

Complex Node.js Assignment – Async File Processing + Queue + Webhook

Objective

Build a backend service in Node.js that supports large CSV file uploads, processes them asynchronously, stores the results in a database, and notifies an external system via a webhook with retry logic.

Assignment Requirements

1. API Endpoints

A. POST /upload

Create an API endpoint that:

1. Accepts a CSV file upload (10k–50k rows).
2. Validates:
 - File type must be CSV
 - File size \leq 10 MB
3. Stores file metadata in the database with:
 - fileId (UUID)
 - status (PENDING)
 - uploadedAt
4. Enqueues a background job to process the file asynchronously.

2. Asynchronous Processing

Implement a background worker using any queue system (BullMQ, RabbitMQ, SQS, or a custom queue) that:

1. Reads the CSV using streaming (not loading entire file into memory).
2. Validates each row (e.g., email format, age > 0).
3. Stores:
 - Valid rows in a valid_records table/collection
 - Invalid rows with error_reason in invalid_records
4. Updates processing statistics.
5. Marks status=COMPLETED and records completedAt timestamp.

3. Webhook Notification

After processing completes:

1. Send POST request to webhook URL with:
 - fileId, rowsProcessed, validRows, invalidRows, completedAt
2. Implement retry logic:
 - Retry delays: 1 min \rightarrow 5 min \rightarrow 30 min
 - Stop after 5 attempts
 - Set webhookStatus="FAILED" if all retries fail

4. Status API

GET /status/:fileId returns:

fileId, status, validRows, invalidRows, webhookStatus, uploadedAt, completedAt

5. Architecture Requirements

Must include:

- /routes
- /controllers
- /services
- /workers
- /queues
- /models

Plus environment config, error handling, logging, and comments.

