Overview

My solution is built as a standalone **Next.js** application, utilizing **Azure OpenAI** for processing. The architecture was designed in order to meet the latency requirements of the field agents (5 seconds) and the strict network security constraints of a federal Azure environment.

Technical Stack

- **Framework:** Next.js (App Router)
- **AI Engine:** Azure OpenAI (GPT-4.1-mini)
- **Image Processing:** Sharp (for pre-processing/refinement)
- **Deployment Target:** Azure Static Web Apps (to ensure FedRAMP/Firewall compatibility)

Tools Used

- **Github Copilot**

Design Decisions

1. **Sarah Chen's 5 Second Latency and Batch Upload Requirements.**

   To meet Sarah Chen's requirement for results in ~5 seconds, I opted to use GPT-4.1-mini, a lightweight model with high OCR benchmark scores, for the label extraction and validation process.

   - **Why not a different model:** While "Reasoning" models (like GPT-5-Mini) offer higher accuracy, their 15s+ latency would lead to agents abandoning the solution.
   - **Making GPT 4.1-mini more accurate:** I used a Parallel Ensemble Approach. By running three extraction calls simultaneously, and taking the best parts of each, I achieve the speed of a "mini" model with the accuracy of a much larger one through redundancy.
   - **Batch Upload:** The system supports batch processing, allowing the upload of an image folder and a corresponding CSV file (ideally provided by the COLA system) containing the necessary application data. Labels are matched to the CSV data and processed in parallel batches of 5 to manage current API rate limits.
2. **Marcus William's Firewall Requirement:**

   Following Marcus Williams' feedback regarding blocked outbound traffic, I chose not to use APIs outside of **Azure-native services**. This ensures that the prototype can be white-listed within the existing TTB infrastructure without the "scanning vendor disaster" of 2023.

3. **Dave Morrison’s “Human Judgement” and Jenny Park’s exact Government Warning**

   To address Dave’s concerns about nuance (e.g., "Stone's Throw" vs. "STONE'S THROW"), I implemented a dual-layer validation:

   - **AI Judgment Layer:** For fields like Brand Name and Class/Type, the AI evaluates "semantic equivalence" rather than strict string matching.
   - **Hard-Coded String Matching:** For the **Government Warning**, the system enforces 100% literal accuracy (caps/bolding/text), meeting Jenny Park’s strict compliance standards.

The Extraction Pipeline: Parallel Consensus

Rather than relying on a single (potentially hallucinatory) pass, the system follows a three-step synthesis:

1. **Extraction (Parallel):** Three separate instances of the model extract data from the label. Each is given the "definition" of the fields (e.g., "What is a Brand Name?") without being told the "expected" answer to prevent confirmation bias/hallucination.
2. **Evaluation (Parallel):** A separate AI evaluates each extracted field against the application data, assigning a binary (0 or 1) score based on accuracy and compliance rules.
3. **Heuristic Synthesis:** A final layer selects the highest-scoring extraction for each category, combines them together then presents the final decision of if the label is valid or not to the user.

Assumptions, Limitations & Errors

Limitations:

- **API Rate Limits:** Current API request limits (100 requests per minute on the personal Azure account) restrict the speed of batch processing. While the system could potentially handle 100+ labels simultaneously with higher limits, it currently processes large batches in a parallel queue of 5 labels to manage the constraint.
- **Limited AI Model Options:** Azure AI API gives access to a limited number of models (my account only had access to OpenAI models). I would likely have used Gemini 3 Flash if I had access to the google library.

Assumptions:

- **COLA System CSV Export:** It is assumed that the COLA system can export a CSV file containing the expected label information. If this export functionality is unavailable, users will be required to manually input the necessary details for each bottle for batch requests.
- **Image Format:** Labels are assumed to be uploaded as a single continuous image, consistent with the examples provided on the TTB website.

Errors:

- **Azure API Content Filter:** Occasionally, the Azure AI filter may flag alcohol bottle labels as inappropriate, resulting in a 400 "Image processing blocked due to content policy violation" error. In the event this occurs, my system is designed to re-upload the image and retry the process, which typically resolves the issue.
- **Api Limit:** When the GPT-4.1-Mini hits an api call limit it throws a 500 error. My system starts to use a backup model, GPT 4.1, when this happens.

Example Walkthrough: “OLD TOM DISTILLERY BOTTLE”

Extracting OLD TOM DISTILLERY bottle label

Expected Details

- Brand Name: "OLD TOM DISTILLERY"
- Class/Type: "Kentucky Straight Bourbon Whiskey"
- Alcohol Content: "45% Alc./Vol. (90 Proof)"
- Net Contents: "750 mL"
- Government Warning: [Standard government warning text]

Step 1. Perform 3 extraction calls in parallel of uploaded label image using GPT 4.1 Mini

Extraction 1 Output	Extraction 2 Output	Extraction 3 Output
Brand Name: Old Tom Distillery	Brand Name: Distillery	Brand Name: idTom Dry
Class/Type : “Straight”	Class/Type: "Kentucky Straight Bourbon Whiskey"	Class/Type: “Kentucky”
Alcohol Content: “30% Alc./Vol. (60 Proof)”	Alcohol Content: “40% Alc./Vol. (80 Proof)”	Alcohol Content: “45% Alc./Vol. (90 Proof)”
Net Contents: "750 mL"	Net Contents: "300 mL"	Net Contents: "254 mL"

Government Warning : Issue with label capitalization	Government Warning : Perfect Label	Government Warning : Issue with label wording

Step 2. Evaluating each extraction against expected data in parallel (0 invalid label \| 1 valid)

Evaluation 1 Output	Evaluation 2 Output	Evaluation 3 Output
Brand Name: Old Tom Distillery | 1	Brand Name: Distillery | 0	Brand Name: IdTom Dry | 0
Class/Type : “Straight” | 0	Class/Type: "Kentucky Straight bourbon whiskey" | 1	Class/Type: “Kentucky” | 0
Alcohol Content: “30% Alc./Vol. (60 Proof)” | 0	Alcohol Content: “40% Alc./Vol. (80 Proof)” | 0	Alcohol Content: “45% Alc./Vol. (90 Proof)” | 1
Net Contents: "750 mL" | 1	Net Contents: "300 mL" | 0	Net Contents: "254 mL" | 0
Government Warning : Issue with label capitalization | 0	Government Warning : Perfect Label | 1	Government Warning : Issue with label wording | 0

Step 3. Synthesize the best elements from each extraction and output that to the user

Combined Extraction : VALID
Brand Name: Old Tom Distillery | 1
Class/Type: "Kentucky Straight Bourbon Whiskey" | 1
Alcohol Content: “45% Alc./Vol. (90 Proof)” | 1`
Net Contents: "750 mL" | 1
Government Warning : Perfect Label | 1