**📝 Updated Project Blueprint & Roadmap**

**Overall Goal:** Create an automated system to analyze KSA brands (primarily merchandise-focused) based on social hype and e-commerce data, providing actionable insights for licensing via a user-friendly tool.

**Phase 1: Data Foundation Refinement** 🧱 *(Focus: Fixing & Expanding Data)*

* PENDING ⏳ **(CRITICAL NEXT STEP)** **Fix Twitter Data:** Investigate why tweet text/engagement counts were missing.
  + Check Apify Actor (xtdata/twitter-x-scraper) output logs directly in Apify Console for a successful run.
  + If Actor output *is* complete, debug the saving logic in 1\_scrape\_hype.py.
  + If Actor output is *incomplete*, try a different Apify Twitter Actor (e.g., apify/twitter-scraper, apidojo/tweet-scraper).
  + Re-run the Twitter scrape once fixed.
* PENDING ⏳ **Expand Brand List:** Research and compile a larger list of relevant Saudi/cultural merchandise brands (fashion, characters, food, sports, influencers). Add these to BRANDS\_TO\_TRACK.
* PENDING ⏳ **Re-run E-commerce Scrape:** Run 2\_scrape\_ecommerce\_apify.py (Amazon-only script) for the expanded brand list.
* PENDING ⏳ **(Optional but Recommended)** **Scrape Amazon Reviews:** Implement 3\_scrape\_reviews\_apify.py using a suitable actor (e.g., apify/amazon-reviews-scraper) to get review text for calculating Counterfeit Risk.

**Phase 2: Automated ETL Pipeline** ⚙️ *(Focus: Making Refresh Easy)*

* PENDING ⏳ **Consolidate Processing:** Move the cleaning and feature engineering steps (Sentiment, Topic Modeling (if Twitter data fixed), Volume, Saturation, Quality, Reviews, Counterfeit Risk, Scoring) from the Jupyter Notebook into one or more Python scripts (process\_data.py).
* PENDING ⏳ **Create Master Script:** Develop a main script (run\_pipeline.py) that:
  + Optionally clears old data from the database.
  + Calls the scraping scripts (1\_scrape..., 2\_scrape..., 3\_scrape...) in sequence.
  + Calls the data processing script (process\_data.py).
  + Outputs the final brand\_metrics\_final.csv.
  + Includes basic error handling (e.g., log failures, maybe skip a failing brand).

**Phase 3: Analysis & Application Refinement** 📊 *(Focus: Insights & Tool)*

* PENDING ⏳ **Refine Metrics/Scoring:** Based on the richer data (sentiment, topics, counterfeit risk), adjust the Suitability Score formula and normalization in process\_data.py.
* PENDING ⏳ **Enhance Visualizations:** Update the plotting code (maybe move it into process\_data.py or keep in the notebook for exploration) to use the new metrics.
* PENDING ⏳ **Update GUI Tool:** Modify consultant\_tool.py to:
  + Read the refreshed brand\_metrics\_final.csv.
  + Display the new metrics (Sentiment, Counterfeit Risk).
  + Potentially add filtering options (e.g., by category if we add that).
  + Refine the radar chart and recommendations based on the richer data.

**Phase 4: Specialized Analysis (Future)** 🚀 *(Focus: Tailored Insights for Non-Merchandise Brands)*

* FUTURE 🔮 **PIF Analysis:** If desired, create a separate workflow to analyze sentiment/news around PIF's key portfolio companies.
* FUTURE 🔮 **Aramco Analysis:** If desired, create a workflow to analyze brand reputation, CSR sentiment, etc.
* FUTURE 🔮 **Riyadh Season Analysis:** If desired, create a workflow to analyze buzz around specific events/zones within the season.

**Okay, let's start with Phase 1 - Fixing the Twitter Data.**

**Action:**

1. Go to your Apify Console (console.apify.com).
2. Find one of the successful runs of the xtdata/twitter-x-scraper Actor (like the one for "Fanatics" or "Lazurde" from earlier logs).
3. Go to the "Storage" -> "Dataset" tab for that run.
4. Preview the data (or download the JSON).

**Look closely at a few tweet items. Do they contain the actual text field with the tweet content, and likeCount, retweetCount fields with numbers?** Let me know what you find in the raw output.