DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
Discover. Learn. Empower.

Experiment No:3

Student Name: Sumanyu Singh
UID:23BCS10801
Branch: BE CSE
Section/Group:KRG-3B
Semester: 6th
Date of Performance:28/01/2026
Subject Name: System Design
SubjectCode:23CSH-314

Aim: To design a Social Media Platform similar to Facebook / Instagram .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.

Objective:

- To study how large social networking applications operate at scale.
- To identify the essential system features required for user interaction.
- To define both functional and quality requirements of the system.
- To design communication interfaces using APIs.
- To ensure the system remains responsive, reliable, and expandable for millions of users.

Tools and Technologies Used:

- Diagram designing tools such as Draw.io or Lucidchart for architecture creation.
- Programming languages like Python, JavaScript, or Java for implementation.
- Databases such as MongoDB or MySQL for storing user and post information.
- API testing platforms like Postman.
- Web browsers and development environments such as VS Code.

Functional Requirements :

- New users must be able to create accounts and securely log in.
- Users should publish different types of posts including text, images, or videos.
- The platform should allow connections between users through follow or friend features.
- Interaction mechanisms such as likes and comments must be supported.
- Each user should receive a feed containing updates from connected profiles.

Non - Functional Requirements :

- **Scalability**
  The system must be capable of handling extremely large traffic volumes, potentially supporting hundreds of millions of daily active users without performance degradation.
- **Availability and Consistency (CAP Perspective)**
  The design prioritizes availability over strict consistency. A temporary delay in updating posts across followers is acceptable, but service downtime is not. Continuous access to the platform is essential for user satisfaction.
- **Latency**
  Operations like uploading or publishing content should complete quickly, ideally within half a second to maintain a smooth user experience.

Core-Entites of the System:

The important data components involved in the system include:

- User profiles
- Posts created by users
- Comments added to posts
- Likes or reactions
- Follow or friend relationships
- Personalized feed generation
- Media storage for images and videos

API Designing (Low Level Design – LLD)

1. POST API – User Registration

https://localhost/users/register

- Allows a new user to create an account on the platform.
- Accepts details such as username, email, and password.
- Performs validation before storing user information.

2. POST API – User Login

https://localhost/users/login

- Authenticates users using login credentials.
- Generates session tokens or authentication tokens for secure access.

3. POST API – Create New Post

https://localhost/posts/create

- Enables users to upload text, image, or video content.
- Media files are stored in media storage services.
- Post metadata is saved in the database.

4. GET API – View User Feed

https://localhost/feed/get_feed?user_id={user_id}

- Retrieves personalized feed content based on followed users.
- Pagination and caching are applied to improve performance.
- Displays recent posts in sorted order.

5. POST API – Follow or Send Friend Request
https://localhost/users/follow

- Allows users to follow another profile or send a connection request.
- Updates follower and following relationship records.

6. POST API – Like a Post
https://localhost/posts/{post_id}/like

- Adds or removes a like reaction on a post.
- Updates engagement counters in real time.

7. POST API – Add Comment
https://localhost/posts/{post_id}/comment

- Enables users to write comments on posts.
- Stores comment data linked with user and post IDs.

8. GET API – View Post Details
https://localhost/posts/{post_id}

- Fetches complete information about a specific post.
- Includes media, captions, comments, and like count.

9. DELETE API – Remove Post
https://localhost/posts/delete/{post_id}

- Allows users to delete their own posts.
- Removes media references and updates feeds.

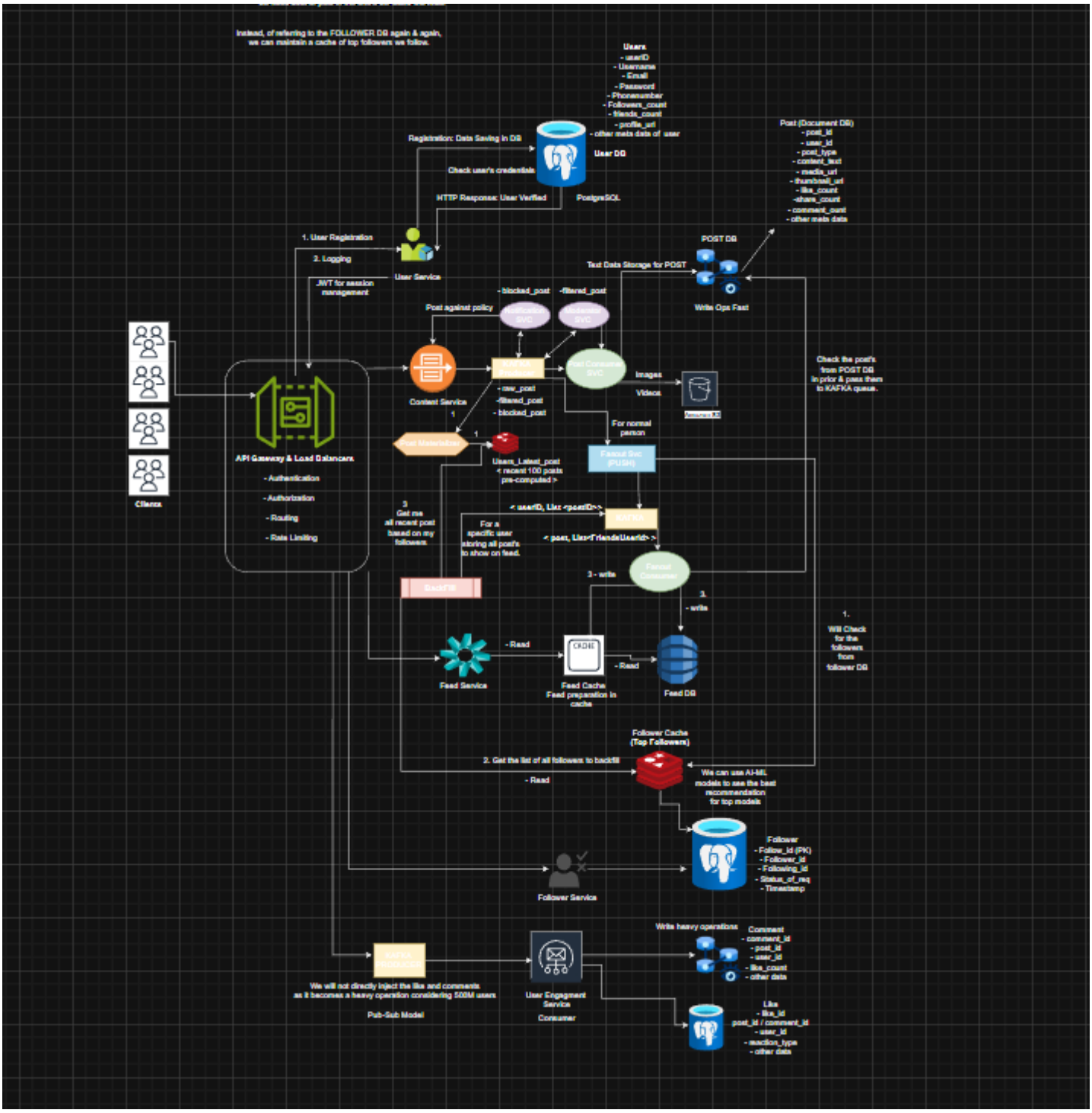
10. GET API – Notification Fetch
https://localhost/notifications?user_id={user_id}

- Retrieves notifications related to likes, comments, and new followers.
- Supports real-time updates.

11. PUT API – Update User Profile
https://localhost/users/update_profile

- Used to modify profile information such as bio or profile picture.

HLD(High level Diagram) -> FINAL :



Learning outcomes:

- Understanding the architecture of large-scale online platforms.
- Learning how distributed services improve performance and reliability.
- Gaining knowledge about real-time data streaming using messaging systems.
- Understanding synchronization between databases using data capture pipelines.
- Developing skills in designing scalable and fault-tolerant software systems.