1. Approaches to Implementing Rate Limiting

Rate limiting can be implemented using various strategies, but two common approaches are:

A. Token Bucket Algorithm

- **Each user has a "bucket" that fills with tokens at a fixed rate (e.g., 5 tokens per second).**
- **Each request consumes a token; if the bucket is empty, the request is denied.**
- **Pros:**
    - **Allows bursts of requests while maintaining a controlled average rate.**
    - **Efficient with low memory usage.**
- **Cons:**
    - **Slightly complex implementation with token refilling logic.**

B. Sliding Window Log

- **Stores timestamps of requests in a list or queue.**
- **When a new request arrives, old timestamps outside the sliding window (1 second) are removed.**
- **If the number of remaining timestamps exceeds the limit, the request is denied.**
- **Pros:**
    - **Precise control over request distribution.**
    - **Handles sudden bursts better than fixed window methods.**

- **Cons:**
    - **Higher memory consumption due to storing timestamps**

Trade-offs Between the Approaches

Approach	Pros	Cons
Token Bucket	**Efficient, allows bursts of requests, predictable refill rate**	**More complex logic for token refill**
Sliding Window Log	**Precise rate limiting, prevents bursts beyond threshold**	**Higher memory usage due to storing timestamps**

For high-performance banking systems, Token Bucket is often preferred due to its efficiency, while Sliding Window Log provides precise control in fraud detection scenarios.