

Executive summary

I can absolutely deliver a real online ordering experience for this liquor store, but I'm not going to pretend a "full-scale custom Amazon clone" is realistic for me (no backend experience) without serious risk. The only sane way I ship something reliable is **platform-first** (Shopify/Square/industry vendor), with a scoped MVP, and with compliance and operations treated as first-class requirements—not afterthoughts.

If I ignore that and try to "vibe code" a custom backend with payments + accounts + delivery + alcohol compliance, the most likely outcome is: delays, broken checkout, inventory chaos, chargebacks, and potentially legal/compliance problems.

What I can do (and ship) without blowing up the project

1) Build a legit storefront that converts

- Clean branded website (mobile-first)
- Categories, search, filters (beer/wine/spirits, brand, price, size)
- Product pages with clear pricing, size, ABV (optional), stock status
- Promotions (featured products, bundles if allowed)
- SEO basics (local keywords + pages that rank)

What this solves: customers can browse easily and buy without friction.

2) Enable real checkout and payments safely

I can implement **real payments** using a platform checkout (not my own code).

- Card payments, Apple Pay/Google Pay (platform-dependent)
- Taxes configuration
- Refunds handled through the platform

Why this matters: I avoid building a fragile payment system and avoid security liability.

3) Support pickup + local delivery workflows

I can set up:

- **Pickup:** time windows, pickup instructions, order status updates
- **Local delivery:** delivery zones, minimums, fees, hours
- Optional: “delivery partner” approach (depending on what they already use)

What this solves: orders actually get fulfilled, not just “placed.”

4) Implement minimum viable alcohol compliance UX

- Age gate on entry (basic)
- Checkout acknowledgment (“I’m 21+ / legal age”)
- Clear policy pages and banners
- Prominent reminders that ID is required at pickup/delivery

Important: UI is not enough. I can build the prompts and policy surfaces, but compliance still depends on how they fulfill orders.

5) User accounts (only if truly needed)

I can add customer accounts for:

- Order history
- Saved addresses
- Faster checkout

But I’m going to be blunt: **accounts are optional** for a successful MVP. Guest checkout converts better early on and reduces complexity.

6) Back-office admin that doesn’t suck

I can set up a workflow where staff can:

- Receive new orders via email/SMS/app
- Update order statuses
- Manage inventory and product pricing via CSV/POS integration (best case)

This is where projects live or die. If staff can't use it, the website is pointless.

My recommended scope: MVP first, then expand

Phase 1 (MVP): “Order online + get it fulfilled”

Deliverables

- Storefront + catalog
- Pickup (and/or local delivery)
- Real payments via platform
- Age-gating UX + policy pages
- Order notifications + basic status handling

This is the version I can confidently launch.

Phase 2: Growth + automation

- Loyalty / rewards
- Discount codes, promos, bundles
- Better search + recommendations
- Email/SMS marketing flows
- Review widgets
- Analytics dashboard

Phase 3: Advanced integrations (only if revenue justifies it)

- Deep POS sync
- Delivery platform integrations
- Multi-location support

- Custom app features
-

What can go wrong (and how it actually happens)

A) Legal/compliance issues (high impact)

What can go wrong

- Selling or delivering alcohol where it's not allowed
- No clear ID verification process at pickup/delivery
- Staff hands off alcohol without proper checks
- Marketing language that implies shipping/delivery beyond legal boundaries

How it bites us

- Fines, license risk, forced shutdown of online sales
- Processor bans or chargeback spikes

My mitigation

- I force a written delivery/pickup policy and “where we serve” rules before launch.
 - I push for pickup-only or tightly bounded local delivery until legal is crystal clear.
-

B) Payments risk (high impact)

What can go wrong

- The payment processor flags alcohol as restricted and freezes funds
- Chargebacks rise because orders are delayed / mismatched / refunded poorly
- Fraud attempts (alcohol is a common fraud target)

How it bites us

- Store loses money or access to card payments
- “We can’t accept payments” = business dead

My mitigation

- Use Shopify/Square/approved gateway and avoid custom payment coding.
- Enforce clear fulfillment times, cancellation rules, and verification steps.

C) Inventory problems (high impact, extremely common)

What can go wrong

- Website shows items in stock that are actually out of stock
- Prices don't match shelf prices
- Staff can't keep product photos/data updated
- DoorDash catalog and website diverge

How it bites us

- Angry customers, refunds, bad reviews, chargebacks
- Staff stops trusting the site → stops using it → site dies

My mitigation

- Start with a smaller curated catalog if needed.
- Decide early: POS sync vs manual inventory.
- Set out-of-stock rules and substitution policy.

D) Delivery failures (high impact)

What can go wrong

- Delivery times are unrealistic
- No driver availability
- Orders placed outside service area
- Failed ID check at door causes returns/refunds

How it bites us

- Customers blame the store (not the tech)
- Refunds + support overhead explode

My mitigation

- Strict delivery zones/hours and hard cutoffs.
- Clear "ID required" messaging plus driver procedure.
- If delivery is complex, I push for pickup-only at launch.

E) Security and privacy mistakes (high impact)

What can go wrong

- Weak admin passwords
- Using sketchy plugins/apps
- Collecting more customer data than necessary
- Storing anything sensitive incorrectly (especially if I build custom)

How it bites us

- Data leaks, legal exposure, reputation damage

My mitigation

- Platform-based auth + checkout.
- Minimal data collection.
- Only vetted apps; keep the stack lean.

F) Project management failure (the silent killer)

What can go wrong

- Owner keeps adding “one more feature”
- No one is responsible for product photos/descriptions
- I get stuck waiting for approvals/assets
- They expect DoorDash-level logistics for a website budget

How it bites us

- Timeline slips, scope explodes, I end up unpaid or blamed

My mitigation

- I define scope in writing, with a change process.
- I set deadlines for content delivery (photos, pricing, catalog).
- I ship MVP first and don’t negotiate against reality.

The hard constraints I will set (so I don't fail)

1. I will not build a custom payment system. I will use a platform checkout.
 2. I will not promise "full-scale" without defining what that means in features and operations.
 3. I will not launch delivery until zones, hours, fees, and ID workflow are operationally agreed.
 4. I will not accept "we'll figure inventory later." Inventory is the product.
-

The platform decision (what I'd pick)

If they already use **Square POS** → I'll choose **Square Online** to reduce inventory/payment headaches.

If they want the most polished storefront + flexibility → I'll choose **Shopify**.

If they want industry-specific turnkey + budget allows ,I'll recommend a liquor-focused vendor.

I will not choose "custom backend from scratch" unless they pay for a real backend developer or I bring one in.

What I need from the store (or the project stalls)

- Their POS system (Square/Clover/Lightspeed/none)
 - Their product list (CSV/export) OR permission to build a smaller curated catalog first
 - Delivery decision: pickup only vs local delivery
 - Service area + hours + minimum order + delivery fee rules
 - Photos/branding assets (logo, colors, store photos)
 - Written policies (refunds, substitutions, ID requirements)
-

My "go/no-go" criteria before I start

I will proceed if:

- They accept an MVP scope and platform approach
- They can provide catalog/inventory source

- They agree on fulfillment workflow and compliance steps

I will walk away if:

- They demand “full-scale custom” + “do it fast” + “cheap”
- They refuse to define delivery/ID workflow
- They expect me to magically sync inventory without POS access/data