**Task-B: Enhancement Plan — Multi-language Sentiment Support**

**Goal:** Extend the app to accurately analyze sentiment in multiple languages.

1. **Language detection**
   * Add langdetect or fasttext language ID at input time.
   * Return the detected ISO code (e.g., “en”, “hi”, “fr”) into the backend response for logging/debugging.
2. **Per-language pipelines**
   * **Rule-based (quick):** If language is supported by lexicon tools (e.g., VADER-like resources for specific languages), load the corresponding lexicon.
   * **Transformer (recommended):** Use multilingual models (e.g., cardiffnlp/twitter-xlm-roberta-base-sentiment, nlptown/bert-base-multilingual-uncased-sentiment, or mBERT-based heads).
   * Create a registry: language → model\_fn so runtime can dispatch to the right pipeline.
3. **Preprocessing locale awareness**
   * Tokenizer & lemmatizer appropriate to language:  
     • spaCy language models (es\_core\_news\_sm, fr\_core\_news\_sm, de\_core\_news\_sm) where licensing allows.  
     • Unicode-aware punctuation, diacritics, and clitics.
   * Stopwords per language (NLTK/spaCy lists or curated lists).
   * Maintain **two code paths** (like we do now): raw text for rule/lexicon models, cleaned text for transformers.
4. **Unified 3-class output**
   * Some multilingual checkpoints output 2 classes or 5 stars; map to {pos, neu, neg}:  
     • 5-star → {1-2: neg, 3: neutral, 4-5: pos}.  
     • 2-class → derive neutral via confidence band (as in current code).
5. **Evaluation & calibration**
   * Build a small multilingual validation set (e.g., 100–200 samples per language from public datasets like **Multilingual Amazon Reviews**, **Twitter Sentiment datasets**).
   * Compute macro-F1 per language. Tune neutral band thresholds per language or per model.
6. **Caching & performance**
   * Lazy-load models when a language is first seen; cache in memory.
   * Optionally serve with **Uvicorn + FastAPI** for higher throughput and async I/O if scaling.
7. **UI & UX**
   * Display detected language and model used.
   * Add a dropdown override so users can manually set the language if detection is wrong.
8. **Documentation & i18n**
   * Internationalize the frontend (labels/messages) using a lightweight i18n library.
   * Provide instructions to add a new language (where to drop stopwords/lexicon & how to register a model).

**Notes on risks/mitigations**

* **Sarcasm & domain shift:** Prefer domain-adapted fine-tuning and augmentation.
* **Mixed-language inputs:** Segment by sentence and route per segment; aggregate with length-weighted averaging.
* **Latency:** Preload popular languages; set sensible max text length with truncation + proportional sampling.