

# Take-Home Project: AI-Powered Alcohol Label Verification App

## Project Background & Stakeholder Context

*The following document contains notes from our discovery sessions with the Compliance Division, along with technical requirements for the prototype. We've included stakeholder feedback to give you context on how this tool will be used.*

---

### Interview Notes: Sarah Chen, Deputy Director of Label Compliance

*Conducted Tuesday, 3:15 PM — Sarah was running late from her daughter's school play rehearsal*

"Thanks for meeting with me. Sorry about the delay—my daughter's playing the lead in her school's production of *Annies* next week and rehearsals have been crazy. Anyway, let me tell you about what we're dealing with here.

So the TTB reviews about 150,000 label applications a year. Our team of 47 agents handles all of them. Back in the 80s—before my time—they actually had over 100 agents, but budget cuts, you know how it goes. We've been doing things basically the same way since the COLA system went online in 2003. That was a big upgrade from paper forms, believe it or not.

The actual review process is pretty straightforward. An agent pulls up an application, looks at the label artwork, and checks that what's on the label matches what's in the application. Brand name matches? Check. ABV is correct? Check. Government warning is there? Check. It takes maybe 5-10 minutes per application for a simple one, longer if there are issues.

Here's the thing though—and this is what got leadership interested in AI—a lot of what we do is just... matching. Like literally just making sure the number on the form is the same as the number on the label. My agents spend half their day doing what's essentially data entry verification. It's not that they can't do more complex analysis, it's that they're drowning in routine stuff.

Oh, I should mention—we tried a pilot with the scanning vendor last year. Disaster. The system would take 30, 40 seconds sometimes to process a single label. Our agents just went back to doing it by eye because they could do five labels in the time it took the machine to do one. **If we can't get results back in about 5 seconds, nobody's going to use it.** We learned that the hard way.

What else... The agents really vary in their tech comfort level. Dave's been here since the Clinton administration and still prints his emails. Meanwhile, Jenny's fresh out of college and probably could have built this tool herself. We need something **my mother could figure out**—she's 73 and just learned to video call her grandkids last year, if that gives you a benchmark. Half our team is over 50. Clean, obvious, no hunting for buttons.

One more thing that came up in our last team meeting—during peak season, we get these big importers who dump 200, 300 label applications on us at once. Right now we literally have to process them one at a time. If there was some way to **handle batch uploads**, that would be huge. Janet from our Seattle office has been asking about this for years."

---

## **Interview Notes: Marcus Williams, IT Systems Administrator**

*Coffee chat, Thursday morning*

"Sarah probably gave you the business side. Let me fill you in on some of the technical landscape.

Our current infrastructure is... well, it's government infrastructure, let's leave it at that. We're on Azure now after the migration in 2019. That was a whole thing—don't get me started on the FedRAMP certification process. Took 18 months just for the paperwork.

The COLA system is built on .NET, though there's been talk about modernizing it for years. We had a contractor come in last summer to do an assessment and they quoted us \$4.2 million for a full rebuild. That went nowhere, obviously.

For this prototype, we're not looking to integrate with COLA directly—that's a whole different beast with its own authorization requirements. Think of this as a standalone proof-of-concept that could potentially inform future procurement decisions. If it works well, maybe we look at how to incorporate it into the workflow. But that's years away, realistically.

Security-wise, we'd need to be careful with any production deployment—there's PII considerations, document retention policies, the usual federal compliance stuff. But for a prototype? Just don't do anything crazy. We're not storing anything sensitive for this exercise.

Oh, and our network blocks outbound traffic to a lot of domains, so keep that in mind if you're thinking about cloud APIs. During the scanning vendor pilot, half their features didn't work because our firewall blocked connections to their ML endpoints. Classic."

---

## **Interview Notes: Dave Morrison, Senior Compliance Agent (28 years)**

*Brief hallway conversation*

"Look, I'll be honest, I've seen a lot of these 'modernization' projects come and go. Remember the automated phone system they put in back in 2008? Supposed to reduce call volume. We ended up with more calls because nobody could figure out how to navigate it.

The thing about label review is there's nuance. You can't just pattern match everything. Like, I had one last week where the brand name was 'STONE'S THROW' on the label but 'Stone's Throw' in the application. Technically a mismatch? Sure. But it's obviously the same thing. You need judgment.

That said, I'm not against new tools. If something can help me get through my queue faster, great. Just don't make my life harder in the process. I spend enough time fighting with COLA as it is."

---

### **Interview Notes: Jenny Park, Junior Compliance Agent (8 months)**

*Teams call, Friday afternoon*

"I'm so excited you're working on this! When I started here, I was kind of shocked at how manual everything is. Like, I literally have a printed checklist on my desk that I go through for every label. Brand name—check with my eyes. ABV—check with my eyes. Warning statement—check with my eyes. It's 2024!"

The one thing I'd say is the warning statement check is actually trickier than it sounds. It has to be **exact**. Like, word-for-word, and the 'GOVERNMENT WARNING:' part has to be in all caps and bold. Sarah probably mentioned this but people try to get creative with the warning all the time. Smaller font, different wording, burying it in tiny text. I caught one last month where they used 'Government Warning' in title case instead of all caps. Rejected.

Also—and this is maybe out of scope for a prototype—but it would be amazing if the tool could handle images that aren't perfectly shot. I've seen labels that are photographed at weird angles, or the lighting is bad, or there's glare on the bottle. Right now if an agent can't read the label they just reject it and ask for a better image. But if AI could handle some of that..."

---

## **Technical Requirements**

You are free to use any programming languages, frameworks, or libraries you prefer. We want to see what kind of engineering, design, and integration decisions you make.

---

# Additional Context

## About TTB Label Requirements

For reference, TTB requires specific information on alcohol beverage labels. The exact requirements vary by beverage type (beer, wine, distilled spirits) but common elements include:

- Brand name
- Class/type designation
- Alcohol content (with some exceptions for certain wine/beer)
- Net contents
- Name and address of bottler/producer
- Country of origin for imports
- **Government Health Warning Statement** (mandatory on all alcohol beverages)

We encourage you to review TTB's guidelines at ttb.gov for additional context on label requirements.

## Sample Label

Your app should handle labels containing information like the example below:

### Example Distilled Spirits Label Fields:

- 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]

*We encourage you to create or source additional test labels—AI image generation tools work well for this.*

---

# Deliverables

1. **Source Code Repository** (GitHub or similar)
  - All source code
  - README with setup and run instructions
  - Brief documentation of approach, tools used, assumptions made
2. **Deployed Application URL**
  - Working prototype we can access and test

---

## Evaluation Criteria

- Correctness and completeness of core requirements
- Code quality and organization
- Appropriate technical choices for the scope
- User experience and error handling
- Attention to requirements
- Creative problem-solving

We understand this is time-constrained. A working core application with clean code is preferred over ambitious but incomplete features. Document any trade-offs or limitations.

---

*Questions? Reach out for clarification—though we also value how you fill in gaps independently.*

Good luck!