

TagSafe AI

Trademark-Safe Keywords for E-Commerce Listings

Contact: Taha Disbudak (mtdi2024@mymail.pomona.edu) Pomona College Computer Science & Psychology

as manager and Zaan Saeed (zasaeed@hmc.edu) Harvey Mudd Computer Science & Physics as co-manager

Project Overview: This project builds an AI assistant that helps online sellers generate trademark-safe titles, tags, and descriptions for their product listings. Users upload a product photo or a short description; the system extracts candidate phrases, identifies the relevant Nice classes (e.g., Class 25 for apparel), checks against U.S. trademark records, flags risky terms, and proposes compliant, high-intent alternatives ready to paste into Etsy/Amazon/Shopify.

Tech Stack

- **Languages:** Python
- **Frameworks/Libraries:** FastAPI, Playwright (scraping where allowed), Pandas, NumPy
- **AI:** LLM for keyword extraction + classification, RAG for similarity checks. (Can run a local LLM model for beta testing)
- **Tools:** GitHub, Jupyter, or Google Colab for prototyping

Dataset

- **Primary:** USPTO trademark data (marks, owners, status, classes).
- **Supplemental:** Marketplace policy docs (Etsy, Amazon, Shopify) for phrase-level guidance; public synonym lists/word embeddings to propose safe alternatives.

Ethical & Legal Considerations

It is crucial that we **let the end user know** the tool provides decision support, not legal advice, as using a trademarked term can have dire consequences for small businesses. **Strict respect** for site Terms of Service and robots.txt; prefer official bulk/OPD feeds where possible over scraping. **Privacy-focused** design: do not store user images/text beyond processing unless

users opt in.

Team Structure

- **Target Team Size:** 2–3 students
- **Full-Stack / Backend SWE:** APIs, ingestion pipelines, vector index, auth/billing.
- **ML/NLP Engineer:** LLM prompt/eval loops, keyword extraction, risk scoring, synonym generation.

Timeline (Rough)

<ul style="list-style-type: none">• Exploring the trademark rules and tools we can use.• Deciding on the optimal LLM to use.• Deciding on the infrastructure the model should run on.	Week 1-2
<ul style="list-style-type: none">• Set up USPTO ingestion: start with compliant web-scrape prototype (ToS-safe) while evaluating official bulk feeds; draft schema for marks/classes/owners.• Build a minimal vector index (pgvector/FAISS) and similarity endpoints.	Week 3-4
<ul style="list-style-type: none">• LLM pipeline for keyword extraction + product class detection; initial risk scoring (exact/near-match, class overlap, fame list).	Weeks 5
<ul style="list-style-type: none">• CLI/Notebook demo for internal testing.• Instrumentation (latency, errors) and human-in-the-loop feedback UI.	Weeks 6-7
<ul style="list-style-type: none">• Private beta with known sellers; measure time saved per listing and takedown avoidance.• Improve precision/recall; add basic “phrase rationales” (why flagged, sources).	Weeks 7-9
<ul style="list-style-type: none">• Potentially make a front-end website.	Week 10

Project Member Requirements

Strong requirements:

- Some experience in LLMs
- Fluency in Python.

Preferred requirements:

- Experience on a similar project.
- Infrastructure knowledge (hosting LLMs, API usage, Web Scraping)

Time Commitment:

- 4 hours per week.

