Features in ‘consumer.js’ :-  
  
1) Summary:- Connects to RabbitMQ, , listens for messages, buffers them, and inserts them into MongoDB.  
2) Uses batches of 5 messages at once with max time 5 mins before processing a batch.  
3) Temporarily stores messages of RabbitMQ in buffers before inserting them in MongoDB.  
4) Consists of an async function ‘consumeMessages’ that stores messages received from RabbitMQ.  
5) We are using queue (FIFO) in processing the messages.  
6) Has error handling and retrying feature that applies 5 seconds into channel closure.  
7) Messages survive even after the reconnection of RabbitMQ showcasing durability.  
8) Has logs that tracks real-time system behaviour.   
9) Prevents overloading by making small batches.  
10) Prevents crashes by handling errors safely using ‘try’ block.  
  
Features that can improve ‘consumer.js’:-  
  
1) Parallel processing by using multiple consumers.  
2) Alert System by sending notification when message processing fails too many times.  
3) If a message fails multiple times, move it to a "dead-letter" queue instead of retrying infinitely.  
  
Features in ‘generator.js’ :-  
  
1) Protects sensitive user data by sharing only a part of token.  
2) Summary:- Generates structured logs for incoming requests, successful operations, and errors. Ensure sensitive data is masked.  
3) Logs processing time.  
4) Tracks and monitors the APIs.  
  
Features that can improve ‘generator.js’:-  
  
1) Writing logs to a database will keep it stored permanently.  
2) Can use IP for tracking user locations.  
  
Features in ‘logger.js’ :-  
  
1) Uses AMQP for logging instead of files.  
2) Ensures logs are not suppressed ‘ silent : false, ’.   
3) Has unique ‘correlationID’ to track every log across microservices.  
4) Has timestamps on each log.  
5) In the end, has a feature to send logs to RabbitMQ using AMQP.  
6) Can also use file based logging in case it is needed.  
  
Features that can improve ‘logger.js’:-  
1) Use file logging as RabbitMQ can fail sometimes. 2) We can also log response time.