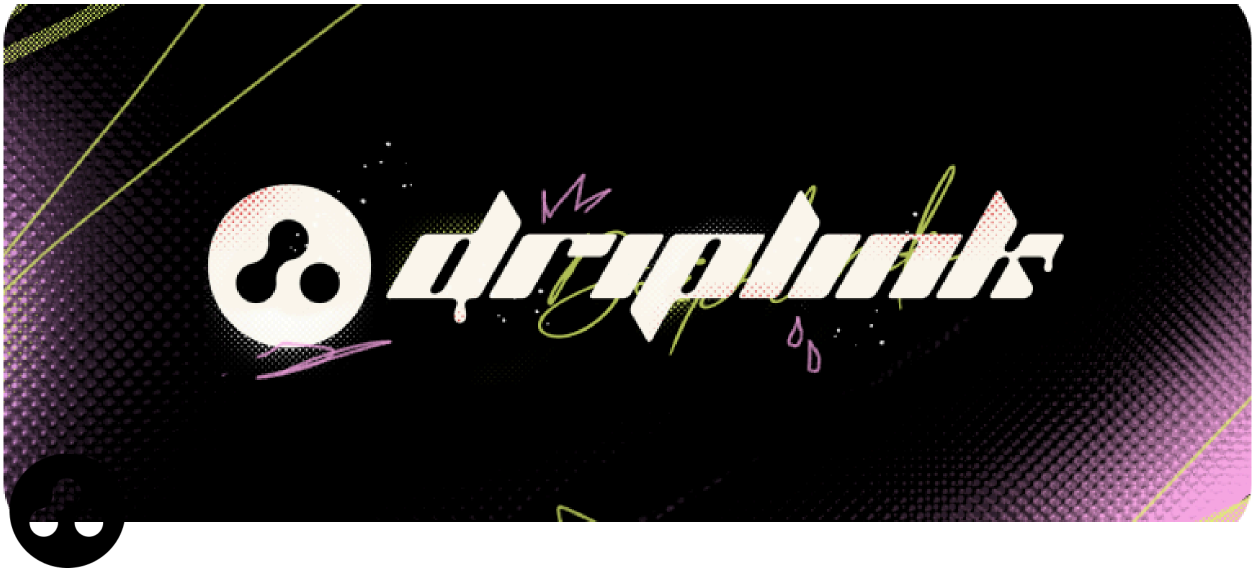

DripLink BE Intern Assignment

Assignment: Building a Language Detective Service for Audio

Goal

Build a service that detects the spoken language in an audio file by integrating with multiple AI providers.

Core Requirements

1. **Provider Connectors**
    - Create separate Python functions (connectors) for these 4 providers:
        - **OpenAI**
        - **Google Gemini**
        - **Sarvam AI**
        - **ElevenLabs**
    - Each connector:
        - Accepts an audio file path.

        - Returns a language code (e.g., `"en"`, `"hi"`, `"ta"`).
        - Handles errors without crashing.
    - **At least 2 connectors must be fully implemented** (real API calls); the other 2 can return fixed mock responses.

2. **Coordinator Function**
    - Orchestrates calls to all 4 connectors for a given file.
    - For each provider, record:
        - Provider name
        - Detected language
        - Time taken (seconds)
        - Estimated cost (tokens and dollar usage per provider)
        - Status (`success`, `failure`, `error`)
        - Error message, if any
3. **API Endpoint**
    - Implement a **FastAPI** `POST /detect/language` endpoint.
    - Request: JSON with `audio_file_path` and `ground_truth_language` (for context only).
    - Response: JSON list of results from all 4 providers, with details above.

Deliverables

- Python codebase with separate modules for connectors, coordinator, and API.
- Working integration for at least 2 real providers.
- Robust handling of API/network errors.

Evaluation

- Accuracy of detection for implemented providers.
- Proper reporting of timing, cost, status, and errors.
- Bonus point for accuracy with indian languages.

- Bonus points for minimising the language detection time for connectors.

💻 Tools You'll Use

- **Python 3.10+**
- **FastAPI**: For building the web service.
- UV (Astral): For Python package management