updates.
- Designed an event-driven system using Kafka for reliable order and inventory state propagation.
- Enabled efficient product discovery using indexed search and optimized data storage.
- Incorporated fault tolerance and resilience through asynchronous processing and service isolation.
- Analyzed system performance in terms of latency, throughput, and scalability under high user load.
- Demonstrated practical application of HLD and LLD concepts using industry-standard design patterns.
- Gained hands-on experience in designing production-grade distributed systems.

High Level Design: -



Low Level Design:-