# Technical Design Document

## 1. Introduction

- **Project Title**: Image Processing System
- **Description**: This document provides a detailed technical design for an image processing system. The system processes image data from CSV files, compresses images, and stores the processed image data in a database. The system also includes asynchronous processing, status tracking, and a webhook flow for notification upon completion.

## 2. System Overview

### Components and Their Functions

- **Image Processing Service Interaction**: Integrates with the asynchronous image processing service to compress images.
- **Webhook Handling**: Processes callbacks from the image processing service to update the image processing status.
- **Database Interaction**: Stores product data and tracks the status of each processing request.
- **API Endpoints**:
- **Upload API**: Accepts CSV files and returns a unique request ID.
- **Status API**: Allows users to check the processing status using the request ID.

## 3. System Architecture Diagram

- Link to the Document: - https://drive.google.com/file/d/1WCH8EwPdUllWw1k9dxDxEv89wmonkIdR/view?usp=drive_link

## 4. Database Schema

- **Processing Request**
  - id: UUID, Primary key
  - created_at: DateTime, Timestamp of request creation
  - status: Char, Status of the processing request
- **Product**
  - id: Integer, Primary key
  - request: ForeignKey, Reference to ProcessingRequest
  - serial_number: Integer, Serial number of the product
  - product_name: Char, Name of the product

    - input_image_urls: Text, Comma-separated input image URLs
    - output_image_urls: Text, Comma-separated output image URLs (after processing)

### 5. API Documentation

- **Upload API**
    - **Endpoint**: /api/upload/
    - **Method**: POST
    - **Request**: file (CSV file containing product data and image URLs)
    - **Response**: request_id (Unique request ID for tracking status)

- **Status API**
    - **Endpoint**: /api/status/<request_id>/
    - **Method**: GET
    - **Response**:
        - request_id: The request ID
        - status: Status of the processing request
        - products: List of products with their input and output image URLs

### 6. Asynchronous Workers Documentation

- **Worker Functions**
    - **Task**: process_images
        - **Description**: Processes images asynchronously, compresses them, and updates the database with output image URLs.
        - **Steps**:
            1. Fetch the ProcessingRequest by request_id.
            2. Retrieve associated Product entries.
            3. For each product, download and compress images.
            4. Save compressed images and update output_image_urls.
            5. Update the status of the ProcessingRequest to 'completed'.