Grocery Delivery Marketplace MVP

Comparable to: Instacart / Shipt

A lean on-demand grocery marketplace for a defined service zone. Customers browse local store catalogs, place paid orders, and receive delivery updates.

Estimated MVP cost
$21,750
Estimated effort
385 hours

To get your own customised cost for an app like this, visit www.chatstack.app and start a ChatStack interview describing how you want it customised.

Key Features

- Browse admin-managed local store catalogs with simple availability updates
- Let customers select substitution or refund preferences before checkout
- Use card payment authorization at checkout and capture the final adjusted total after shopping
- Provide a combined shopper-driver workflow for picking and delivery
- Show customer order status and an estimated delivery time
- Provide admin operations management

Out of Scope for This MVP

- Initial launch is limited to one city or tightly defined service zone
- Use a simple flat delivery fee
- US hosting, payments, and regulatory compliance
- Lean MVP scope with simple defaults for non-essential capabilities
- Store catalogs and availability are managed by marketplace administrators
- A single worker completes both shopping and delivery for each order
- Final order totals can change based on substitutions, refunds, and item availability

Full Cost Breakdown

Item	Hours	Cost
The system shall allow visitors to view the grocery delivery service description, operating model, and configured service area before registration.	3	$170
The system shall support customer registration, login, logout, account profile management, privacy disclosures, and an in-app account-deletion flow.	10	$550
The system shall allow logged-in customers to browse products grouped by local store and see simple in-stock or unavailable states.	8	$440
The system shall allow customers to add available products to a cart, change quantities, remove items, and maintain a single-store order context.	8	$440
The system shall allow customers to define substitution and refund preferences at item or order level before checkout.	6	$330
The system shall allow customers to enter and manage a delivery address and contact details and shall reject addresses outside the configured service zone.	8	$440
The system shall present an order summary including items, quantities, delivery fee, and applicable totals, and shall authorize payment before order processing begins.	12	$660
The system shall create an order with customer, store, item, preference, delivery, payment, and status information after successful checkout.	11	$610
The system shall allow shopper-drivers to view assigned orders, item details, customer preferences, delivery information, and required operational actions.	9	$500
The system shall allow shopper-drivers to record item availability, substitutions, refunds, and quantities while shopping in accordance with customer preferences.	14	$770
The system shall allow shopper-drivers to update order status through assigned, shopping, on the way, and delivered stages.	9	$500

The system shall calculate and display the final adjusted order total after shopping changes and capture or refund the appropriate amount.	14	$770
The system shall allow customers to view current order status, adjusted totals, order details, and an operational estimated delivery time.	8	$440
The system shall provide optional order-progress notifications to customers when supported by the device and permissions.	8	$440
The system shall allow marketplace administrators to create, update, deactivate, and manage local store products and basic availability.	11	$610
The system shall allow marketplace administrators to view and manage orders, oversee statuses, assign shopper-drivers, and perform basic operational actions.	14	$770
The system shall allow marketplace administrators to manage customer, shopper-driver, and administrator roles and configure basic marketplace settings and service-zone rules.	13	$720
Planning	42	$2,310
Project Management	34	$1,870
Testing	20	$1,100
DevOps	17	$940
Design & Frontend Build	50	$2,750
AI Tokens	0	$540
Security	20	$1,100
Risk (10% Contingency)	36	$1,980
Total	**385**	**$21,750**

Estimate Assumptions

- Estimates cover base implementation only and exclude NFR overhead, project management, design discovery, and legal review.
- Supabase Auth, PostgreSQL, Stripe, and the specified mobile and web frameworks are available and configured using standard

patterns.

- The MVP uses one store per order, one flat delivery fee, manually maintained availability, and a simple operational ETA.
- Admin workflows are implemented in the Next.js web application; customer and shopper workflows are implemented in Flutter.
- Notification delivery is optional and does not block checkout, shopping, status updates, or tracking.
- Service-zone validation uses configured geographic rules rather than advanced geocoding or routing integrations.

Risks That Could Change This

- Stripe authorization, adjusted capture, refunds, and webhook edge cases may require additional integration and reconciliation effort.
- Payment, privacy, account deletion, and mobile-store compliance requirements may change after legal or platform review.
- Operational status and adjustment rules may expand beyond the assumed concise MVP workflow.
- Third-party notification permissions and delivery behavior vary by platform and device.
- Address validation and delivery ETA accuracy may require a mapping or geocoding provider not currently specified.

User Stories

US-1 - Visitor
As a visitor, I want to understand the grocery delivery service and service area so that I can decide whether to register.

US-2 - Customer
As a customer, I want to create and manage an account so that I can place and track grocery orders.

US-3 - Logged In Customer
As a logged-in customer, I want to browse available products by local store so that I can choose groceries to order.

US-4 - Logged In Customer
As a logged-in customer, I want to add available products to a cart so that I can prepare my grocery order.

US-5 - Logged In Customer
As a logged-in customer, I want to specify substitution or refund preferences so that unavailable items are handled according to my choices.

US-6 - Logged In Customer
As a logged-in customer, I want to provide a delivery address and contact details so that my order can be delivered to the correct location.

US-7 - Logged In Customer
As a logged-in customer, I want to authorize payment at checkout so that the marketplace can begin processing my order.

US-8 - Logged In Customer
As a customer, I want to see the final adjusted order total after shopping so that I am charged accurately for substitutions, refunds, and availability changes.

US-9 - Shopper-Driver
As a shopper-driver, I want to view assigned orders and their item details so that I can shop for the correct groceries.

US-10 - Shopper-Driver
As a shopper-driver, I want to record substitutions, refunds, and item availability while shopping so that the customer's order and final total remain accurate.

US-11 - Shopper-Driver
As a shopper-driver, I want to update order progress through shopping and delivery so that customers and administrators know the current status.

US-12 - Customer
As a customer, I want to view my order status and estimated delivery time so that I know when to expect my groceries.

US-13 - Marketplace Administrator
As a marketplace administrator, I want to manage local store catalogs and basic availability so that customers can order from accurate product listings.

US-14 - Marketplace Administrator
As a marketplace administrator, I want to monitor and manage orders so that I can resolve operational issues

and support delivery execution.

US-15 - Marketplace Administrator
As a marketplace administrator, I want to manage marketplace users and operational settings so that the service can run within its defined launch zone.

Requirements

FR-1 - Service Discovery
The system shall allow visitors to view the grocery delivery service description, operating model, and configured service area before registration.

FR-2 - Account Management
The system shall support customer registration, login, logout, account profile management, privacy disclosures, and an in-app account-deletion flow.

FR-3 - Product Catalog
The system shall allow logged-in customers to browse products grouped by local store and see simple in-stock or unavailable states.

FR-4 - Shopping Cart
The system shall allow customers to add available products to a cart, change quantities, remove items, and maintain a single-store order context.

FR-5 - Order Preferences
The system shall allow customers to define substitution and refund preferences at item or order level before checkout.

FR-6 - Delivery Details
The system shall allow customers to enter and manage a delivery address and contact details and shall reject addresses outside the configured service zone.

FR-7 - Checkout and Payment
The system shall present an order summary including items, quantities, delivery fee, and applicable totals, and shall authorize payment before order processing begins.

FR-8 - Order Processing
The system shall create an order with customer, store, item, preference, delivery, payment, and status information after successful checkout.

FR-9 - Shopper Workflow
The system shall allow shopper-drivers to view assigned orders, item details, customer preferences, delivery information, and required operational actions.

FR-10 - Shopping Adjustments
The system shall allow shopper-drivers to record item availability, substitutions, refunds, and quantities while shopping in accordance with customer preferences.

FR-11 - Order Status
The system shall allow shopper-drivers to update order status through assigned, shopping, on the way, and delivered stages.

FR-12 - Adjusted Totals
The system shall calculate and display the final adjusted order total after shopping changes and capture or refund the appropriate amount.

FR-13 - Customer Order Tracking
The system shall allow customers to view current order status, adjusted totals, order details, and an operational estimated delivery time.

FR-14 - Notifications
The system shall provide optional order-progress notifications to customers when supported by the device and permissions.

FR-15 - Catalog Administration
The system shall allow marketplace administrators to create, update, deactivate, and manage local store products and basic availability.

FR-16 - Operations Administration
The system shall allow marketplace administrators to view and manage orders, oversee statuses, assign shopper-drivers, and perform basic operational actions.

FR-17 - User and Settings Administration
The system shall allow marketplace administrators to manage customer, shopper-driver, and administrator roles and configure basic marketplace settings and service-zone rules.

NFR-1 - Security and Privacy
The system shall protect authentication credentials, payment information, personal data, addresses, phone numbers, and delivery information using appropriate access controls and encryption.

NFR-2 - Payment Compliance
The system shall use permitted third-party payment methods for physical goods and delivery services and shall not use Apple In-App Purchase or Google Play Billing.

NFR-3 - Performance
The system shall target API response times below 200 milliseconds for standard requests under expected MVP load.

NFR-4 - Scalability
The system shall use a managed database platform capable of scaling with customers, products, orders, and operational activity.

NFR-5 - Observability
The system shall provide application performance monitoring, error visibility, and operational diagnostics for customer, shopper, administrator, and payment workflows.

NFR-6 - Reliability and Data Integrity
The system shall preserve consistent order, item-adjustment, payment, and status data during retries, intermittent connectivity, and partial workflow failures.

NFR-7 - Availability and Degraded Operation
The system shall provide graceful degradation when optional notifications or location permissions are unavailable and shall not block core ordering or tracking functions.

NFR-8 - Usability and Accessibility
The system shall provide clear, consistent, mobile-oriented workflows for registration, shopping, checkout, delivery execution, and order tracking.

NFR-9 - Regulatory and Store Compliance
The system shall support applicable US privacy, payment, and mobile application store requirements for the defined marketplace functionality.

NFR-10 - Auditability
The system shall record timestamps and responsible actors for material order changes, shopper adjustments, status transitions, administrative actions, and account deletion requests.

Requirements Assumptions

- The MVP is English-only and launches in one city or tightly defined US service zone.
- No external integrations are included unless explicitly specified; payment-provider integration is required for checkout.
- Customers purchase physical groceries and delivery services, so permitted third-party payment methods are used instead of mobile in-app purchase billing.
- A single shopper-driver completes shopping and delivery for each order.
- A single-store cart and one flat delivery fee are used for the MVP.
- Catalogs and basic availability are managed manually by marketplace administrators.
- Location services are optional and requested only for shopper-driver workflows that require them.
- The backend uses Supabase managed database capabilities and Firebase Performance Monitoring unless implementation decisions change.
- Advanced inventory synchronization, complex pricing, promotions, subscriptions, tips, multi-language support, and multi-store carts are out of scope.

- Operational delivery estimates are informational and are not guaranteed service commitments.

Technical Specification

TS-1 - Client Applications
Build a Flutter/Dart mobile application for customers and shopper-drivers using Riverpod state management and Material Design 3.

TS-2 - Web and Administration
Build a responsive Next.js, React, and TypeScript web application for marketplace administrators using Shadcn/ui and Tailwind CSS.

TS-3 - Authentication
Use Supabase Auth with JWT-based sessions for registration, login, logout, profile management, account deletion, and secure session handling.

TS-4 - Authorization and RBAC
Implement role-based authorization for customers, shopper-drivers, and marketplace administrators across mobile, web, API, and database access.

TS-5 - Backend and API
Implement a REST API with optional Supabase Edge Functions for order creation, cart handling, shopper adjustments, payment workflows, status updates, notifications, and administration.

TS-6 - Data Storage
Use a Supabase-managed PostgreSQL database for users, roles, stores, products, carts, orders, adjustments, payments, delivery details, statuses, and audit records.

TS-7 - Payments
Integrate Stripe for permitted physical-goods and delivery payments, using tokenization and authorization followed by adjusted capture, refund, or partial refund.

TS-8 - Notifications and Messaging
Support optional order-progress notifications through OneSignal and Firebase Cloud Messaging, with SendGrid for transactional email and Twilio for SMS where required.

TS-9 - Hosting and Environments
Host the web application and admin panel on Vercel and deploy the Supabase backend using managed project environments for development, staging, and production.

TS-10 - CI/CD and Testing
Use GitHub Actions for backend and web CI/CD, CodeMagic for mobile app build and store pipelines, and automated unit, integration, and end-to-end testing.

TS-11 - Security and Compliance
Apply HTTPS, encryption in transit and at rest, secure secrets management, tokenized payments, input validation, audit logging, and role-based data protection.

TS-12 - Observability
Implement application performance monitoring, structured error logging, operational diagnostics, and crash reporting across customer, shopper, admin, and payment workflows.

TS-13 - Reliability and Data Integrity
Design transactional order workflows to preserve consistent cart, order, adjustment, payment, and status data during retries, partial failures, and intermittent connectivity.

Technical Assumptions

- The MVP is English-only and launches in one city or tightly defined US service zone.
- Flutter/Dart is used for the mobile application; Next.js, React, and TypeScript are used for web and administration.
- Supabase provides managed PostgreSQL, authentication, database security, and optional Edge Functions.
- Vercel hosts web applications; GitHub Actions and CodeMagic provide CI/CD pipelines.
- Stripe is the baseline payment provider for physical goods and delivery services.

- One shopper-driver completes both shopping and delivery for each order.
- The MVP uses a single-store cart, one flat delivery fee, and manually managed catalog availability.
- Notifications, location services, email, SMS, analytics, and crash monitoring are optional supporting capabilities.
- No advanced inventory synchronization, promotions, subscriptions, tips, multi-language support, or multi-store carts are included.
- US data residency and retention requirements require confirmation during detailed security and compliance design.