Image Deduplication Assignment

Overview

Build a full-stack application that deduplicates images based on visual content similarity.

Requirements

API Integration

- Base Endpoint: `https://vibemyad.com/api/test-assignment`
- Available brand datasets (use `brand_key` parameter):
  - `https://vibemyad.com/api/test-assignment?brand_key=nike`
  - `https://vibemyad.com/api/test-assignment?brand_key=levis-1`
  - `https://vibemyad.com/api/test-assignment?brand_key=gonoise`
- Fetch the array of images returned by the API
- You can test with one or multiple brand datasets

Core Functionality

- Usually ads are run with the same image in different aspect ratios. These look visually the same to humans.
- We need to group these visually similar images together and show them to the user
- Group duplicate images together (even if they have different sizes, aspect ratios, or formats)
- Display the results in a user interface

Technical Constraints

- Choose your own tech stack (backend + frontend)
- You may use AI services/APIs for image analysis if desired
- Solution should handle at least 100+ images efficiently

Deliverables

1. **Working Application**
   - Backend service that processes images
   - Frontend interface to display results

2. **Documentation**
    - README with setup instructions
    - Explanation of your approach and algorithm choice
    - Trade-offs and decisions made
3. **Code Repository**
    - Clean, organized code structure
    - Include any configuration files needed

Evaluation Criteria

Primary Focus:

- **Deduplication Quality (50%)** - Accuracy in identifying true duplicates while avoiding false positives
- **Approach & Algorithm (30%)** - Choice of similarity detection method, justification, and understanding of trade-offs
- **Engineering Quality (20%)** - Code structure, error handling, scalability considerations

UI is just for demonstration - it can be minimal/basic as long as results are visible

Questions to Consider

- How do you determine if two images are "the same"?
- What happens if images are slightly different sizes or quality?
- How do you optimize for performance with many images?
- How do you handle errors (API failures, corrupted images, etc.)?

Good luck! 🚀