Software Development Engineer (SDE-1) Backend Assignment

Name: Abhiram Masna

Email: masnaabhiram99@gmail.com

GitHub Repository: https://github.com/AbhiramMasna/SDE_BACKEND_ASSIGNMENT

----

1. Introduction

This document outlines the solution for the backend assignment, which involves processing image data from CSV files asynchronously.

The system ensures efficient image processing, storage, and retrieval using FastAPI, Celery, PostgreSQL, and Redis.

----

2. System Overview

Objective:

- Accept a CSV file containing product names and image URLs.
- Validate the CSV format.
- Asynchronously process and compress images to 50% of their original quality.
- Store the processed image data in a database.
- Provide API endpoints to check processing status.
- (Bonus) Implement a webhook to notify when processing is complete.

Tech Stack:

- Backend: Python (FastAPI)
- Database: PostgreSQL

- Image Processing: Pillow (PIL)

- Async Processing: Celery with Redis

- Storage: Local or Cloud (AWS S3/Cloudinary)

----

3. System Architecture & Low-Level Design (LLD)

Components:

- Upload API: Accepts CSV and returns a request ID.

- Asynchronous Worker (Celery): Downloads, compresses images, and stores results.

- Database: Stores request IDs, image URLs, and statuses.

- Status API: Checks processing status.

- Webhook (Bonus): Sends a notification when processing is complete.

----

4. Database Schema

Table: requests

request_id	status	created_at	updated_at
UUID	pending	timestamp	timestamp

Table: images

id	request_id	product_name	input_url	output_url	status
1	UUID	SKU1	img1.jpg	img1_compressed.jpg	done

----

5. API Documentation

1. Upload API

- Endpoint: POST /upload

- Description: Uploads CSV, validates it, and returns a request ID.

- Response: { "request_id": "12345" }

2. Status API

- Endpoint: GET /status/{request_id}

- Description: Returns processing status.

- Response: { "request_id": "12345", "status": "completed" }

3. Webhook (Bonus)

- Endpoint: POST /webhook

- Description: Triggered after all images are processed.

----

6. Conclusion

This project successfully demonstrates how to asynchronously process image data from a CSV file using FastAPI, Celery, PostgreSQL, and Redis.

Further improvements could include cloud storage integration and Docker-based deployment.