



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

Experiment-3

Student Name:Astik Joshi

Branch: CSE

Semester: 6th

Subject Name: System Design

UID: 23BCS10627

Section/Group: KRG-3B

Date of Performance: 28/01/26

Subject Code: 23CSH-314

1. AIM : To design a social media platform similar to Facebook or Instagram

2. Objective:

- To architect and build a scalable social media platform enabling users to register, connect, and engage through posts.
- To offer essential social networking functionalities, including posting, following, liking, commenting, and feed generation.
- To guarantee high availability and minimal latency, ensuring the platform remains accessible to a large user base at all times.
- To design RESTful APIs that manage user onboarding, content operations, and user interactions efficiently.

3. Tools Required:

- **Frontend:** HTML, CSS, JavaScript, React.js
- **Backend:** Node.js, Express.js
- **Database:** MongoDB
- **API Testing:** Postman
- **Authentication:** JWT
- **Media Storage:** Cloudinary / AWS S3
- **Version Control:** Git, GitHub
- **Deployment:** AWS / Render / Vercel

4. SYSTEM DESIGN / SYSTEM SPECIFICATION:

4.1. Functional Requirements:

- Client should be able to register and login to the application.
- Client should be able to create post (text / image / videos)
- Client should be able to follow each other (or send friend requests).
- Client should be able to like or comment on the post.
- Client should be able to view the feed of post from users they follow.



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

4.2. Non-functional Requirements:

- Scalability: 500M daily active users.
- Consistency & Availability:
For social media applications, **availability is more important than consistency**.

Availability is prioritized over Consistency.

- Latency: (Uploading speed of publishing post): 500ms to upload post

4.3. Core-Entites of the System:

1. User
2. Posts
3. Followers
4. Like and Comment
5. Feed

4.4. API Endpoints Creation:

A. User On-boarding API's

1. **User Registration:** POST API CALL: POST / api / users / register_user
2. **User Login:** POST API CALL: POST / api / users / login
3. **User Data Display:** GET API CALL: GET / api / users / {user_id} / profile
4. **User Data Update:** PUT API CALL: PUT / api / users / {user_id} / profile

B. User Post's

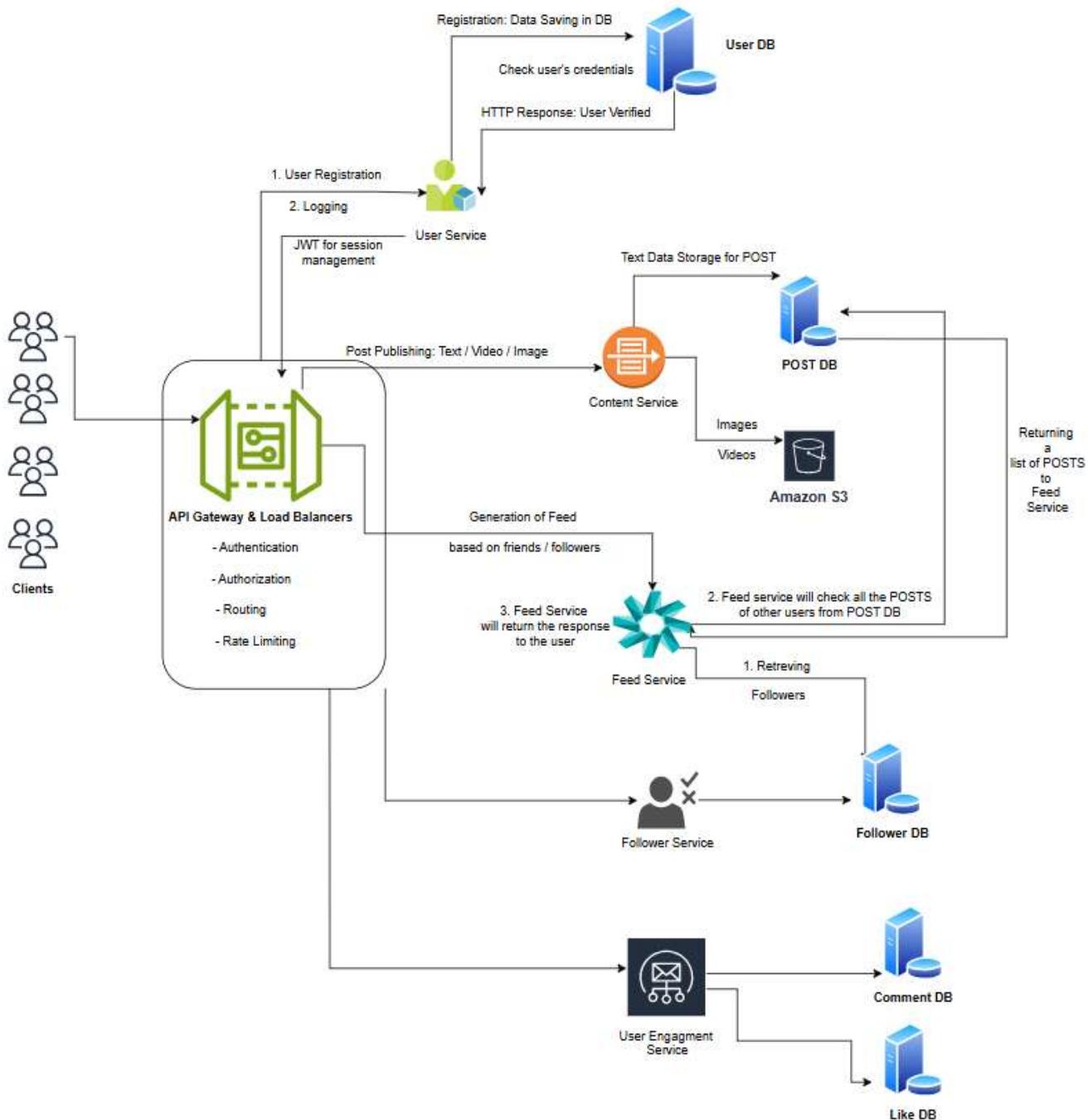
1. POST / api / user_id / posts
2. GET / api / posts / {post_id}
3. PUT / api / posts / {post_id}
4. DELETE / api / posts / {post_id}
5. GET / api / posts / feed / limit = {limit} & offset = {offset} : PAGINATION
6. GET / api / users / {user_id} / posts: PAGINATION

C. User Interactions

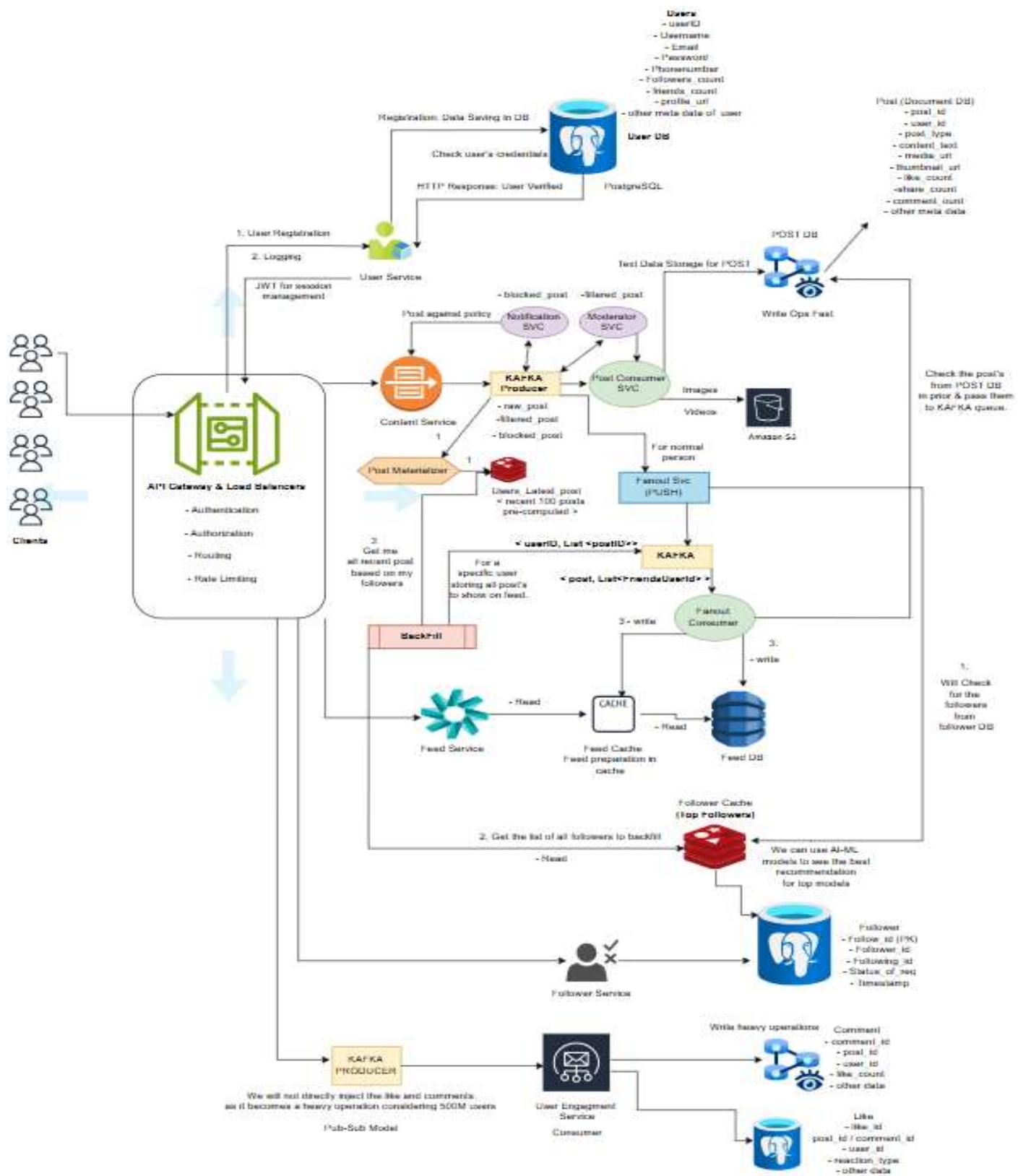
1. POST / api / posts / {post_id} / like
2. DELETE / api / posts / {post_id} / unlike
3. POST / api / posts / {post_id} / comments
4. GET / api / posts / {post_id} / comments
5. PUT / api / {post_id}/ comments / {comment_id}
6. DELETE / api / {post_id}/ comments/ {comment_id}
7. POST / api / users / {user_id} / follow
8. DELETE / api / users / {user_id} / unfollow

5. HLD(High Level Design):

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



6. LLD(Low Level Design)





DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

7. Learning Outcomes

- Analyze and specify both functional and non-functional requirements of a social media platform.
- Identify and model key entities such as Users, Posts, Followers, Likes, Comments, and Feeds.
- Develop RESTful API endpoints to manage users, handle posts, and support social interactions.
- Incorporate principles of scalability, low latency, and pagination into the system design.