1. Upload
2. Status
3. ProcessImage
4. CompressImage
5. webHook

1. Upload

Validate and Store Request

- Checks if a CSV file is uploaded and validates the file type.
- Creates a new entry in the `requests` table with status **"PENDING"**.

Parse CSV File

- Reads the file and extracts product details (serial number, product name, input image URLs).
- Stores each product in the `products` table, linking it to the request.

Process Images

- Iterates over the extracted products and processes their input images.
- Generates output image URLs and updates the `products` table.

Trigger Webhook (After Processing)

- Updates request status to **"COMPLETED"** once all images are processed.
- Sends a webhook notification with the request ID and completion status.

2. Status

Fetch Request Status
- Retrieves the processing status (`PENDING` or `COMPLETED`) of a CSV upload request based on `requestId`.

Return Processed Data
- If processing is completed, it provides the processed product details along with their output image URLs.

3. ProcessImage

Validates Image URLs
- Iterates through `inputImageUrls` and skips invalid URLs. Sends imageUrl to compressImage function and gets a imageBuffer in return.

Processes Each Image
- (Currently commented) Intended to **compress images** for optimization.

Uploads to Storage
- (Currently commented) Once blob storage is available, compressed images will be uploaded.

4. CompressImage

Fetches Image Data
• Downloads the image from the given URL as a binary buffer using Axios.

Compresses Image
• Uses Sharp to convert the image to JPEG format with **50% quality** and returns the compressed buffer.

5. WebHook

Sending Notification
• Triggers a webhook notification upon completion of CSV processing, sending the request ID and status update.