Image Processing System Documentation

1. Low-Level Design (LLD)

System Components

A. Web Server (Flask Application)

- **Purpose: Handles HTTP requests, file uploads, and status checks**
- **Key Components:**
  - ○ **Upload endpoint (/upload)**
  - ○ **Status endpoint (/status/<request_id>)**
  - ○ **Download endpoint (/download/<request_id>)**

B. Image Processor

- **Purpose: Compresses images and generates processed URLs**
- **Functions:**
  - ○ **Downloads images from URLs**
  - ○ **Compresses images to 50% quality**
  - ○ **Saves processed images**
  - ○ **Generates output URLs**

C.CSV Handler

- **Purpose: Processes input CSV and generates output CSV**
- **Function:**
  - ○ **Validates CSV format**
  - ○ **Extracts product and image information**
  - ○ **Generates output CSV with processed image URLs**

D. Database (MongoDB)

- **Purpose: Stores request status and product information**
- **Collections:**
  - ○ **processing_requests: Tracks overall request status**
  - ○ **products: Stores product and image URL information**

E. Webhook Notifier

- **Purpose: Sends processing completion notifications**
- **Function:**
  - ○ **Accepts webhook URL during upload**
  - ○ **Sends status updates to specified URL**
  - ○ **Includes output CSV download link**

Data Flow

1. ### File Upload Flow:

   Client Upload API ⟶ CSV Validation ⟶ Request ID Generation ⟶ Background Processing Start ⟶ Response to Client

2. ### Processing Flow:

   Background Worker ⟶ Read CSV ⟶ Process Each Row ⟶ Download & Compress Images ⟶ Store Results ⟶ Generate Output CSV ⟶ Trigger Webhook

3. ### Status Check Flow:

   Client -> Status API -> Database Check -> Return Current Status

API Documentation

1.Upload API

Endpoint:/upload
Method: POST
Content-Type:multipart/form-data

Request Parameters:

```
{
    "file": "CSV File (required)",
    "webhook_url": "String (optional)"
}
```

Success Response (200 OK):

```
{
    "request_id": "uuid-string",
     "status": "processing"
}
```

Error Response (400 Bad Request):

```
{
    "error": "Error message"
}
```

2. Status API

Endpoint: `/status/<request_id>` **Method: GET**

Success Response (200 OK):

```
{
   "request_id": "uuid-string",
   "status": "processing|completed|failed",
   "output_csv_url": "/download/uuid-string" // Only when completed
}
```

Error Response (404 Not Found):

```
{
 "error": "Request ID not found"
 }
```

3. Download API

Endpoint: `/download/<request_id>` **Method: GET**

Success Response: CSV file download Error Response (404 Not Found):

```
{
   "error": "Output CSV not found"
}
```

3. Database Schema

Collection: processing_requests

```
{
   "request_id": "String (UUID)",
   "status": "String (processing|completed|failed)",
   "created_at": "DateTime",
   "completed_at": "DateTime",
   "filename": "String",
   "webhook_url": "String",
   "output_csv_path": "String"
}
```