Flickd AI Hackathon — Full Competition Document

Overview

Flickd is reimagining how Gen Z shops — through scroll-native, video-first, vibe-led discovery. Our shopping journey doesn't start at a search bar — it starts with a Reel. We want to automate the tagging of products and the classification of fashion "vibes" from these short videos. This hackathon invites AI/ML engineers to build the backbone of this intelligent system.

This document provides a comprehensive breakdown of the 7-day hackathon competition to hire Flickd's first ML engineer. It covers:

- Objective and context
- Scope and deliverables
- Input datasets
- Evaluation metrics
- Stack and tools
- Timeline and daily plan
- Output format and submission expectations
- Hiring process and reward
- Terms and FAQ

Objective

Build a fully working MVP (Minimum Viable Product) of Flickd's "Smart Tagging & Vibe Classification Engine." The engine should:

1. Extract frames from videos
2. Use a pretrained object detection model (YOLOv8) to identify fashion items
3. Match detected items to a small catalog of products using image embeddings (CLIP)
4. Analyze captions or transcripts to classify the video's vibe using NLP
5. Output structured data via API or JSON

You are not required to build the UI — only the backend ML logic and return outputs.

What You'll Build

1. Object Detection (YOLO)

Use YOLOv8 to detect fashion items from video keyframes. These should include:

- Tops, bottoms, dresses, jackets
- Accessories like earrings, bags, shoes

The model should return:

- Class name
- Bounding box (x, y, w, h)
- Confidence score
- Frame number

2. Product Matching (CLIP + FAISS)

Each detected item must be compared against a fixed product catalog (Shopify image URLs) to identify the best match. This step should:

- Crop the detected object from the frame
- Generate its CLIP image embedding
- Match it using cosine similarity against a pre-embedded FAISS database of catalog items

Label results as:

- Exact Match (similarity > 0.9)
- Similar Match (0.75–0.9)
- No Match (< 0.75)

3. NLP-Based Vibe Classification

Given the caption + hashtags or an optional audio transcript, apply NLP (rule-based or transformer model) to classify the video into 1–3 vibes from this list:

- Coquette
- Clean Girl
- Cottagecore
- Streetcore
- Y2K
- Boho
- Party Glam

You can use spaCy, HuggingFace Transformers (e.g., DistilBERT), and your own logic to assign vibes.

4. Final Output

Your system should return a JSON output per video:

```
{
  "video_id": "abc123",
  "vibes": ["Coquette", "Evening"],
  "products": [
    {
      "type": "dress",
      "color": "black",
      "match_type": "similar",
      "matched_product_id": "prod_456",
      "confidence": 0.84
    }
  ]
}
```

Dataset

We will provide:

- 10 sample creator videos (~5–15s each)
- 1 CSV file with 200 product entries:
    - Product name
    - Shopify image URL (CDN)
    - Product ID
- A vibe taxonomy list (static list of supported vibes)
- A small caption set (if needed)

Tech Stack Recommendations

You can use:

- **YOLOv8** via `ultralytics` Python package
- **CLIP** (OpenAI or HuggingFace variant)
- **FAISS** for fast embedding matching
- **spaCy** or **HuggingFace Transformers** for NLP
- **Whisper** or AssemblyAI for audio-to-text (optional)
- **FastAPI** or Flask to expose your API (if applicable)

Use **Cursor IDE** for speed, AI suggestions, and auto-complete assistance. Not mandatory, but encouraged.

Evaluation Criteria

Metric	Weight	Description
Detection Accuracy	30%	Are bounding boxes accurate + sensible?
Match Quality	25%	Are product matches meaningful?
Vibe Classification	20%	Is the vibe relevant to the content?
Code Quality	15%	Clean logic, naming, modularity
Output Format & API	10%	JSON/API should be clearly structured

Bonus:

- Web demo or minimal frontend
- Logging and error handling
- Confidence scoring UI

Submission Format

- GitHub repo (clean structure)
  - `/frames`, `/models`, `/data`, `/api`
- README.md with setup + instructions
- Loom demo (5 mins max)
- Postman collection or endpoint (optional)
- Evaluation JSONs (per video)

Reward

- Top candidate gets hired as Flickd's ML/AI Engineer (contract-to-full-time)
- Immediate production work on live video tagging and recommendation pipelines
- Work directly with founders on Gen Z's next breakout product

---

Terms

- This is a hiring challenge, not a prize-based open competition.
- You must submit within 7 days from when you receive access.
- All your code is yours, unless you join Flickd, after which it will be productionized.

Apply Now

Send your name, GitHub, and LinkedIn to: **d@flickd.app** Subject line: “Flickd AI Hackathon”

We'll reply within 12 hours with access links, dataset, and onboarding instructions.

Feel free to shoot queries to d@flickd.app