**1.API Documentation**

**1.1 Upload API**

* **Endpoint:** /upload/
* **Method:** POST
* **Description:** Accepts a CSV file containing product data (Serial Number, Product Name, Input Image URLs) for processing. The file is validated, and a unique request ID is returned to track the processing status.
* **Request Parameters:**
  + **file** (type: form-data, required): The CSV file containing the product and image data.
* **Response:**
  + **Success:**
    - **Status Code:** 200 OK
    - **Body:**

**1.2 Status API**

* **Endpoint:** /status/{request\_id}
* **Method:** GET
* **Description:** Retrieves the status of the image processing for the given request\_id.
* **Path Parameters:**
  + **request\_id** (string, required): The unique ID assigned to the CSV upload request.
* **Response:**
  + **Success:**
    - **Status Code:** 200 OK
    - **Body:**

**2. Database Schema**

**2.1 Database Structure**

* **Database:** MongoDB
* **Collection:** products

**2.2 Collection Fields**

* **id** (string): Unique identifier for the request.
* **serial\_number** (integer): Serial number for the product.
* **product\_name** (string): Name of the product.
* **input\_image\_urls** (list of strings): URLs of the original images.
* **output\_image\_urls** (list of strings): URLs of the processed images.
* **status** (string): Status of the processing (e.g., "Processing", "Completed").

**3. Design Structure**

**3.1 System Components**

* **CSV Upload & Validation:**
  + The system accepts a CSV file with product details and image URLs.
  + Validates the structure and content of the CSV.
* **Asynchronous Image Processing:**
  + Compresses images to 50% of their original quality asynchronously.
  + Updates the database with the compressed image URLs.
* **Database Interaction:**
  + Stores product data, including input and output image URLs, in MongoDB.
  + Tracks the processing status for each request.

**3.2 Workflow**

1. **CSV Upload:**
   * User uploads a CSV file via the /upload/ API endpoint.
   * The server validates the CSV format and assigns a request\_id.
2. **Image Processing:**
   * Images listed in the CSV file are processed asynchronously.
   * Compressed images are saved, and URLs are updated in the database.
3. **Status Check:**
   * Users can check the status of their request using the /status/{request\_id} API endpoint.
4. **Results Retrieval:**
   * Optionally, users can retrieve the processed image URLs using the /results/{request\_id} endpoint.