

## **Experiment-3**

**Student Name** Sandeep kumar

**UID:** 23BCS11489

**Branch:** CSE

**Section/Group:** KRG\_3B

**Semester:** 6th

**Date of Performance:** 28/01/26

**Subject Name:** System Design

**Subject Code:** 23CSH-314

**1. AIM :** To design a Social Media Platform similar to Facebook / Instagram

Description: A Social Media platform is a platform which allows users to share photos, videos, and text with their friends and followers. To design a scalable and highly available **Social Media System** where users can register, create posts, follow others, and interact with content through likes and comments.

### **2. Objectives**

- To understand the working of a large-scale social media system
- To design functional and non-functional requirements
- To identify core system entities
- To design API endpoints for communication
- To ensure scalability, availability, and low latency

### **3. Tools Required**

- System Design Tools (Draw.io, Lucidchart, etc.)
- Programming Language (Java / Python / JavaScript)
- Database (MySQL / MongoDB)
- API Testing Tool (Postman)
- Web Browser
- IDE (VS Code, IntelliJ, etc.)

## **4. SYSTEM DESIGN / SYSTEM SPECIFICATION**

### **4.1 Functional Requirements**

1. User should be able to register and login.
2. User should be able to create posts (text / image / video).
3. Users should be able to follow each other / send friend requests.
4. Users should be able to like and comment on posts.
5. Users should be able to view feed from people they follow.

### **4.2 Non-Functional Requirements**

#### **1. Scalability:**

- System should support 500 million DAU.

## 2. Consistency & Availability (CAP Theorem):

- This system needs to be highly available first, then consistent.
- Reason: If the system is not operational, it becomes useless.
- Example:
  - If Instagram is down for 1 hour → huge issue.
  - But if a post takes 500ms to reach followers → acceptable.
- Hence:  
Availability >>> Consistency

### 3. Latency:

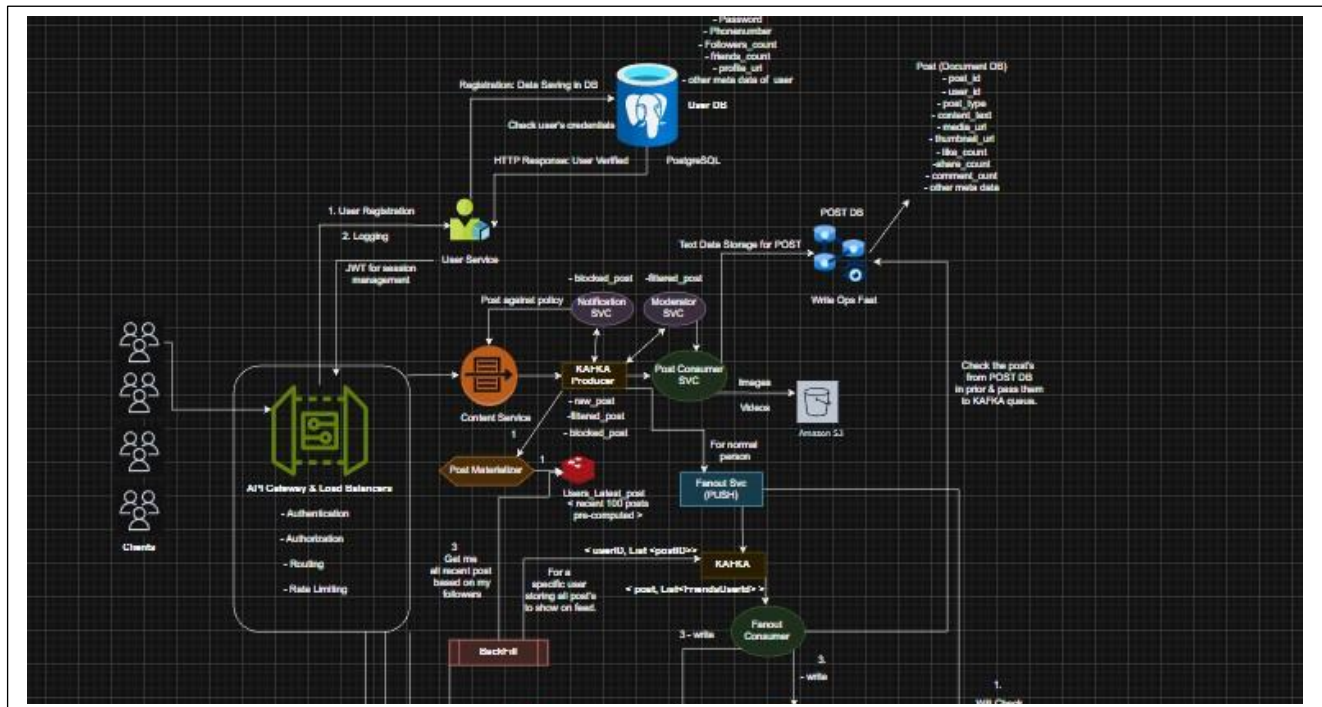
- **Uploading / publishing post should take  $\leq 500$  ms.**

### 4.3 Core Entities of the System

- User
- Post
- Comment
- Like
- Follow / Friend Request
- Feed
- Media (Image/Video)

### 5. HLD(High Level Design):

We have to follow a distributed / micro-services approach not the monolithic one



## **7. Learning Outcomes**

- Understand the design and architecture of a scalable E-commerce platform.
- Gain hands-on experience with Apache Kafka for real-time data streaming.
- Learn to implement fast and efficient search using Elasticsearch.
- Understand Change Data Capture (CDC) pipelines for real-time data synchronization.
- Develop skills in integrating distributed systems for high availability and scalability.