Experiment 3

Student Name: Umang Kaushik **UID: 23BCS10712**
Branch: BE CSE **Section/Group: KRG-1-B**
Semester: 6th **Date of Performance:26th Jan**
Subject Name: SD **Subject Code: 23CSH-314**

1. **Aim:** To design a social-media application

2. **Objectives:**

   - Facilitate Connectivity: Allow users to find and follow others.
   - Content Management: Provide tools to create, share, and interact with multimedia content.
   - Real-time Interaction: Enable instant notifications and messaging.
   - Data Security: Ensure the protection of user privacy and personal information.

3. **Procedure:**
   - Requirement Analysis: Define target audience and core features.
   - UI/UX Design: Create wireframes and high-fidelity prototypes focusing on intuitive navigation
   - Architecture Design: Plan the database schema (SQL/NoSQL) and system backend.
   - Development: Build the frontend (React/Flutter) and backend (Node.js/Python) APIs.
   - Testing: Conduct unit, integration, and user acceptance testing (UAT).
   - Deployment: Host the application on cloud services (AWS/Azure) with CI/CD pipelines.

4. **Functional Requirements:**
   - User Authentication: Secure sign-up, login, and profile management.

- News Feed: An algorithm-based or chronological feed of posts from followed accounts.
- Interactions: Capabilities to like, comment on, and share posts.
- Direct Messaging: Private one-on-one or group chat functionality.
- Search: Ability to find users, hashtags, or specific topics.

5. Non-Functional Requirements:

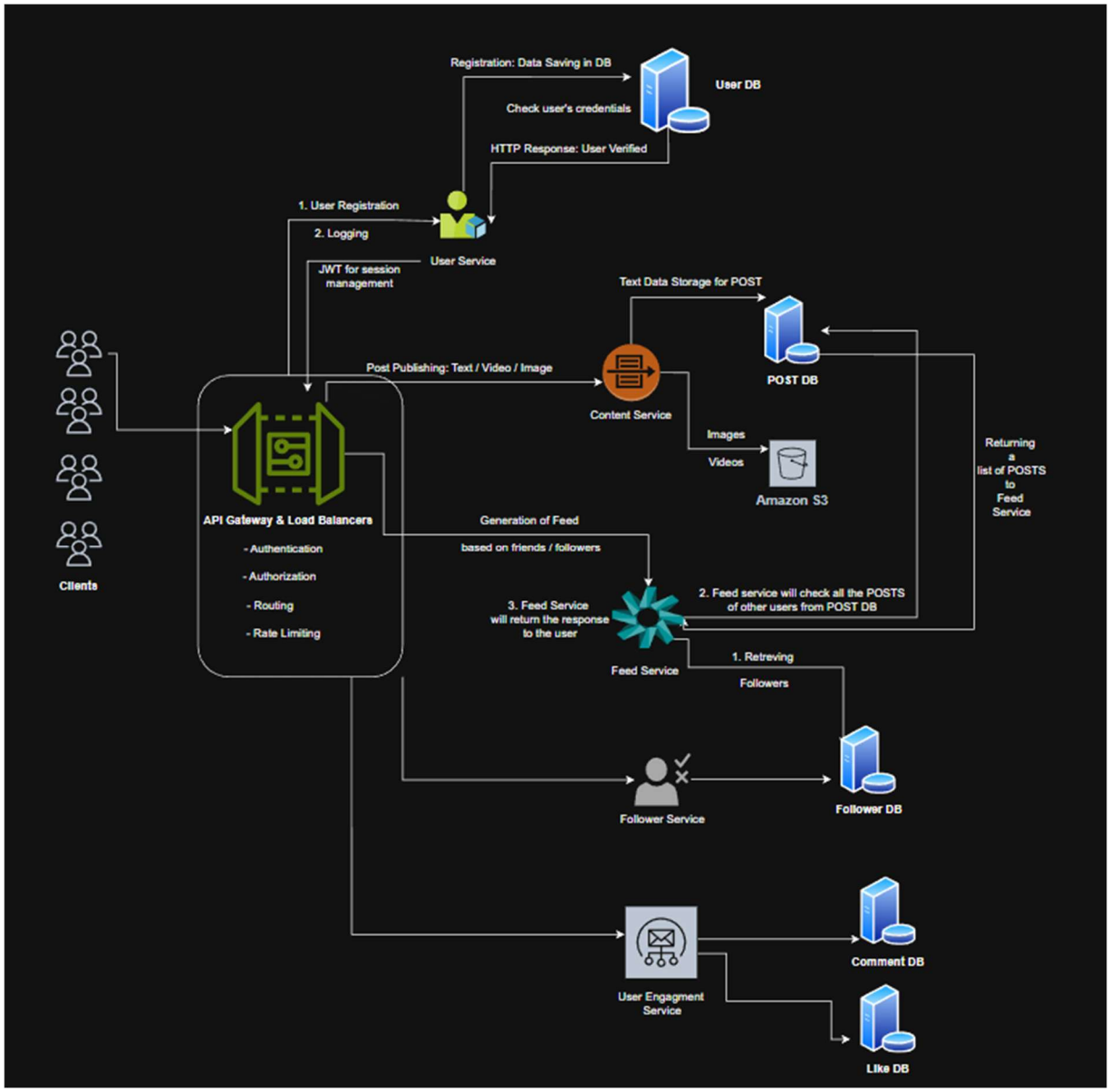
- Scalability: The system must handle a growing number of users and concurrent requests.
- Latency: Post updates and messages should reflect in under 2 seconds.
- Availability: Target 99.9% uptime to ensure the service is always accessible.
- Security: Use of End-to-End Encryption (E2EE) for messages and OAuth for authentication.

6. Outcome:

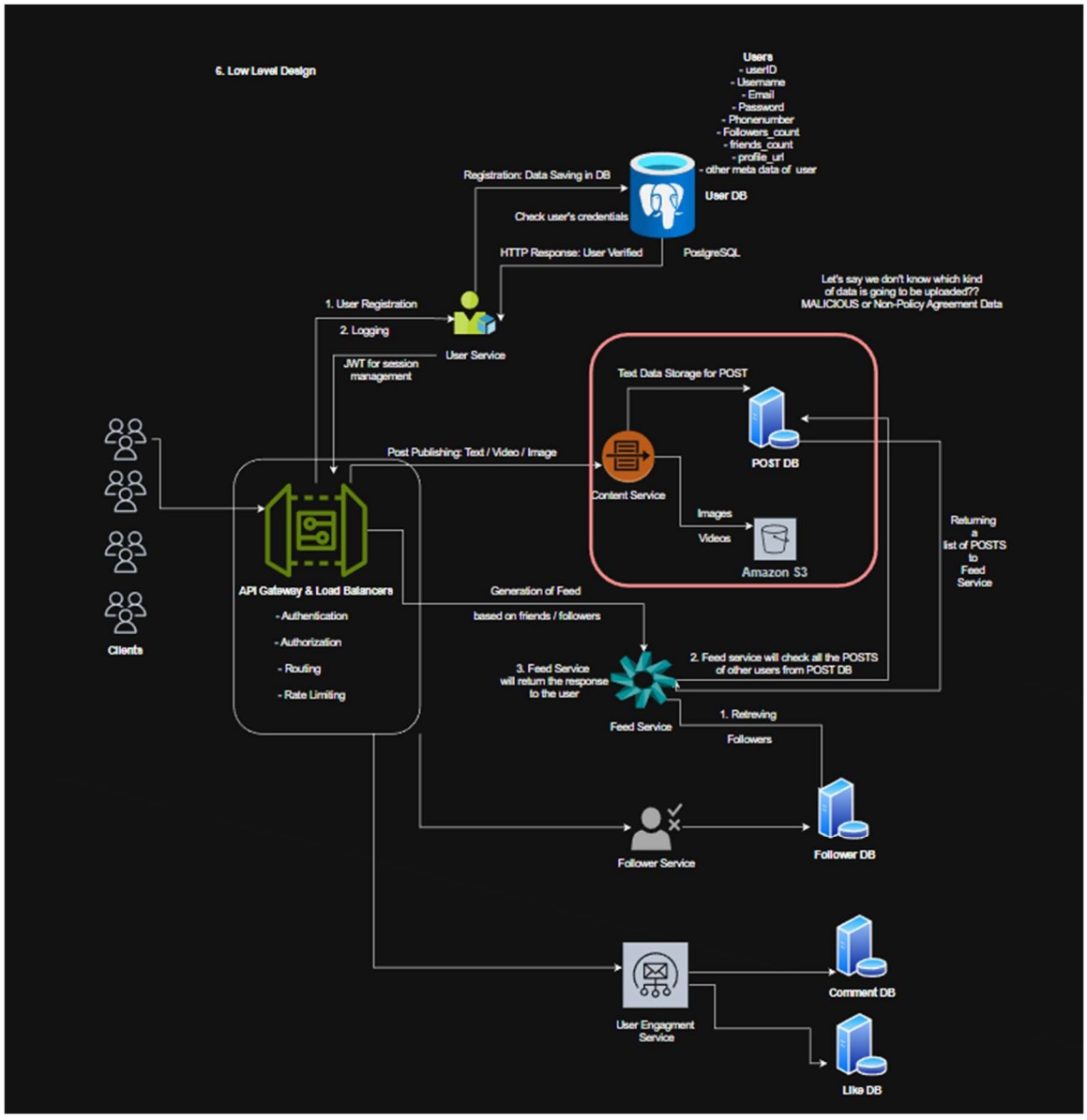
The result is a fully functional social media prototype capable of supporting high-traffic interactions, providing a seamless user experience, and maintaining a secure environment for digital socialization.

7. Required System Design:

HLD:


Registration: Data Saving in DB
User DB
Check user's credentials
HTTP Response: User Verified
1. User Registration
2. Logging
User Service
JWT for session management
Text Data Storage for POST
POST DB
Post Publishing: Text / Video / Image
Content Service
Images
Videos
Amazon S3
Returning a list of POSTS to Feed Service
Clients
API Gateway & Load Balancers
- Authentication
- Authorization
- Routing
- Rate Limiting
Generation of Feed
based on friends / followers
2. Feed service will check all the POSTS of other users from POST DB
3. Feed Service will return the response to the user
Feed Service
1. Retreving
Followers
Follower Service
Follower DB
Comment DB
User Engagment Service
Like DB

LLD


6. Low Level Design
Users
- userID
- Username
- Email
- Password
- Phonenumber
- Followers_count
- friends_count
- profile_url
- other meta data of user
User DB
PostgreSQL
Registration: Data Saving in DB
Check user's credentials
HTTP Response: User Verified
1. User Registration
2. Logging
JWT for session management
User Service
Let's say we don't know which kind of data is going to be uploaded??
MALICIOUS or Non-Policy Agreement Data
Text Data Storage for POST
POST DB
Post Publishing: Text / Video / Image
Content Service
Images
Videos
Amazon S3
Returning a list of POSTS to Feed Service
Clients
API Gateway & Load Balancers
- Authentication
- Authorization
- Routing
- Rate Limiting
Generation of Feed
based on friends / followers
3. Feed Service will return the response to the user
2. Feed service will check all the POSTS of other users from POST DB
Feed Service
1. Retreving
Followers
Follower Service
Follower DB
User Engagment Service
Comment DB
Like DB